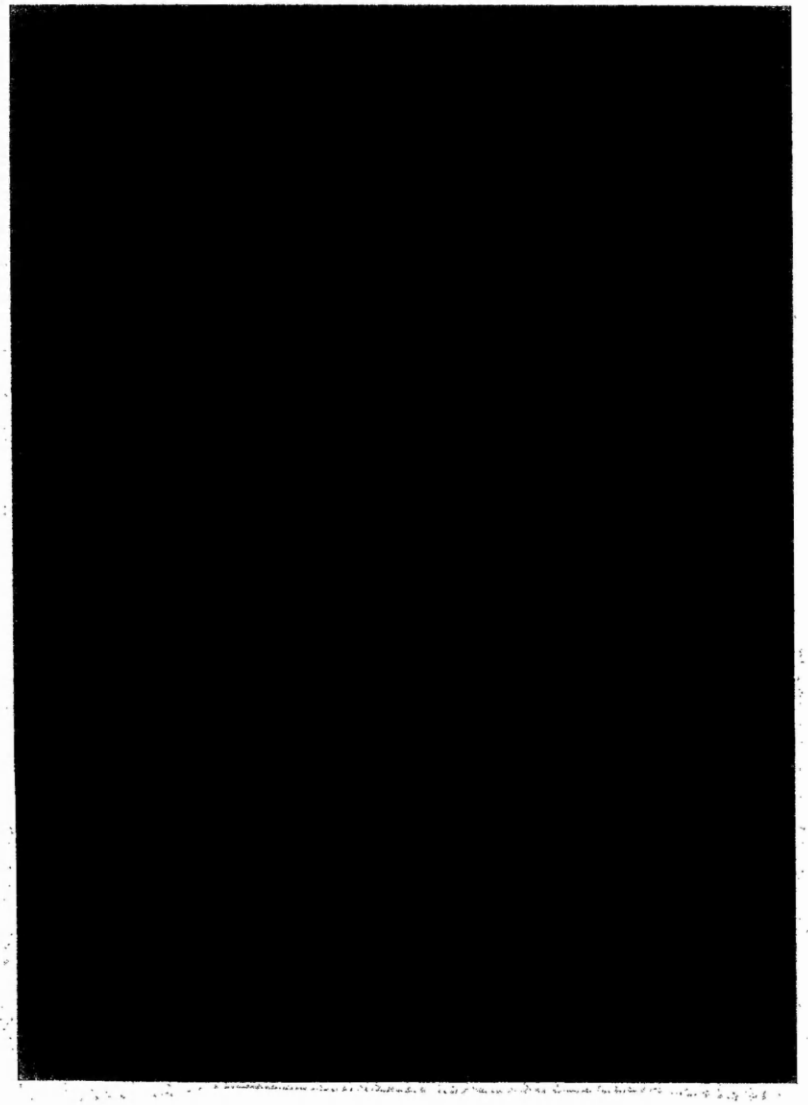


ANALYSIS METHODS FOR SOFTWARE RELIABILITY DATA

BY

CHRISTOPHER McCOLLIN BSc MSc MIQA

This thesis has been submitted in partial fulfilment of the requirements of the University of Nottingham Trent University for the degree of Doctor of Philosophy.

Sponsoring establishment: Nottingham Trent University

January 1993

Christopher McCollin :

Analysis Methods for Software Reliability Data

This thesis reviews the statistical models commonly applied to software reliability data. A data set encompassing the typical fields to be found on a software defect record sheet is analysed in a systematic way to initially determine where data was corrupted or uncollected. The data when summarised into failure counts, proportions, waiting times to failure and cumulative failure times are analysed by a number of statistical analyses : Exploratory Data Analysis, Box and Jenkins time series, proportional hazards modelling, proportional intensity modelling and a number of multivariate techniques. A comparison of the analyses is undertaken.

The time series analysis using a standard computer package was able to forecast when the software would become failure free, a useful metric to determine time to release the software to a customer. The results are verified by proportional hazards modelling.

The intensity functions of most of the non-homogeneous Poisson processes are shown to be equivalent to proportional hazards models with appropriate explanatory factors and hazard functions. The technique may be used as a diagnostic tool for the selection of the most appropriate software reliability model for a given data set as nonsignificant proportional hazards formulations are rejected from the analyses. Covariates which describe the attributes of the software, e.g. source program type, may also be incorporated in a proportional hazards formulation.

The proportional intensity model is applied to the twelve least reliable program sources of Alvey data set number 3, the first analysis of this type for software data. This formulation can model all the software and hardware reliability growth models which can be expressed as Non-homogeneous Poisson processes. The findings are compared with those from exploratory data analysis and proportional hazards modelling. The proportional intensity model is also shown to be a limiting form of the proportional odds model.

The use of multivariate techniques such as principal components analysis, discriminant analysis and also generalised linear modelling to model software reliability data are described and the results are compared to the results of the analyses from exploratory data analysis and proportional intensity modelling.

Objectives

The main objectives of the research for this thesis:

(i) To review statistical techniques applied to software reliability data and to investigate a variety of methods for analysing the data which has been collected under a data collection scheme.

(ii) To propose a methodology for the analysis of a large software development failure data set as collected by a standard and/or standardised (e.g. BS5750) data collection method.

(iii) To analyse subsets of the collected data by exploratory methods using available and purpose written software and thus determine appropriate statistical models for the data.

(iv) To develop the proportional hazards technique as a diagnostic tool to subsume the well known software reliability models within it's framework and apply the methodology to part of the described data set.

(v) To investigate proportional intensity modelling for software reliability modelling and show it's relationship to generalised linear modelling.

Advanced Studies

The following advanced studies were undertaken in connection with the programme of research for this thesis:

(i) Participation at the following conferences and seminars:

Safety and Reliability Society Symposium 1988
The Institute of Mathematics and its Applications (lecture on Proportional hazards modelling) 1989
The Institute of Quality Assurance (regional meeting lecture on Software Quality and Reliability Assurance)
6th Euredata Conference 1989, Siena
Reliability 1989, Brighton
Mechatronics 1990, Cambridge
Safety and Reliability Conference, Symposium 1990, Altrincham
Reliability 1991, London

(ii) Attendance at the following seminars:

Software testing seminar London, 1987
Euredata seminar on the use of RAM data, Stockholm, Sweden 1988
Software Quality Management seminar London 1988
Euredata seminar on Process Safety, Hovik, Norway 1989
Presentation of Ansell and Phillips Paper at Royal Statistical Society 1989 (Comments on paper are within this thesis)
Computer aided Acquisition and Logistic Support seminar 1989
Concurrent Engineering seminar 1990

(iii) Attendance at the following meetings:

19th Euredata Assembly meeting at Stockholm Sweden 1988
20th Euredata Assembly meeting at Siena, Italy 1989
21st Euredata Assembly meeting at Hovik, Norway 1989
22nd Euredata Assembly meeting at ISPRA, Italy 1990 (Comments on data analysis benchmark exercise are within this thesis)

(iv) Participation in a BRITE proposal for Optimised FMEA at:

CETIM, Senlis, France
Technische Hochschule, Darmstadt, West Germany (twice)
Heckler and Koch GmbH, Stuttgart, West Germany
Prosyst, Valenciennes, France

(v) Meetings with Alvey Software Reliability Modelling Project collaborators at

The Centre for Software Reliability City University, London
Logica Space and Defence Systems, London
SRD (UKAEA), Risley, Warrington
STC, Newcastle under Lyme

Alvey Task 9 meetings (Secretary) at Nottingham Polytechnic

Meeting at Rolls Royce and Associates Derby 1991 on Fault Tree Software

(vi) Appropriate reading

Throughout the period of registration the author has undertaken part time and full time lecturing in statistics for B.Sc Urban Estate Surveying, B.Eng Mechanical Engineering, B.Sc Industrial Studies, HND/HNC in Computer Studies, HND in Civil Engineering, B.Sc Occupational Safety and Health and B.A in International Hospitality Management.

The author taught various aspects of reliability to the B.Eng Mechanical Engineering, B.Eng and M.Eng final year Manufacturing Engineering degree students; Rolls Royce, Derby; Lucas, Belfast and on the Postgraduate Diploma in Reliability within the Polytechnic.

Acknowledgements

Work for this thesis has been completed at Nottingham Polytechnic. The author would like to thank the institution for the resources provided and its assistance.

The author would like to thank the collaborating establishments during the Alvey Software Reliability Modelling Project including British Aerospace, AEA Associates (formerly UKAEA) Logica, STC, City University and GEC-Marconi.

Thanks are due to Peter Dixon with his help with multivariate techniques and to Graham Dawson for help with the data sorting programs and to all other members of staff for their aid in completing this thesis.

The author is grateful to his supervisors Professor A. Bendell, Dr. D. Wightman and Dr. N. Davies for their help and guidance and especially to Dr. D. Wightman for his software implementation.

Declaration

During the period of registration for M.Phil/Ph.D. the author has not been registered as a candidate for any other award of the CNAA, nor any award of a University.

CONTENTS

CHAPTER 1. INTRODUCTION 1

1.1. SOFTWARE RELIABILITY PREDICTION 3

1.2. THE STATE OF THE ART OF SOFTWARE RELIABILITY 6
.....

1.3. THE DIFFERENCE BETWEEN SOFTWARE AND HARDWARE 7
RELIABILITY .....

1.4. RELIABILITY THEORY 10

1.4.1. REPAIRABLE SYSTEMS RELIABILITY 15

1.4.2. NON-HOMOGENEOUS POISSON PROCESSES 17

1.4.3. RELATIONSHIP BETWEEN INTENSITY FUNCTION 18
AND HAZARD RATE .....

1.4.4. MULTIVARIATE TECHNIQUES 19

1.5. REQUIREMENTS AND METHOD OF ANALYSIS 20

1.5.1. DESCRIPTION OF THE SOFTWARE SYSTEM 23

1.5.2. CHECK FOR MISSING OR CORRUPT DATA 24

1.5.3. DISCRIMINANT ANALYSIS 24

1.5.4. MULTIVARIATE ANALYSES FOR DETERMINING 25
STRUCTURE .....

1.5.5. EXPLORATORY DATA ANALYSIS (EDA) 25

1.5.6. MODELLING 26

1.5.7. CONCLUSIONS AND RECOMMENDATIONS 26

1.5.8. FEEDBACK, GUIDELINES AND IMPROVED 26
PROCEDURES .....

1.6. TABLE OF ANALYSIS METHODS 27

CHAPTER 2. THE ALVEY SOFTWARE RELIABILITY MODELLING PROJECT 29

2.1. WORK CARRIED OUT IN TASK AREA 3 31

2.1.1. SUB-TASK 3.1. INVESTIGATION OF GROUNDS FOR POTENTIAL MODELS 31

2.1.2. SUB-TASK 3.2. IDENTIFICATION OF NATURE OF DEVELOPMENT, USE SCENARIO AND EXPLANATORY VARIABLES 31

2.1.3. SUB-TASK 3.3. DEVELOPMENT / SPECIALISATION OF MODELS 32

2.1.4. SUB-TASK 3.4. PROTOTYPE SOFTWARE DEVELOPMENT 33

2.1.5. SUB-TASK 3.5. GUIDELINES FOR RELIABILITY AND SAFETY ASSESSMENT OF SOFTWARE (GRASS) 34

2.2. WORK CARRIED OUT IN TASK AREA 4 34

2.2.1. SUB-TASK 4.1. OTHER STOCHASTIC POINT PROCESSES AND NON-PARAMETRIC PROCEDURES 34

2.2.2. SUB-TASK 4.2. ENTROPY APPROACH 34

2.2.3. SUB-TASK 4.3. TIME SERIES METHODS 35

2.2.4. SUB-TASK 4.4. MULTIVARIATE TECHNIQUES . 35

2.2.5. SUB-TASK 4.5. EXPLORATORY DATA ANALYSIS (EDA) APPROACH 37

2.3. SOFTWARE DATA COLLECTION 37

2.3.1. APPROACH OF WALSTON AND FELIX (1977) .. 38

2.3.2. APPROACH OF MUSA (1980) 38

2.3.3. APPROACH OF NAGEL AND SKRIVAN (1981) .. 39

2.3.4. APPROACH OF KITCHENHAM (1984) 39

2.3.5. APPROACH OF MARTINI ET AL (1990,1991) . 40

2.4. ALVEY SRM PROJECT TASK 9 ACTIVITIES 40

2.4.1. DATA MANAGEMENT 41

2.4.2. PROGRESS OF DATA SETS: ACQUISITION, COLLECTION, TRANSFER TO DATABASE AND PRELIMINARY ANALYSIS 42

2.4.3. ESTABLISHMENT OF THE COMPUTER NETWORK AND CREATION OF THE SOFTWARE RELIABILITY RELATIONAL DATABASE AND ASSOCIATED DOCUMENTATION 45

2.4.4. PROBLEMS ENCOUNTERED DURING THE PROJECT 46

2.4.5. CONCLUSIONS AND RECOMMENDATIONS FOR AREAS OF RESEARCH BASED ON THE TYPES OF DATA COLLECTED 46

CHAPTER 3. EXPLORATORY ANALYSIS OF ALVEY DATA SET NUMBER 3 49

3.1. SOFTWARE FOR DATA CHECKING 49

3.1.1. DESCRIPTION OF THE DATA 50

3.2. CHECKING FOR INCORRECT DATA 50

3.3. OBSERVATIONS OF THE DATA SET 52

3.3.1. CONCLUSION 61

3.4. SIMPLE PLOTS OF THE DATA 61

3.5. CORRELATION AND REGRESSION USED FOR EDA 65

3.6. RELATIONSHIPS BETWEEN FAULT COUNT AND SOFTWARE ATTRIBUTES 68

3.7. SUMMARY 71

CHAPTER 4. TIME SERIES ANALYSIS 72

4.1. STATE OF THE ART OF TIME SERIES 72

4.1.1. BAYESIAN ANALYSIS OF TIME SERIES 72

4.1.2. TIME SERIES AND PROPORTIONAL HAZARDS MODELLING 73

4.2. ANALYSIS OF FAILURE COUNTS 74

4.2.1. A THEORY OF SOFTWARE DEVELOPMENT 75

4.2.2. DERIVATION OF THE BOX-JENKINS APPROACH 78

4.2.3. ANALYSIS OF SOFTWARE FAILURE COUNTS PER DAY FOR ALVEY DATA SET NUMBER 3 79

4.2.4. EXPLORATORY APPROACH 81

4.2.5. FIVE DAY SEASONAL TIME SERIES 85

4.3. TIME SERIES OF LOGGED DATA 90

4.3.1. RESIDUAL PLOTS 93

4.3.2. FORECASTS FOR LOGGED DATA 98

4.4. ANALYSIS OF THE LIVE PHASE 102

5. SOFTWARE RELIABILITY THEORY 104

5.1. THE NON-HOMOGENEOUS POISSON PROCESS 106

5.2. THE RELATIONSHIP BETWEEN NHPP'S TO PHM 108

5.3. DESCRIPTION OF MODELS FOR SOFTWARE RELIABILITY GROWTH 109

5.3.1. BINOMIAL TYPE MODELS OF THE EXPONENTIAL CLASS 109

5.3.2. THE WEIBULL MODEL 111

5.3.3. THE LITTLEWOOD NHPP 111

5.3.4. POISSON TYPE MODELS OF THE EXPONENTIAL CLASS 111

5.3.5. THE MUSA BASIC EXECUTION TIME MODEL ... 113

5.3.6. THE S-SHAPED INFLECTION MODEL 115

5.3.7. THE GOEL-OKUMOTO MODEL 117

5.3.8. THE S-SHAPED MODEL 117

5.4. DESCRIPTION OF MODELS FOR HARDWARE AND SOFTWARE RELIABILITY GROWTH 118

5.4.1. THE DUANE MODEL 118

5.4.2. THE COX-LEWIS MODEL 119

5.4.3. THE IBM MODEL 120

5.4.4. THE BOUNDED INTENSITY MODEL 121

5.4.5. THE LOGARITHMIC MODEL 121

5.4.6. THE SQUARE ROOT MODEL 122

5.5. APPLICATION OF PHM FORMULATIONS TO SOFTWARE RELIABILITY DATA 122

CHAPTER 6. PROPORTIONAL HAZARDS MODELLING 126

6.1. ANALYSIS OF FAILURE COUNTS 128

6.2. ANALYSIS OF FAILURE TIMES USING PHM 132

6.3. ANALYSIS OF THE TWELVE LEAST RELIABLE SOURCES AS A GROUP 136

6.3.1. HAZARD ANALYSIS OF FORMULATION (1) 138

6.3.2. HAZARD ANALYSIS OF FORMULATION (2) 143

6.3.3. HAZARD ANALYSIS OF FORMULATION (3) 145

6.3.4. HAZARD ANALYSIS OF FORMULATION (4) 148

6.4. STRATIFICATION OF WAITING TIMES TO FAILURE OF SOURCES 150

6.4.1. HAZARD ANALYSIS FOR THE COVARIATE $i-1$ PHM FORMULATION 155

6.4.2. HAZARD ANALYSIS FOR THE COVARIATE t_{i-1} PHM FORMULATION 156

6.4.3. HAZARD ANALYSIS FOR THE COVARIATE $\log(i)$ PHM FORMULATION 158

6.4.4. WEIBULL HAZARD PLOTTING 161

6.4.5. QUADRATIC HAZARD FITTING 164

6.5. PHM ANALYSES USING SOFTWARE ATTRIBUTES 166

6.6. SUMMARY 167

CHAPTER 7. MULTIVARIATE TECHNIQUES 168

7.1. DISCRIMINANT ANALYSIS 169

7.2. PRINCIPAL COMPONENTS ANALYSIS (PCA) 170

7.3. LOG-LINEAR MODELS 173

7.3.1. CONCLUSIONS OF LOG-LINEAR MODELLING ... 175

7.4. GENERALISED LINEAR MODELLING 177

7.4.1. DESCRIPTION OF MODELS 178

7.4.2. DATA PLOTS AND ANALYSIS 181

7.5. RELATIONSHIP BETWEEN RESPONSE MODELS AND PROPORTIONAL INTENSITY MODELS 188

7.6. PROPORTIONAL INTENSITY MODELLING 189

7.6.1. RELATIONSHIP TO PROPORTIONAL HAZARDS MODELLING 189

7.6.2. MODEL FORMULATION 190

7.6.3. COVARIATES 192

7.6.4. ANALYSIS OF THE TWELVE LEAST RELIABLE SOURCES 192

7.6.5. ANALYSIS OF THE BASELINE INTENSITY 195

7.6.6. FURTHER PROPORTIONAL INTENSITY MODELLING OF THE TWELVE SOURCES 199

CHAPTER 8. OTHER WORK 200

8.1. COMMENTS ON THE EUREDATA BENCHMARK EXERCISE . 200

8.2. COMMENTS ON THE ANSELL AND PHILLIPS 1989 PAPER 205

CHAPTER 9. CONCLUSIONS 208

9.1. CONTRIBUTIONS TO KNOWLEDGE AND REVIEW OF THESIS 208

9.2. TYPES OF ANALYSES UNDERTAKEN 209

9.3. FURTHER WORK 214

REFERENCES 216

INDEX OF APPENDICES

APPENDIX 1

Bendell, A., McCollin, C., Wightman, D.W., Linkman, S. and Carn, R. (1988). Software Reliability Data Collection, Problems and Possibilities. Proceedings of the 20th Euredata Conference, Siena.

APPENDIX 2

McCollin, C., Bendell, A. and Wightman D.W. (1989). Effects of Explanatory Factors on Software Reliability. Proceedings of Reliability 1989 Vol 2, pp 5Ba/1/1-11.

APPENDIX 3

McCollin, C., Wightman, D.W., Dixon, P. and Davies, N. (1990). Some Results of the Alvey Software Reliability Modelling Project. Proceedings of the SARSS. Altrincham, 1990.

APPENDIX 4

Wightman, D.W., McCollin, C. and Dixon, P. (1991). Recent Applications of Some Statistical Techniques to Software Reliability Data. Proceedings of the 1991 Reliability Conference. Reliability 91. Elsevier Science Publishers.

INDEX OF FIGURES

FIGURE 1.1. PLOT OF REPAIRABLE SYSTEMS 15

FIGURE 3.1. ALVEY DATA SET NUMBER 3 FAILURE AND REPAIR RECORD INFORMATION 55

FIGURE 3.2. PLOT OF FAILURE COUNT AGAINST TIME IN DAYS 58

FIGURE 3.3. PLOT OF NUMBER OF FAILURES PER DAY AGAINST DAYS 62

FIGURE 3.4. PLOT OF CUMULATIVE TTF AGAINST SOURCE NUMBER 63

FIGURE 3.5. PLOT OF THE PROPORTION OF FAILURES AGAINST TTF FOR EACH SOURCE TYPE 65

FIGURE 3.6. PLOT OF NUMBER OF FAULTS AGAINST SOURCE LANGUAGE AND SIZE 69

FIGURE 3.7. PLOT OF NUMBER OF FAULTS AGAINST SOURCE TYPE AND SIZE 70

FIGURE 4.1. EXPONENTIAL PLOT OF DATA AS PER LONGBOTTOM REFERENCE 76

FIGURE 4.2. BOX AND WHISKER PLOT OF ALVEY 3 WEEKDAY FAILURE COUNT DATA 83

FIGURE 4.3. PLOT OF THE NORMALISED RESIDUALS AGAINST WEEKDAY 93

FIGURE 4.4. TIME SERIES PLOT OF THE RESIDUALS 94

FIGURE 4.5. PLOT OF THE ORIGINAL DATA AGAINST THE RESIDUALS 96

FIGURE 4.6. NORMAL PROBABILITY PLOT OF THE RESIDUALS 97

FIGURE 4.7. BARCHART OF THE RESIDUALS 98

FIGURE 6.1. PLOT OF ESTIMATED CUMULATIVE HAZARD AGAINST TIME SINCE LAST FAILURE 139

FIGURE 6.2. PLOT OF ESTIMATED LOG CUMULATIVE HAZARD AGAINST LOG TIME SINCE LAST FAILURE 141

FIGURE 6.3. PLOT OF CUMULATIVE HAZARD AGAINST Yi FOR FORMULATION (2) 143

FIGURE 6.4. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR FORMULATION (2) 144

FIGURE 6.5. PLOT OF CUMULATIVE HAZARD AGAINST TIME FOR FORMULATION (3) 146

FIGURE 6.6. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR FORMULATION (3) 147

FIGURE 6.7. PLOT OF CUMULATIVE HAZARD AGAINST TIME FOR FORMULATION (4) 148

FIGURE 6.8. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR FORMULATION (4) 149

FIGURE 6.9. PLOT OF CUMULATIVE HAZARD AGAINST TIME FOR SOURCE NUMBER 6 AND COVARIATE i-1 156

FIGURE 6.10. PLOT OF CUMULATIVE HAZARD AGAINST Yi FOR SOURCE NUMBER 12 158

FIGURE 6.11. PLOT OF CUMULATIVE HAZARD AGAINST TIME FOR SOURCE NUMBER 9 160

FIGURE 6.12. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR SOURCE NUMBER 6 AND COVARIATE ti-1 .. 162

FIGURE 6.13. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR SOURCE NUMBER 12 AND COVARIATE i-1 .. 163

FIGURE 6.14. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR SOURCE NUMBER 9 AND COVARIATE log(i) 164

FIGURE 6.15. QUADRATIC HAZARD PLOT FOR SOURCE NUMBER 6 AND COVARIATE log(i) 165

FIGURE 7.1. ILLUSTRATIVE PLOT OF g(p) AGAINST x WITH MODEL (1) FITTED 179

FIGURE 7.2. ILLUSTRATIVE PLOT OF g(p) AGAINST x WITH MODEL (2) FITTED 180

FIGURE 7.3. ILLUSTRATIVE PLOT OF g(p) AGAINST x WITH MODEL (3) FITTED 181

FIGURE 7.4. PLOT OF THE LOGIT PROPORTION OF FAILURES AGAINST LOG CUMULATIVE TIME TO FAILURE .. 182

FIGURE 7.5. PLOT OF THE LOGIT CUMULATIVE PROPORTION OF FAILURES AGAINST LOG CUMULATIVE TIME FOR EACH SOURCE TYPE 184

FIGURE 7.6. PLOT OF FIRST SIX SOURCES: FAILURE NUMBER AGAINST TIME IN DAYS 194

FIGURE 7.7. PLOT OF LAST SIX SOURCES: FAILURE NUMBER AGAINST TIME IN DAYS 195

FIGURE 7.8. PLOT OF OBSERVED AND EXPECTED CUMULATIVE INTENSITY AGAINST TIME 196

INDEX OF TABLES

TABLE 1.1. TABLE OF DELIVERED ALVEY DATA SET NUMBER 3 FILES 22

TABLE 1.2. TABLE OF ANALYSIS METHODS 27

TABLE 2.1. TABLE OF STATISTICAL METHODS USED IN STATED REFERENCES 40

TABLE 2.2. TABLE OF ALVEY DATA SETS (NOTTINGHAM POLYTECHNIC INPUT) 44

TABLE 3.1. NUMBER OF PRODUCT VERSION FAILURES 56

TABLE 3.2. NUMBER OF TIMES SOURCES REPAIRED 59

TABLE 3.3. NUMBER OF TIMES SOURCE VERSIONS REPAIRED 59

TABLE 3.4. NUMBER OF FAULTS AGAINST REPAIRS PER FAULT 60

TABLE 3.5. CORRELATION AND REGRESSION INFORMATION OF ALVEY DATA SET NUMBER 3 67

TABLE 4.1. TABLE OF THE ARIMA (2 0 0)(0 1 1)5 MODEL VALUES 86

TABLE 4.2. TABLE OF THE ARIMA (1 0 1)(0 1 1)5 MODEL VALUES 87

TABLE 4.3. FORECASTS FOR THE ARIMA (2 0 0)(0 1 1)5 MODEL 89

TABLE 4.4. FORECASTS FOR THE ARIMA (1 0 1)(0 1 1)5 MODEL 89

TABLE 4.5. TABLE OF ARIMA (0 1 1)(0 1 1)5 MODEL ESTIMATES 91

TABLE 4.6. TABLE OF BOX-PIERCE STATISTICS 92

TABLE 4.7. FORECASTS FROM PERIOD 100 99

TABLE 4.8. FORECASTS FROM PERIOD 105 100

TABLE 4.9. FORECASTS FROM PERIOD 110 101

TABLE 5.1 TABLE OF NON-HOMOGENEOUS POISSON PROCESSES 107

TABLE 5.2. TABLE OF PHM FORMULATIONS 125

TABLE 6.1. PHM RESULTS FOR FAILURE COUNTS 130

TABLE 6.2. PHM RESULTS FOR TIMES SINCE LAST FAILURE 137

TABLE 6.3. COVARIATE INFORMATION OF WITHIN SOURCES VARIATION 152

TABLE 6.4. TABLE OF COEFFICIENTS OF DETERMINATION FOR THE FOUR HAZARD MODELS FOR SOURCE NUMBER 6 ... 154

TABLE 6.5. TABLE OF COEFFICIENTS OF DETERMINATION FOR THE FOUR HAZARD MODELS FOR SOURCE NUMBER 9 ... 154

TABLE 6.6. TABLE OF COEFFICIENTS OF DETERMINATION FOR THE FOUR HAZARD MODELS FOR SOURCE NUMBER 12 .. 154

TABLE 6.7. TABLE OF ESTIMATES FOR THE GOEL-OKUMOTO MODEL 157

TABLE 6.8. TABLE OF TWO PARAMETER WEIBULL ESTIMATES FOR THE NINE MODELS 161

TABLE 7.1. TABLE OF EIGENANALYSIS RESULTS FOR PCA 172

TABLE 7.2. TABLE OF LOG-LINEAR MODELLING RESULTS . 175

TABLE 7.3. GOODNESS OF FIT ESTIMATES FOR THE GLIM ANALYSIS MODELS..... 186

TABLE 7.4. ESTIMATES OF MODEL PARAMETERS FOR THE GLIM ANALYSIS..... 187

TABLE 7.5. TABLE OF MEAN VALUES E(N(t)) 197

TABLE 7.6. TABLE OF TIMES TO REACH A GIVEN INTENSITY 198

TABLE 8.1 COMPARISON OF ANALYSES FOR THE EUREDATA BENCHMARK EXERCISE: DEFINITIONS OF POPULATION AND FAILURE 202

TABLE 8.2 COMPARISON OF ANALYSES FOR THE EUREDATA BENCHMARK EXERCISE: QUANTITATIVE ANALYSIS 203

TABLE 9.1 TABLE OF COMPARISON OF METHODS 211

1 INTRODUCTION

In recent years, the scope and complexity of software has grown enormously and many problems relating to unreliable software have been highlighted in the media. Examples of some catastrophic software failures include the 1991 Boeing 767 Lauda Air disaster in Thailand in which 223 people were killed (ref. Sunday Independent, 21 July 1991) in which the engine control system switched on the reverse thrust system while the engine was on maximum power; the Colorado river flooding in 1983 was due to faulty weather data and/or a faulty model in which too much water was kept dammed prior to spring thaws; a Japanese mechanic was killed by a malfunctioning Kawasaki robot (ref. Electronic Engineering Times, 21 December 1981) and a woman killed her daughter and tried to kill her son and herself after a computer error led to a false report of their all having an incurable disease (ref. IEEE AES Magazine, July 1985).

The problems of creating error free software has been addressed by a number of European and North American software research initiatives including STARTS (Software Tools for Application to Real Time Systems), ESPRIT, EWICS TC7 and Alvey however there is no development method which can guarantee a completely reliable (software) system. The need to assess the reliability of a (hardware and software) system arises typically from a customer requirement, to understand the safety implications, to predict the optimal time to software release or for safety critical certification; the Health and Safety Executive are 'putting in systems whose specification states that they should not fail more often than once in 100 million reactor years' (New Scientist, 1 April 1989)).

The assessment of reliability for a hardware system is well developed, however there has been very little work carried out creating a reliability assessment method for software similar to U.S. Mil-Handbook 217 for hardware systems.

The reliability goal of a hardware/software system is usually defined in a system specification which is tendered to a prospective supplier of the system. A customer may specify a reliability requirement such as a specified reliability of 0.9 or a Mean Time Between Failure (MTBF) of 1000 hours. This figure is typically complied with to the customer's satisfaction by either a MIL-HDBK-217 reliability prediction or a reliability demonstration test.

The main reliability activities in a hardware development project (to achieve a reliability goal) are:

to determine the types of failure that may occur (by analysis) and ensure that the most critical are either eliminated by design or have adequate contingency on failure (e.g., by redundancy).

to identify failure by some form of development testing and then subsequently design out the failure mode.

to demonstrate that the reliability of the built system satisfies the customer requirement.

to predict the reliability of the system before it is built.

Within the hardware field, there are a number of established techniques for dealing with each of the above activities. These are Failure Modes and Effects Analysis (FMEA), Reliability Development Testing (RDT), Reliability Demonstration and Reliability Prediction. These techniques are described in O' Connor (1982) among others. Software testing, inspection and fault finding are covered in a number of texts (Myers (1976), Anderson and Randell (1979).

Reliability prediction is more relevant to software rather than hardware reliability. This is because software information (such as type of fault, input and output parameters, etc) is collected during development which aids the systems programmers and analysts in the location and subsequent removal of software errors. Using this data, the system reliability may be estimated before delivery. Also, since faults in the software may only be revealed under certain input conditions, there may be a need to know approximately how many faults are left undiscovered in a piece of software code on delivery to a customer.

1.1 SOFTWARE RELIABILITY PREDICTION

Many software reliability prediction models have been developed to try to determine the remaining number of faults and these are discussed in a later section. Here, an approach to modelling similar to hardware reliability modelling is considered.

Reliability prediction in the hardware field such as by the use of MIL-HDBK-217 is well developed in areas such as the aerospace and the nuclear industry. Reliability prediction is used to compare alternative designs and

provide figures for life cycle costing and spares provisioning. It is not a technique for predicting service-use hazard rate.

Hardware component hazard-rates or failure-rates are usually assumed to be constant so that times to failure follow the exponential distribution and hazard-rates may be added together to provide a system hazard-rate figure. The best known database of failure rates is US MIL-HDBK-217 which contains equations for failure-rates for most electrical and electronic devices.

The physical model employed in this Standard which relates part base-hazard-rate to temperature stress is the Aarhenius equation. A hazard-rate for most operating environments, quality factors and component stresses may be calculated by multiplying part base-hazard-rate by multipliers for each of these factors.

The document's main advantage is that it relates component hazard rate to explanatory factors. Hence by choosing components which have a low valued explanatory factor (e.g. voltage stress, current rating), the predicted hazard rate is reduced.

There has been criticism of the MIL-HDBK-217 (O'Connor (1991)) regarding among other things the inappropriateness of the exponential distribution for the statistical model. For instance, the hazard rate figures for lasers are based on the wearout mechanisms of the laser cavities so they would not be expected to fail randomly. Thus the exponential distribution is not applicable in this case. However, it has been shown by Landers and Kolarik (1986) that the hazard rates are a special case of a proportional hazards

model. The assumption of the exponential distribution is more for mathematical convenience (the hazard rates may be added) than the exact physical representation of the component failure mechanism.

Software reliability is harder to model as a physical process. However, modelling software reliability is facilitated, as actual data is usually available from a specific project. Guidelines for software reliability prediction have been attempted previously by Sukert (1980), however, the methods outlined do not seem to have been pursued.

A prediction method should take into account external factors as well as the attributes of the software and the wide variety of software reliability models available. For software, the explanatory factors are more diverse and problems can arise in estimating these factors due to external influences such as data collection methodology, quality of data and inappropriate statistical models. It will be shown that proportional hazards and proportional intensity modelling are adequate structures for software reliability modelling.

This thesis provides a framework for analysing software reliability data It also provides an objective approach to the use, or rejection, of well known software reliability growth models for modelling reliability data sets. A number of software failure data sets were collected during the Alvey Software Reliability Modelling (SRM) project at Nottingham, the data collection exercise being described in chapter 2. One of these data sets is analysed in detail to show the approach to statistical analysis. The data to which this is applied is usually collected within a software

failure data collection scheme. The analyses carried out have given some insight into the final system reliability, an optimal time to release as well as a description of the failure behaviour during development.

1.2 THE STATE OF THE ART OF SOFTWARE RELIABILITY

A problem in software reliability modelling is that, since the early seventies, a large number of models have been specified (see chapter 5) and the term 'Model Wars' has been coined where papers have been written to describe the relative merits of each model. One solution to this problem is to classify the models into a group, for example as exponential order statistic (EOS) models, etc. The classification of EOS models is discussed in Mellor (1987).

Dale (1991) states that a natural approach to predict the reliability at various stages of development is to develop a model which incorporates explanatory variables which explain 'variation in terms of features of the software or its development'. Dale suggests generalised linear modelling as a possibility. Features of the software and it's environment such as source size and day of failure detection have been incorporated into a time series structure and generalised linear models within this thesis.

Wightman in Mellor and Bendell (1986) describe the use of proportional hazards modelling among others and this approach may be used as an exploratory tool to highlight data structure. They also state that Exploratory Data Analysis, (EDA) may be used as an aid in determining structure in the data. The proportional hazards formulations in chapter 5 provide a diagnostic tool to aid the selection of the most appropriate software reliability

model for a given set of data and has the advantage over the previous groupings of models in being able to reject specific models which do not fit the data.

Dale (1991) states 'The problem of the choice of possible explanatory variables is very wide and there is very little science which can be drawn upon to aid an intelligent initial selection'. An added advantage of using proportional hazards modelling is that extra explanatory information may be included. If data is collected by a standard data collection mechanism, then explanatory variables which are relatively easy to collect such as source type or size may be used within a statistical analysis which may show at an early stage of development which sources are unreliable. Fagan (1976) suggests that these unreliable sources should be monitored and reviewed at an early stage for possible redevelopment. The application of these proportional hazards formulations to the least reliable sources of Alvey data set 3 is described in chapter 6.

1.3 THE DIFFERENCE BETWEEN SOFTWARE AND HARDWARE RELIABILITY

The following software terms are taken from Mellor (1986) and are used throughout the thesis.

In hardware, the lowest level of assembly is a **part** such as a transistor or a coil. A **subassembly** is made up of parts and is tested before it is installed into an assembly. A number of subassemblies may be connected together in a functional block called a **system** and this also undergoes a test to an acceptance procedure to satisfy a customer that the system will work.

In software, the lowest level of assembly is a line or statement of the software **code.** These instructions are translated by a compiler into a binary code which a computer processor can execute, for example to transfer values within the code from one register to another. The code is written in lines to form a **source** which carries out a specific task (e.g. to calculate a sample mean for a set of data). A source may also be a data file. A number of sources connected together form a **module.** The collection of modules when executed together is called a **system** or a **product.**

In both hardware and software, the customer defines how the system should operate and where and how it interfaces with the external environment such as other systems and/or the operational and environmental conditions. These conditions are stated in the **system specification.**

A **fault** is an error in the code (or specification) which may cause the product to fail (a failure) or would have caused a failure if it had not been found before the software execution. A **failure** is defined as the fault which makes the software crash on software execution.

Reliability may be defined within a system specification as follows.

The reliability of a system is the probability that it will perform as required by the correct specification for a given period of time under the given operational and environmental conditions. This definition is similar to the one in Anderson and Randell (1979).

The contributory causes for how a hardware or software system fails may now be derived from the above definitions. The failure of a system may be due to:

1. a fault in writing, understanding and interpreting a specification.
2. a fault in the interface between modules or assemblies
3. a fault in testing the code or part incorrectly
4. a fault in the code or part.

For items 1 and 2 above, the time element of the reliability definition is difficult to quantify since no test time will have been accumulated. For item 3, defining the test correctly will certainly influence the system reliability for software and hardware. The effect of different tests on system reliability may be explored statistically by incorporating the information within a model (such as proportional hazards) but very little work has been carried out in this area (see Nagel and Skrivan (1981) as an example of controlled experimentation). A description of the software lifecycle incorporating planning and testing may be found in the references by Rook (1990), Myers (1976) or Conte, Dunsmore and Shen (1986).

A fault at the lowest level of assembly is dealt with in one of two different ways depending on whether the system is undergoing development or is in full production. During development, a fault can be designed out of a system, the reliability growing as successive design faults are removed. In service use, a hardware part which has failed is replaced by an equivalent unfailed part, however, for software if the fault is due to incorrect code, the code may be rewritten (repaired). Thus, assuming that environmental conditions remain as per system specification

and that repairs to faults do not increase the initial fault stock, the main difference between software and hardware reliability is that eventually all faults in the software may be found and removed whereas, in a hardware system, parts will continually degrade and wearout. Thus, to model software reliability requires the statistical model to take this initial number of faults into account. General reliability theory for hardware is developed below. Extensions to this theory to software are described in chapter 5.

1.4 RELIABILITY THEORY

We define the time to failure (TTF) as the random variable $\{X : x \geq 0\}$. Assuming that TTF is continuous, F(x) is the distribution function of x, which is the probability of a value of the time to failure X being less than or equal to some value x (cumulative density function). The reliability R(x) is the probability that there is no failure before x so that

$$R(x) = 1 - F(x).$$

Also, $F(x) = \int_0^x f(t)dt$ where f(x) is the probability density function (pdf) of x.

The hazard rate, force of mortality or instantaneous failure rate h(x) is defined as

h(x) dx = probability of failure in the interval (x, x+dx) given survival to time x.

It may also be written as $h(x) = \frac{f(x)}{R(x)}$.

The reliability may be derived from the three equations above as

$$R(x) = e^{-\int_0^x h(t)dt} \qquad -(1).$$

Certain hazard functions have gained popularity in modelling hardware reliability over the last thirty years. In software reliability theory, various statistical models have been derived which may account for the failure process of the software but these models have not been directly related to these well-known hazard functions. The following distribution theory will be applicable in chapter 5 when deriving the well-known software reliability models in hazard terms.

Following Thompson (1988), there are three interpretations which may be applied to interarrival times. These are the length of the gap between the arrivals which may be designated X_t, the separation of arrivals near a fixed time $X_{N(t)+1} = Y_{N(t)+1} - Y_{N(t)}$ where N(t) is the number of arrivals in the interval 0 to t and the forward waiting time from t to the next arrival $W_t = Y_{N(t)+1} - t$.

Various results have been derived for the gap lengths however they are not simple. For example, Parzen (1962) derived the reliability function for the kth interval (i.e. the time from the (k-1)st event to the kth event) given by

$$R(l_k) = \int_0^{\infty} e^{-M(l-s)} \lambda(s) \frac{M(s)^{k-2}}{(k-2)!} ds \qquad k \geq 2$$

and the pdf of the kth gap length is

$$f_k(l) = e^{-M(l)} \frac{M(l)^{k-1}}{(k-1)!} \lambda(l).$$

The forward waiting time is the time to wait for an event to happen. The backward waiting time is $B_t = t - Y_{N(t)}$ so that $X_{N(t)+1} = W_t + B_t$.

On testing a source code, the source will run in a certain execution time given no external influences, since each item of code takes a precise time for the computer microprocessor to process. Assume that there are a number of faults in the code. The waiting time to failure for each fault may be designated $x_1, x_2, \ldots, x_n$. Then the waiting time to the first failure will be the shortest time to reach any fault. The distribution of the smallest time will have the form one of the three smallest extreme value distributions. The next waiting time to failure will also follow a smallest extreme value distribution but will be modulated in some way by it being the second waiting time. The hazard rates for each of the $x_3, x_4, \ldots, x_n$ may then be determined in a similar way. This is described in more detail in chapter 5.

By contrast, the Weibull distribution (a distribution of large extremes) is typically used to model the times to failure of hardware systems and components.

From Gumbel (1958), for smallest extremes, the three distribution functions are

the Frechet distribution given by

$$F(x) = 1 - e^{-\left(-\frac{x-\gamma}{\alpha}\right)^{-\beta}} \qquad -\infty < x < \gamma, \alpha, \beta > 0$$

the Weibull distribution given by

$$F(x) = 1 - e^{-\left(\frac{x-\gamma}{\alpha}\right)^{\beta}} \qquad \gamma \leq x < \infty, \alpha, \beta > 0$$

and the Gumbel distribution given by

$$P(X \leq x) = F(x) = 1 - e^{-e^{\frac{(x-\gamma)}{\alpha}}} \qquad -\infty < x < \infty, \alpha > 0.$$

As failure times are non-negative random variables, then by using the transformations

$$e^{bT} - 1 = e^{-\frac{X}{\alpha}}, \quad \alpha = e^{\frac{\gamma}{\alpha}}$$

in the standard form of the Gumbel distribution above, then

$$P(T \leq t) = 1 - e^{-\alpha(e^{bt}-1)}, \quad 0 < t < \infty.$$

The hazard rate for this distribution is

$h(t) = \alpha b e^{bt} \quad \alpha, b > 0.$ (see the Cox-Lewis intensity in the proportional hazards formulation of chapter 5.4.2).

Alternatively, by using the transformations

$$1 - e^{-bT} = e^{\frac{X}{\alpha}}, \quad \alpha = e^{-\frac{\gamma}{\alpha}}$$

in the standard form of the Gumbel distribution above, then

$P(T \leq t) = 1 - e^{-a(1-e^{-bt})}, \quad 0 < t < \infty$.

The hazard rate for this distribution is $h(t) = abe^{-bt} \quad a, b > 0$. This hazard rate is used in chapter 5.3.4 on the Poisson type intensity models, chapter 5.3.5 on the Musa basic model, chapter 5.3.6 on the Ohba model, chapter 5.3.7 on the Goel-Okumoto model and chapter 5.3.8 on the S-shaped model.

The Gompertz hazard rate and Makeham's formula (Jordan (1975), Gross and Clark (1975)) are hazard rates used in actuarial science and take the form $h(t) = a\exp(bt) \quad a, \ b > 0$ and $h(t) = c + a\exp(bt) \quad a, \ b, \ c > 0$ or $h(t) = c + a\exp(bt) \ c > a\exp(bt)$ respectively. These are both special cases of the Gumbel hazard rate.

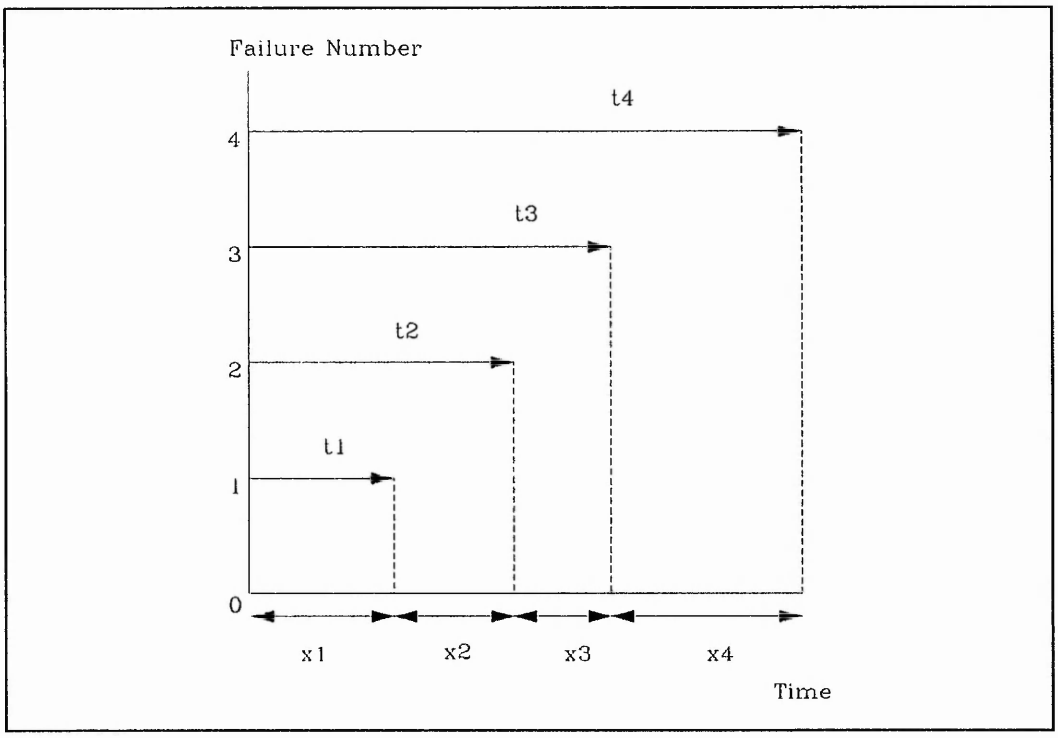
The two parameter Weibull distribution takes the hazard rate $h(t) = \frac{b}{a^b} t^{b-1}$ with cumulative hazard $H(t) = \left(\frac{t}{a}\right)^b$. When logs are taken of both sides of this formula, then $\log(H(t)) = b\log t - b\log a$, a straight line relationship, so that if the estimated cumulative hazard is plotted against time, the values of a and b may be calculated by linear regression modelling, (O'Connor (1982)). The Gumbel distribution may be derived by taking logs of a Weibull variate.

Gross and Clark (1975) describe a hazard rate of the form $h(t) = a + bt$ and Gaver and Acar (1979) describe the quadratic hazard rate $h(t) = a + bt + ct^2$.

1.4.1 REPAIRABLE SYSTEMS RELIABILITY

Repairable systems data may be represented in the following figure.

FIGURE 1.1. PLOT OF REPAIRABLE SYSTEMS



where
$x_1, x_2, \ldots, x_i$ are interarrival times of failures and 1,2....i are the failure counts and $t_1, t_2, \ldots$ are the cumulative times to failure. Censoring times are times (in a software application) when there may be a fix to a source code even though the particular source code had not failed. The total time of observation may be a terminated by a failure (failure termination) or a time (time termination).

Analysis of this data may be carried out by considering the number of failures within fixed time intervals (a time

series), proportions of items failed over the total number of items against time (a logistic model), the cumulative time to failure for a number of failed sources (a proportional intensity model) or time since last failure for the failed sources (a proportional hazards model).

Consider the cumulative times to failure. It is usual to express a series of cumulative failure times as a point process (see the next section). Some well documented point processes are the renewal process which is a sequence of random variables $\{Y_1, Y_2, \ldots\}$ of the form $Y_n = X_1 + \ldots + X_n$ where the interfailure times $\{X_1, X_2, \ldots\}$ are statistically independent with a common distribution F(X); the homogeneous Poisson process which is a renewal process with the common distribution being the exponential distribution and the non-homogeneous Poisson process (NHPP) where the distribution of interfailure times vary over time.

A description of specific NHPP's and their equivalent proportional hazard formulations are derived in chapter 5.

An alternative approach is to model the data as times since last failure within a proportional hazards framework using the supplementary information as explanatory variables. Since NHPP's are primarily used to model trend in the data, then the proportional hazards formulation should include covariate information to identify the types of trend which NHPP's highlight.

Another important factor to take into account when software reliability modelling is the usefulness of explanatory

information such as source size or type which may have an influence on the software reliability. The incorporation of this information is discussed at the end of this chapter.

1.4.2 NON-HOMOGENEOUS POISSON PROCESSES

The following derivations of point processes is taken from Parzen (1962) and more recently from McCollin (1980) and Thompson (1988).

A point process $\{N(t), t \geq 0\}$ is a collection of usually interrelated random variables. Suppose the points (usually called arrivals) represent times $t_1, t_2, \ldots, t_n$ at which failures have occurred where $0 < t_1 < t_2 < \ldots < t_n$. Then the random variables $x_1 = t_1 \quad x_2 = t_2 - t_1 \quad x_3 = t_3 - t_2 \quad \ldots \quad x_n = t_n - t_{n-1}$. Let N(t) be the number of arrivals that have occurred in the interval [0,t]. Then N(t+h)-N(t) assumes only non-negative integer values. For a point process, the expected number of failures in time t, E(N(t)), is written as M(t), which is also known as the mean value function. If this function can be differentiated, then $\mu(t) = M'(t)$ is called the arrival rate or the instantaneous rate of change of the expected number of arrivals with respect to time.

A point process has no simultaneous arrivals if each jump of N(t) is of unit magnitude. The intensity of a point process is the same as the arrival rate of the process if there are no simultaneous arrivals. The intensity may be written as $\lambda(t) = \lim_{h \to 0} \frac{(1 - P(N(t+h) - N(t) = 0))}{h}$. The form that the intensity function $\lambda(t)$ takes is not unique and some forms are described in chapter 5.

For a homogeneous point process, for any two points t>s>=0 and h>0, the random variables N(t)-N(s) and N(t+h)-N(s+h) are identically distributed and hence the intensity and arrival rate are constant for all time.

A homogeneous Poisson process is homogeneous point process such that

$P(N(t,t+x)=k)=\frac{M(x)^k e^{-M(x)}}{k!} \qquad k=0,1,2,.... \quad 0<M(x)<\infty$

where $M(x)=\lambda x \quad 0<\lambda<\infty$.

A Non-homogeneous Poisson process may take various formulations for M(x) and these are described in chapter 5.

1.4.3 RELATIONSHIP BETWEEN INTENSITY FUNCTION AND HAZARD RATE

The cumulative distribution function of the waiting time for a Poisson process is given in Thompson (1988) page 64 as

$$F_t(w)=Prob(W_t \le w)=Prob(N(t,t+w)\ge 1)$$

which is the same as

$$1-Prob(N(t,t+w)=0)=1-\exp^{-(M(t+w)-M(t))}.$$

Now the hazard rate of forward waiting time is the differential of the -log of the reliability function in (1) of chapter 1.4 so that

$$h_t(w)=\frac{d}{dw}(-\ln(R_t(w)))=\mu(t+w)$$

where $\mu(t)$ is the arrival rate of the process. Assuming

that there are no simultaneous arrivals, the arrival rate is the intensity function of a Poisson process and so

$$h_t(w) = \lambda(t + w).$$

This result is stated by Parzen (1962) and Musa (1987), derived by Thompson (1988) and will be used when re-formulating some well-known software reliability models into a proportional hazards structure.

1.4.4 MULTIVARIATE TECHNIQUES

It is beneficial to software project managers to collect diverse information to aid him/her in the best use of the resources at his/her disposal. Diverse information was collected during the data collection exercise of the Alvey project, so it is possible to consider the effect of such additional variables on the hazard rate of the software and the associated development environment. The class of statistical models which will allow incorporation of point process models with explanatory information is commonly known as generalised linear modelling. In this thesis, the proportional hazards models and log-linear models are considered. These will be derived in the specific forms required for analyses of software failure data collected during the Alvey Software Reliability Modelling Project and are discussed in more detail in later chapters. The relationship of proportional intensity modelling to generalised linear modelling is also investigated.

Multivariate techniques are not just used for modelling but also for structural simplification as in principal components analysis (see chapter 7), cluster analysis

(which classifies diverse data into groups), and discriminant analysis which may be used as an aid to determine missing data (also in chapter 7).

1.5 REQUIREMENTS AND METHOD OF ANALYSIS

The data collected from a data collection scheme may be analysed in a systematic way. The following analysis procedure is derived from similar data collection schemes within the process and nuclear industries described in Teichmann et al (1985), Samanta et al (1985) and Bendell (1988).

A number of data sets were collected during the Alvey SRM project to aid the selection of appropriate statistical models. Seven out of eight of the Nottingham collected data sets did not contain software execution times to failure which could provide the physical basis for certain software reliability models, (Jelinski-Moranda, Musa, etc). Various data came from installations of the software on different sites (data set 1) or was not coded sufficiently to determine if the software or associated hardware had failed (data sets 7 and 8). Among the collected data sets came failures recorded at a personal computer, service reports, development projects for new operating systems, computer controlled safety plant, an operating system for local government records, etc.

The reasons for analysing just one of these data sets (number 3) of the seventeen collected during the Alvey Software Reliability Modelling (SRM) project were that

- the data came from a large software development project where feedback from the data supplier was available
- the data set was almost complete
- there were no specific requests from the customer to collect software-reliability-model-specific data such as execution time to failure or accurate estimates of source sizes and since the collection of this type of data increases the project cost, the data from this project is probably more representative of the data sets collected in industry.

Thus, the type of data available (see table 1.1) is the same as may be collected for any software development project. The collected data included the date of the software system failure, the number of faults found on failure, the size, type and language of each failed and unfailed source version and the product release dates. This information may be readily collected for any medium to large scale development project where there are a reasonably high number of system failures. For this data set, there were about six hundred software system failures.

TABLE 1.1. TABLE OF DELIVERED ALVEY DATA SET NUMBER 3 FILES

NAME OF FILE	COLUMN NAMES IN FILE
FAILURE	FAILURE, FAULT, PRODUCT, VERSION, PRODUCT REPAIR VERSION GROUP, INSTALLATION, WHEN FAILED, TYPE-_OF_USE.
FAULT	FAULT.
FAULT.PV.PRVG	FAULT, PRODUCT, VERSION, PRODUCT VERSION.
FAULT.SV.SRVG	FAULT, SOURCE, SOURCE VERSION, SOURCE REPAIR VERSION GROUP.
INVESTIGATION	FAILURE, REPAIR PROGRAMMER, WHEN REPAIRED.
PRODUCT.VER	PRODUCT, VERSION, WHO CHANGED THE VERSION, START OF PRODUCT VERSION, END OF PRODUCT VERSION.
PRODUCT_RVG	PRODUCT REPAIR VERSION GROUP.
PV.PRVG	PRODUCT, VERSION, PRODUCT REPAIR VERSION GROUP.
PV.PRVG.INS	PRODUCT, VERSION, PRODUCT REPAIR VERSION GROUP, INSTALLATION, START TIME OF PRODUCT VERSION, END TIME OF PRODUCT VERSION.
REPAIR	REPAIR, FAULT.
REPAIR.VER	REPAIR, REPAIR VERSION, REPAIR PROGRAMMER.
REPORT	REPORT, FAILURE, DATE OF REPORT.

TABLE 1.1. TABLE OF DELIVERED ALVEY DATA SET NUMBER 3 FILES (CONTINUED)

RV.SRVG	REPAIR, REPAIR VERSION, SOURCE REPAIR VERSION GROUP.
SOURCE	SOURCE, DESCRIPTION.
SOURCE.VER	SOURCE, SOURCE VERSION, PROGRAMMER, WHEN COMPLETED, LANGUAGE, SIZE, MEASURE.
SOURCE_RVG	SOURCE REPAIR VERSION GROUP.
SRVG.PRVG	SOURCE REPAIR VERSION GROUP, PRODUCT REPAIR VERSION GROUP.
STAFF	STAFF, STAFF RATE.
SV.PV	SOURCE, SOURCE VERSION, PRODUCT VERSION.
SV.SRVG	SOURCE, SOURCE VERSION, SOURCE REPAIR VERSION GROUP.

1.5.1 DESCRIPTION OF THE SOFTWARE SYSTEM

A detailed description of how the software system is supposed to work and the procedures to be carried out on failure, fault finding procedures and repair documentation should be available to the data analyst so that certain statistical assumptions may be checked. For instance, suppose two different repair procedures (e.g. factory and customer) were being followed. If the statistical analysis were to show that number of failures after repair by one procedure were significantly smaller than the number of failures after repair by the second procedure, then this improvement could be tracked down to one of the repair

procedures being more user friendly than the other. A recommendation may then be to use the more helpful procedure.

For the Alvey data set number 3 to be analysed, very little subsidiary information was available so that a number of possible conclusions to the analyses are listed. Therefore the recommendations of these analyses can be only tentative without further knowledge of the system.

1.5.2 CHECK FOR MISSING OR CORRUPT DATA

At each stage of the process of a collection, there may be missing or incorrectly written information and so the first objective of analysis of failure reports is to determine and complete the missing data. This may require a designer to work through the reports to check that what has been written is a true record of how the failure happened and whether the failure was actually due to the fault as described in the report. This procedure is not usually implemented as it is too costly with very little return of investment.

After the incident reports have been coded into a database, cross referencing files which contain the same data but in a different format may highlight incorrect data and this has been carried out on Alvey data set number 3 and is described in chapter 3.

1.5.3 DISCRIMINANT ANALYSIS

This analysis was carried out to determine if missing values may be predicted with a high probability based on

the data which has already been collected. The results pertaining to the analysis of Alvey data set number 7 are presented in chapter 7.

1.5.4 MULTIVARIATE ANALYSES FOR DETERMINING STRUCTURE

A number of multivariate methods are available to determine whether there is structure in the data which may be utilised (e.g. correlation between different variables). Hill (1974) and Teichmann et al (1985) describe applications of these methods to data collection. Principal components analysis was attempted on Alvey data set number 3 (in chapter 7) but without much success and it was felt that more relevant information could be gleaned from working from the original data rather than artificially generated variables which may be difficult to interpret.

1.5.5 EXPLORATORY DATA ANALYSIS (EDA)

The EDA approach is to look at simple plots of the data to determine possible structure and which statistical methods may be applicable for further data analysis. This approach may aid multivariate analyses such as principal components or cluster analysis mentioned in section 1.5.4 above. Three plots of the original data are shown in chapter 3 and for each, appropriate statistical modelling methods were applied based on the EDA results. The use of box and whisker plots, correlation and regression are also used as exploratory tools.

1.5.6 MODELLING

Statistical modelling of the data is useful in determining if there is any structure, to provide estimates of reliability and to determine the effect of the structure on the reliability. The statistical modelling methods which were applied after EDA were time series, log-linear modelling, generalised linear modelling, proportional intensity and proportional hazards modelling. These methods were recommended by the Alvey consortium as being useful in finding and modelling structure for the software reliability data which was collected during the Alvey SRM project. It is shown in this thesis that different methods may be applied to the same data to give similar conclusions.

1.5.7 CONCLUSIONS AND RECOMMENDATIONS

The conclusions of the analyses should ideally be discussed with the software project manager so that any recommendations may be determined as cost effective and/or applicable for implementation in some future similar project. Recommendations based on the analyses has been fed back to the data supplier for Alvey data set number 3.

1.5.8 FEEDBACK, GUIDELINES AND IMPROVED PROCEDURES

Every analysis should provide feedback into the system to aid future decision making. A detailed statistical analysis of a project should provide guidelines for any future project and possibly improved methods of software development and/or data management.

1.6 TABLE OF ANALYSIS METHODS

The following table lists the analysis methods; if they have been implemented on Alvey data set number 3, and in which chapter of this thesis they may be found.

TABLE 1.2. TABLE OF ANALYSIS METHODS

Procedure	Implemented	Chapter
1. Describe physical and functional system	Yes, Brief Description of Project	3
2. Check for missing or corrupt data	Yes by file comparison	3
3. Discriminant analysis	Yes	7
4. Multivariate Analyses for Determining Structure:- correspondence analysis, cluster analysis, correlation analysis, measures of distance, etc	No	-
4.1. Principal components analysis	Yes	7
5. EDA	Yes	3, 4

TABLE 1.2. TABLE OF ANALYSIS METHODS (CONTINUED)

Procedure	Implemented	Chapter
6. Modelling	Yes	See Below
6.1. Time series	Yes	4
6.2. Proportional hazards modelling	Yes	5
6.3. Logistic regression	Yes	6
6.4. Proportional intensity modelling	Yes	6
6.5. Log-linear modelling	Yes	7
7. Conclusions and Recommendations	Yes	8
8. Guidelines and Feedback	No	-

2 THE ALVEY SOFTWARE RELIABILITY MODELLING PROJECT

The Alvey software reliability modelling project was a multi-tasked project consisting of a collaborative team from UK industry and academia. Over the duration of the project 1985-1990, the membership consisted of the National Centre of Systems Reliability (AEA Technology), British Aerospace, STC, Logica, Nottingham Polytechnic and City and Newcastle Universities. The objectives of the software reliability modelling project were to investigate a wide variety of methods, to judge the relative merits of each method, to effectively communicate the results of the research and to indicate the direction of future research. The project consisted of a number of tasks within which this thesis describes areas in which Nottingham Polytechnic were task leaders; these are task 3 (statistical models with explanatory variables), task 4 (statistical models with different underlying assumptions) and task 9 (data collection and initial analysis).

There has been growing concern in the software industry about unreliable software for many years and as a result there have been some initiatives aimed at reducing the impact of the problem. Customers have imposed codes of practice on suppliers, lists of "approved" software have been specified and work has been done on improved testing strategies. However, up until the instigation of the Alvey project, little co-ordinated research has been done nationally on modelling software reliability.

The majority of the work within this thesis was carried out during the Alvey Software Reliability Modelling (SRM) project. The Alvey programme was set up in 1983 to research software engineering, intelligent knowledge based systems

and VLSI. The software engineering part addressed formal methods, software reliability, associated metrics and use of knowledge based systems. The appearance in 1984 of the "Software reliability and metrics programme" document from the Alvey Directorate formed a natural focus for this work. A consortium was formed containing members from both academic and commercial backgrounds with the intention of conducting a research programme to improve the state of the art. The programme involved active meetings to promote awareness of techniques by an interchange of information and views.

In July 1985, the Alvey Directorate placed a contract for the detailed study of software reliability modelling (SRM) with the aim of producing a plan for a National SRM Programme. The suggested course for the research was instilled into a set of project tasks. These tasks were as follows:-

Task 1: Improving Current Statistical Models
Task 2: Methods of Evaluating Statistical Models
Task 3: Statistical Models with Explanatory Variables
Task 4: Alternative Statistical Models
Task 5: Functional Modelling
Task 6: Models for Special Systems
6.1: Models for VLSI Systems
6.2: Models for Distributed Systems
6.3: Concurrent/Real Time Systems
6.4: Models for Fault Tolerant Systems
6.5: Reusable Software Components
Task 7: Cost Based Models
Task 8: Testing and Reliability
Task 9: Data Collection and Analysis

The project ended in June 1990 and more than sixty documents have been written during the project and a number of these are available to the public. Nottingham Polytechnic was mainly involved in three project tasks. These tasks are listed below and details of some of the work carried out in each sub-task is given under each task listing. This thesis describes the work in task areas 9, 3.1-3.3, 4.1 and 4.3-4.5.

2.1 WORK CARRIED OUT IN TASK AREA 3

Task 3
Participants: Nottingham, British Aerospace, City University, NCSR

2.1.1 SUB-TASK 3.1. INVESTIGATION OF GROUNDS FOR POTENTIAL MODELS

A report, (Wightman (1987)), was written which reviewed the models and techniques which incorporate explanatory variables and can be adapted to software reliability modelling. Techniques and methodologies reviewed were Software Science, Information Theoretic approach, simple regression, multivariate analysis, proportional hazards modelling and generalized linear modelling.

2.1.2 SUB-TASK 3.2. IDENTIFICATION OF NATURE OF DEVELOPMENT, USE SCENARIO AND EXPLANATORY VARIABLES

A comprehensive list of potential explanatory variables is supplied in McCollin, Wightman and Bendell (1989). This contains the types of information which may be collected during each phase of a software project.

2.1.3 SUB-TASK 3.3. DEVELOPMENT/SPECIALISATION OF MODELS

This sub-task was split into a number of areas as follows:- extensions to parametric models, non-parametric and semi-parametric models, models for environment and severity and the information theoretic approach.

The City University has contributed a number of reports on extensions to existing software reliability models. These cover task areas 3, 4 and 1 (improvement of current models). There follows a description of three of these reports.

A Bayesian formulation of the Jelinski-Moranda software reliability model (Csenki (1989)) reports that the model performance seems to be at least as good as some other models. In Brocklehurst (1987), a simulation study is reported which investigates if a general but simple adaptive procedure which improves the accuracy of predictions also increases the variability of the predictions. A City University report by Wright describes an extension to the "u-plot" (used for assessing predictive performance or for obtaining improved "adapted" or "re-calibrated" predictors) to allow for discrete or mixed predictive distributions. Two further modifications of the u-plot are documented which improve the performance of re-calibrated predictors.

Previous applications of the semi-parametric proportional hazards model: Cox (1972a), Wightman (1987), Prentice, Williams and Peterson (1981), Anderson and Gill (1982); has been based upon modulated renewal processes where the explanatory variables modulate the underlying renewal process. Recently, Lawless (1987) has introduced model

formulations which allow explanatory variables to be considered within a Poisson process. These proportional intensity Poisson process models allow the traditional non homogeneous Poisson process software models to be combined with explanatory variables. Details of the approach are given in a chapter 7.

Work has been carried out at Nottingham in expressing binomial type models and Poisson type models of exponential class (as classified by Musa et al (1987)), within a PHM framework. Details are described in a chapter 5. The formulation of the well known software reliability models within a PHM framework is useful for a check of the appropriateness of these models to data in that if the explanatory variable (e.g. such as the number of software failures in the Musa model) is not significant in the PHM formulation then the models are not appropriate for the data analysis.

2.1.4 SUB-TASK 3.4. PROTOTYPE SOFTWARE DEVELOPMENT

A number of program sources were written during the project. These are described in Hufton (1989).

Software has been developed at Nottingham Polytechnic for Poisson Proportional Intensity models with covariates and an unspecified baseline intensity. This software was used in the analysis of Alvey data set 3.

2.1.5 SUB-TASK 3.5. GUIDELINES FOR RELIABILITY AND SAFETY ASSESSMENT OF SOFTWARE (GRASS)

The work in this area was mainly carried out by the Safety and Reliability Directorate at A.E.A. Technology and the reference by Dale (1989) was one of a number of reports which were delivered to Alvey on the subject.

A internal report has been written by Wightman, McCollin and Bendell relating experience of applying a proportional hazards modelling formulation to Alvey data set 5.

2.2 WORK CARRIED OUT IN TASK AREA 4

Task 4
Participants: Nottingham, British Aerospace, City University

2.2.1 SUB-TASK 4.1. OTHER STOCHASTIC POINT PROCESSES AND NON-PARAMETRIC PROCEDURES

The work of task 4.1 is covered in the section on task 3.3 above.

2.2.2 SUB-TASK 4.2. ENTROPY APPROACH

The work in this area was carried out by British Aerospace. A report (Anderson (1986)) was written on the potential usefulness of entropy and information theory and was delivered to Alvey in 1986.

2.2.3 SUB-TASK 4.3. TIME SERIES METHODS

Work has been carried out at Nottingham on the use of time series for reliability. Details are given in chapter 4 on modelling of the data collection process.

2.2.4 SUB-TASK 4.4. MULTIVARIATE TECHNIQUES

Multivariate techniques including proportional hazards modelling and proportional intensity modelling are amongst the most useful techniques available as they have been developed extensively in the biometry field. This is because patient (sic hardware/software system) reliability is analysed with respect to his/her life history.

Classical multivariate techniques were employed via the statistical computer package MINITAB in an attempt to reduce the mostly discrete, many dimensional multivariate data space down to a more workable 2 or 3 dimensional space and discriminate between, for example, programmers on the basis of discriminating explanatory variables which would enable further incidence of faults to be attributed to particular programmers on the basis of a profile of observations across the multivariate data space. These areas are discussed in chapter 7 with respect to the analysis of Alvey data set 3.

The discrete nature of the data suggest that specialised discrete multivariate techniques might be investigated and employed. Some effort by Mr Peter Dixon at Nottingham resulted in the setting up, testing and use of specialist FORTRAN software for discrete multivariate analysis and this was carried out at Nottingham during the Alvey project with a pleasing measure of success.

It is possible to use MINITAB to obtain multiway tables depicting number of faults by programmer, language, size and type of source. Problems of sparseness were overcome by collapsing tables into one another, where reasonable to combine source sizes into 'large', 'medium' and 'small'. Log-linear modelling is recommended as the most powerful analytical tool for the examination of such software data.

Logistic modelling is carried out in chapter 7 and is shown to be a special case of proportional intensity modelling with a Weibull intensity function. Proportional intensity modelling is also used to analyse a subset of the Alvey data set number 3.

In task area 8, a report was written on some applications of generalised linear models to software reliability and this is described in chapter 7 and Hufton and Exley (1989).

A report was written by A.E.A Technology for Alvey (Hufton (1989)) on the possible use of polytomous regression models to estimate software reliability. The conclusions were that the method reveals more precisely than PHM, the effects of testing; there is no assumption of a baseline time metric which is required for PHM and a wide variety of models are available to choose from to select the most appropriate to the data at hand. The disadvantages of the method are that it requires 'detective' work to determine some of the characteristics of the software faults and the models may be difficult to use computationally.

2.2.5 SUB-TASK 4.5. EXPLORATORY DATA ANALYSIS (EDA) APPROACH

A number of simple plots of the data in Alvey data set number 3 are given in chapters 3 and 4. These allow the structure in the data to be seen and to be further investigated by various modelling techniques.

The approach of Walls and Bendell (1985) of identifying trend and serial correlation before distribution fitting has been applied to the residuals from the Box-Jenkins time series analysis in chapter 4.

The use of PHM as a diagnostic tool has been described in Ansell and Phillips (1989) and is also discussed in chapter 6.

Simple correlation and regression is used on cumulative time to failure data to determine possible model structures within the Alvey 3 data set. This is also discussed in chapter 3.

2.3 SOFTWARE DATA COLLECTION

There have been a number of software data collection schemes prior to the Alvey SRM project task 9 activities. These have been reported by Walston and Felix (1977), Sukert (1980), Musa (1980), Basili and Selby (1984), Nagel and Skrivan (1981), Kitchenham (1984) and Martini et al (1990) and (1991). A brief description of some of these is given below. The statistical techniques used to analyse the collected data as described in these references are

detailed in table 2.1 below. These may be compared with the analyses presented in this thesis which are listed in table 9.1.

2.3.1 APPROACH OF WALSTON AND FELIX (1977)

The software measurement project started in 1972 to assess the effects of structured programming on the software development process. Data from sixty completed software development projects was contained in questionnaires submitted at prescribed reporting periods during the projects and were stored in a computer data base. Data such as number of lines of delivered source code; language; effort in man-months; duration of the project in months; use of structured programming, inspections, top-down development, etc and a measure of programming productivity (number of delivered source codes to total effort in man-months). Statistical analysis comprised of calculating means, medians, modes and standard deviations of the specific variables and computing characteristics. Five major parameters were identified : productivity, schedule, cost, quality and size. They showed that total effort was nearly linearly related to product size. A set of twenty nine out of sixty eight variables were shown to be correlated highly with productivity and these variables were formed into a linear combination productivity index. This index was calculated for fifty one of the projects and plotted against actual productivity with a high degree of correlation.

2.3.2 APPROACH OF MUSA (1980)

The Data and Analysis Center for Software (DACS) developed and maintained a computer database containing software

data and documentation. John Musa of Bell Telephone Laboratories submitted a report on sixteen data sets containing failure interval data, execution time, size and application. Modes, confidence intervals, plots of hazard rates, estimates of MTTF were calculated and these data sets have been analysed in detail by time series methods and PHM; (see Davies et al (1987) among others).

2.3.3 APPROACH OF NAGEL AND SKRIVAN (1981)

Boeing Computer Services carried out designed and controlled experiments in which two programmers designed and coded three sources each from three problem specifications. These sources were then executed and their interfailure times were recorded. The conditional distribution of interfailure time given the number of errors corrected was shown to be exponentially distributed and it was observed that the log failure rate of interfailure time was nearly linear as a function of the number of errors corrected. PHM was applied to the data and strong programmer and problem effects were seen to affect the baseline hazard rate.

2.3.4 APPROACH OF KITCHENHAM (1984)

A data collection and analysis system was set up to be used by production staff to produce the ICL VME operating system. Information collected included date of change, programmer, type of change and failure severity. Plots of source program size against number of errors per source were shown; the plot being nearly linear. Tables of numbers of errors against types of errors and number of errors against method of screening were also given.

2.3.5 APPROACH OF MARTINI ET AL (1990,1991)

Failure data collected over twenty seven months were recorded and 461 reports were raised on the software of the TROPICO-R Switching System. Data collected included test and field data. Analysis methods applied were the Laplace test for trend and the Goel-Okumoto model (1979) and S-shaped reliability growth model of Yamada et al (1983) were successfully fitted to the cumulative time to failure data.

TABLE 2.1

TABLE OF STATISTICAL METHODS USED IN STATED REFERENCES

Heading	CLM	Summary Stats.	PHM	Laplace test	NHPP
2.3.1	+	+	-	-	-
2.3.2	-	+	*	-	-
2.3.3	-	-	+	-	-
2.3.4	+	-	-	-	-
2.3.5	-	-	-	+	+

CLM means correlation and linear modelling

* means that PHM was reported in a subsequent reference.

+ means method used.

- means method not used.

2.4 ALVEY SRM PROJECT TASK 9 ACTIVITIES

Task 9

Participants: Nottingham, Logica, STC

Sub-tasks

9.1 Data collection, organisation and management (database design)
9.2 Provision of data :- Logica, STC
9.3 Exploratory data analysis

On the Alvey SRM project, there have been a total of fifteen Task 9 consortium meetings from 2nd February 1987 to 23rd May 1988 which covered the major developments in the task 9 activities over the duration of the project. An abridged version of the task 9 activities are described in a report compiled by McCollin (1990). The activities were

data management

progress of data sets: acquisition, collection, transfer to database and preliminary analysis

establishment of the computer network and creation of the software reliability relational database and associated documentation.

Problems encountered during the project and recommendations for areas of research based on the types of data collected are described below.

2.4.1 DATA MANAGEMENT

A 4 point procedure for data acquisition and the relationship between data providers and data users was put forward:

i) data sources to be identified, documented and put forward for approval by an individual member of the consortium.

ii) quality and relevance of data to be agreed upon and approved by task group 9.

iii) data transfer to be arranged and data made available.

iv) data sets to be subject to initial analysis by Nottingham team.

2.4.2 PROGRESS OF DATA SETS: ACQUISITION, COLLECTION, TRANSFER TO DATABASE AND PRELIMINARY ANALYSIS

Two of the collaborators and a number of external organisations supplied data from on-going projects. Some of the organisations who were requested for data were Marconi Data Systems, Plessey Telecoms and CCTA. CCTA recorded all the hardware and software failures for more than four hundred Government establishments from the period 1980-1986. A survey of fifty of these data sets did not produce many software failures. Alvey data set number 8 is one of these which contained software and hardware field use failures. The first data set (number 1) was delivered to Nottingham for initial analysis in 1987.

At the end of the project in July 1989, out of the sixteen data sets collected by consortium members, four had been installed onto the database, six were not in the correct format for database installation and six were awaiting resources for collection. Four data sets have been analysed, two by Nottingham (numbers 1 and 3), one (number 14 by A.E.A Technology) and one (number 13 by The City

University). The software data library (SWDL) and the Request project have also allowed SRM data users data access. A description of the types of data collected are listed for eight of the data sets below. The other data sets (up to number 15 were kept at City University and A.E.A Technology).

TABLE 2.2

TABLE OF ALVEY DATA SETS (NOTTINGHAM POLYTECHNIC INPUT)

Data set	Time metric	Explanatory Variables
1	CPU time to failures, days to failure of sources	source failed, failure severity, inspect or not, source repaired, cause, test
3	Number of failures per day of system, Days to failure of sources	language, source size, source type, day of week failed, source version, first appearance, final appearance of fault, fault number, programmer
5	Days to failure of sources, run time to failure of sources, days to failure of systems	source version, source size, failure type, severity
6	As data set 5	As data set 5

TABLE 2.2 (CONTINUED)

TABLE OF ALVEY DATA SET NUMBER AGAINST TIME METRIC AND EXPLANATORY VARIABLES (NOTTINGHAM POLYTECHNIC INPUT)

7	Days to failure of system	repair description, repair time, severity, failure description, cause, fault description
8	Number of failures per day of system, times to failure of sources, days to failure of system	installation at fault, fault description, repair programmer, repair description, repair date
11	Days to failure of sources	cause, repaired source numbers, repair date, severity, phase, size, source type, language
15	Days to failure of sources	repair date, repair programmer, severity, fault description, repair hours, test

2.4.3 ESTABLISHMENT OF THE COMPUTER NETWORK AND CREATION OF THE SOFTWARE RELIABILITY RELATIONAL DATABASE AND ASSOCIATED DOCUMENTATION

In Task area 9, considerable effort was invested in the creation of a software reliability database which was installed on a dedicated VAX mainframe computer at The Centre for Software Reliability, City University (TCU). Two Alvey documents which describe the database definition

and structure are Mellor (1986) and Mellor (1987). The database was completed in mid 1988 and Logica supplied a copy of the final version of the database technical guide (Potter (1988)) to TCU who have copied them to the other database users. The database design document and the database manual were made deliverable documents and sent to the DTI, (Simmonds and Potter (1989)). Private computer links were to be available to the database users at Nottingham, Logica and Newcastle University although only one network link between TCU and Nottingham Polytechnic was established.

2.4.4 PROBLEMS ENCOUNTERED DURING THE PROJECT

Nottingham, Logica and STC have written a paper which was presented at the Euredata conference in Siena, (Bendell et al (1988)) which outlines the main problems of software failure data collection. These were:

-the late establishment of the network link, computer facility and database so that very little data had been inputted into the database and analysed before the end of task 9 activities.

-the lack of manpower available for data coding and input into the database.

-the lack of resources available to procure large, potentially useful data sets.

Exploratory data analysis of data set number 3 was carried out and the problems associated with this are given in the next chapter.

2.4.5 CONCLUSIONS AND RECOMMENDATIONS FOR AREAS OF RESEARCH BASED ON THE TYPES OF DATA COLLECTED

The conclusions and recommendations from task 9 were:

There are organisational and technical difficulties in collecting software reliability data.

There are potential features of software data which make the reliability analysis extremely inconvenient.

It is essential that from the inception of a software reliability project, the collection and analysis of reliability data is under strong management control. For example, there must be good feedback to the data providers.

The main purpose of the data collection exercise was to determine suitable models for software reliability estimation and to establish models which would incorporate explanatory factors found during the data collection.

Only two of the data sets provided have some measure of execution time and most of them have days to failure as the time metric for reliability analysis. The main statistical software reliability models, e.g. Jelinski-Moranda (1972), Littlewood (1981), etc incorporate execution time but not calendar time. It was found during this exercise that organisations do not usually collect execution time to failure of sources because:

it is a costly exercise to collect execution time to failure.

the customer only usually requires execution time if he wishes to estimate software reliability by using one of the available models.

the collection of execution time is not a requirement of general software guidelines or standards (e.g. Tick-It, DEF-STAN 00-55).

Data acquired so far points to the main directions of research in task areas 3 and 4 being time series, proportional hazards modelling and multivariate techniques. It is shown in the next chapter that an exploratory approach to a data set shows that these models are the most appropriate for analysis.

3 EXPLORATORY ANALYSIS OF ALVEY DATA SET NUMBER 3

Due to the difficulties and time delays in establishing the central computer facility and to loading data centrally, a number of data sets were delivered directly to Nottingham for initial Task 9 analysis. The format of the data sets have ranged from summaries of failure counts on networked systems, completed failure and repair reports on field data, software test and inspection information and cpu times to failure for individual computer installations. The methods of analysing one of the delivered data sets are discussed in the rest of this chapter. An exploratory approach is adopted to determine possible structure and which statistical models are the most appropriate for further data analysis.

3.1 SOFTWARE FOR DATA CHECKING

As the Alvey SRM database was not on-line when the data set number 3 was delivered, the twenty files containing more than one hundred thousand items of data was transferred to the Nottingham Polytechnic VAX mainframe computer. To analyse this data required a suite of sources to be written so that the data could be manipulated into a form where statistical analysis could take place. A standard database or spreadsheet was not used as there would have been problems in inputting and manipulating alphanumeric data.

The EZ-source.BAS packages were written by Mr Graham Dawson and myself to simplify the task of manipulating data files. The purpose of these sources is listed below:

EZcount	Counts the number of lines within a file.
EZdelete	Deletes an item from all lines.

EZmerge Merges a file into a reference file.
EZremove Removes all excess white space characters.
EZsort Sorts a file with reference to one column.
EZsort2 Sorts a file with reference to two columns.
EZswap Swaps two items in all lines of a file.
EZchron Converts a date to a number of days from a start.
EZdatesort Sorts a file with dates by date order.

3.1.1 DESCRIPTION OF THE DATA

Alvey data set number 3 contains data collected from one software product running on a single installation. The data set was collected during the development phase of the project and the software was continually being operated and repaired after failure. The software comprises of 1198 source versions of which 1096 are written in Cobol, 99 in an operating system language and 3 in a third language. There are 6 types of source : 87 command macros, 6 command macro data files in the operating system language, 608 module main source codes, 78 control binding files, 126 Cobol include files and 21 screen definition files. These were numbered types 1 to 6 respectively for convenience.

3.2 CHECKING FOR INCORRECT DATA

The first of the twenty files to be looked at was the file 'Failure' because this would supply the information gleaned from the failure reports, notably, failure number, fault number and the failure date which would be immediately useful for statistical analysis. By sorting this file by date order, the majority of the failure numbers occurred in the correct numerical order apart from the first twenty two which were all dated 01/01/1985 and the product version

was 0. The failure numbers for these twenty two items were mainly in the 600's. These failures could be assumed to have occurred prior to the start of the project which was given as 16/07/1986 and were collected under a different numbering scheme. However by re-ordering the data by failure number, the 22 failure numbers did not tally with any other numbers. The tentative conclusion was that these 22 failures were not date coded on the original failure reports and so the date 01/01/1985 and product version 0 were default values. After discussion with the software supplier, it was confirmed that this was true.

The two files 'Failure' and 'Fault' were merged to find missing fault information. Merging was carried out using the common 'fault' field. Three fault numbers were missing out of a total of 607 fault numbers in the file 'Fault' and 119 faults were not recorded on failure reports. A similar exercise was carried out on the other files and a number of omissions and anomalies were noted. These are recorded on figure 3.1.

A number of files were merged to reduce the information which had only been collected to establish relationships in the data-base structure and the resulting data was surveyed for usable statistical data. Information pertaining to programmer, repair date and repair programmer could not be utilized as too much data was missing. As an example of missing data, there were 276 failure investigations with the default date 01/01/1985 out of a total of 670.

3.3 OBSERVATIONS OF THE DATA SET

For the purposes of analysis, certain features of the data are quite sensitive to the data supplier. For example, an analysis of the 96 failure free days was carried out and only on two occasions were failures recorded on a weekend. Of these, there were 26 failure free days during the test phase and 70 during the live phase. Out of the 16 weekends of the test phase during which data was generated, there were no failures recorded on 12 Saturdays and 10 Sundays. On two occasions there were long sequences of failure free days during the live phase. One of these periods was identified as Christmas and New Year. This means that very little data was collected over the weekends, even though the staff were booking time to the project in these periods. There was no indication from the data supplier that the system was less busy at weekends. The number of failures at weekends during the live phase were also recorded as zero.
After using the EZ source program package, it was possible to highlight the number of failures, faults, repairs and sources for Alvey data set number 3 showing where data was missing.

Sorting, counting and merging files was carried out initially to find any missing or corrupted data. The following observations were made of the data set:

1. There were 125 days with failure and 96 days without failure.

2. After the software was delivered to the customer (during the "live" phase from product version 17) there was a large increase in the number of days per product version

(see table 3.1).

3. In total, 570 "test" failures and 100 "live" failures were recorded.

4. There were four missing failures: numbers 455, 457, 591 and 660.

5. There were 500 numbered faults (fault numbering taking place when the faults were reported and again when they were repaired) and 170 zero numbered faults, some of the reasons being as follows:

no fault found;
the failure record was superseded and cleared by another record;
the fault was unconfirmed;
a change in the functional specification caused the fault to be nonrelevant; minimal effort was required to effect the fault repair;
the fault was not important enough at the time to be repaired and was left until a later date.

6. The same fault occurred on separate failure reports 28 times.

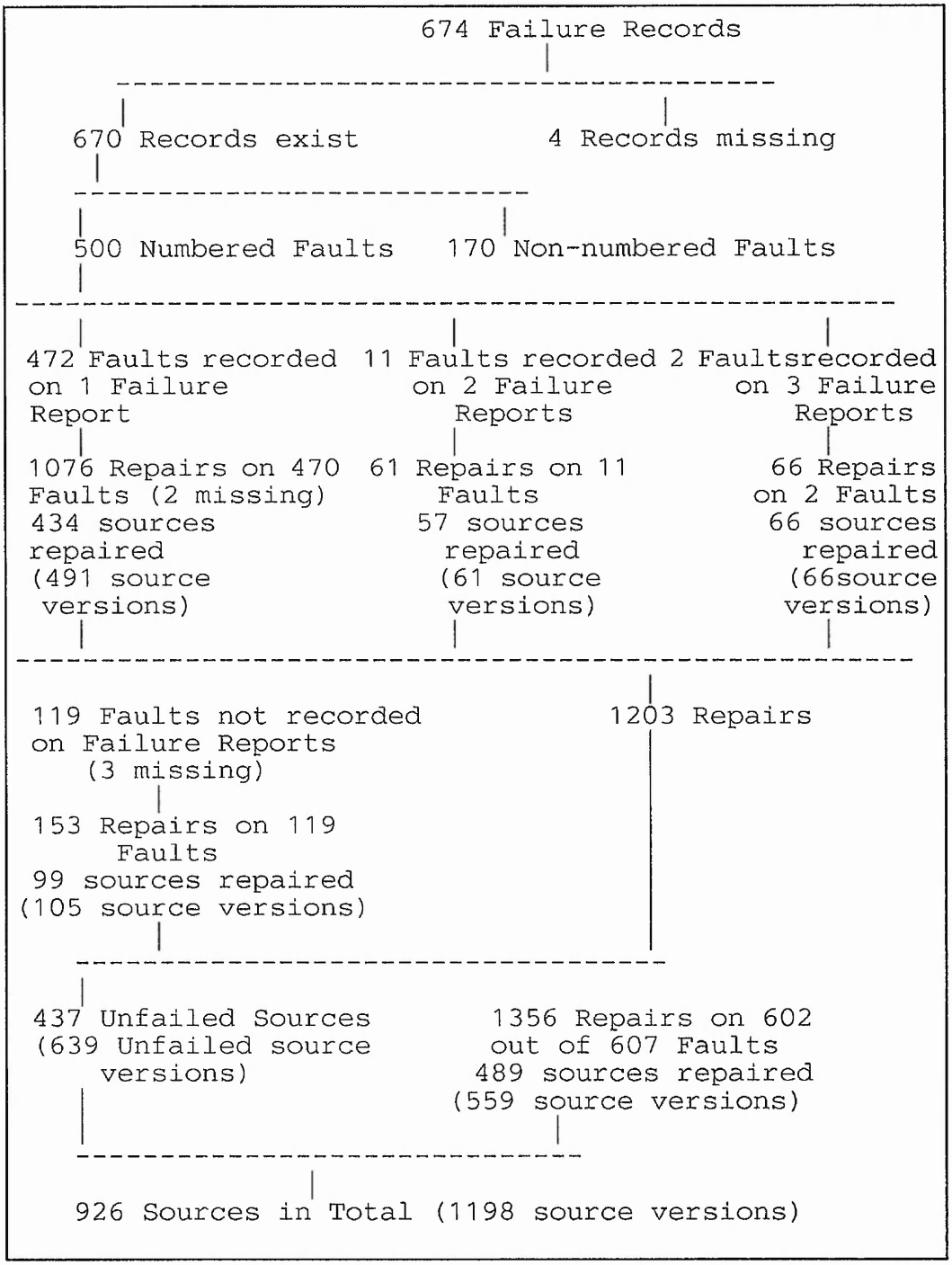
7. Two failures, numbers 118 and 279, did not correspond to any fault or repair information.

8. No fault and repair information was found for fault numbers 35, 547 and 553.

9. There were 677 unfailed source versions and 514 source versions which failed at least once.

Figure 3.1 summarises the failure and repair information of the data set.

FIGURE 3.1 ALVEY DATA SET NUMBER 3 FAILURE AND REPAIR RECORD INFORMATION



As can be seen in figure 3.1, the data format could come from any data collection scheme.

Some of the data in Alvey data set number 3 is presented below in tabular and graphical form. Recommendations for further modelling of the presented data is supplied and the modelling is carried out in chapters 4, 6 and 7.

Tables 3.1, 3.2 and 3.3 show the number of times product versions failed and sources and source versions (whenever a source has accumulated a significant number of repairs, a new version is released) were repaired. Table 3.4 shows the number of faults against the number of repairs per fault.

TABLE 3.1. NUMBER OF PRODUCT VERSION FAILURES

Version Number	0	1	2	3	4	5	6	7	8	9	10
No. of Failures	25	52	12	49	58	22	55	33	24	31	30
Failure Free Days	3	1	0	1	0	0	0	2	1	3	0
Failure Days	4	5	1	4	3	2	5	5	5	5	7
Days per Version	7	6	1	5	3	2	5	7	6	8	7

TABLE 3.1. (CONTINUED) NUMBER OF PRODUCT VERSION FAILURES

Version Number	11	12	13	14	15	16	17	18	19	20	All
No. of Failures	53	10	21	40	27	19	35	25	47	2	670
Failure Free Days	3	3	2	2	2	2	3	22	45	1	96
Failure Days	10	4	5	6	6	2	13	14	18	1	125
Days per Version	13	7	7	8	8	4	16	36	63	2	221

From the table, the following observations are noted :

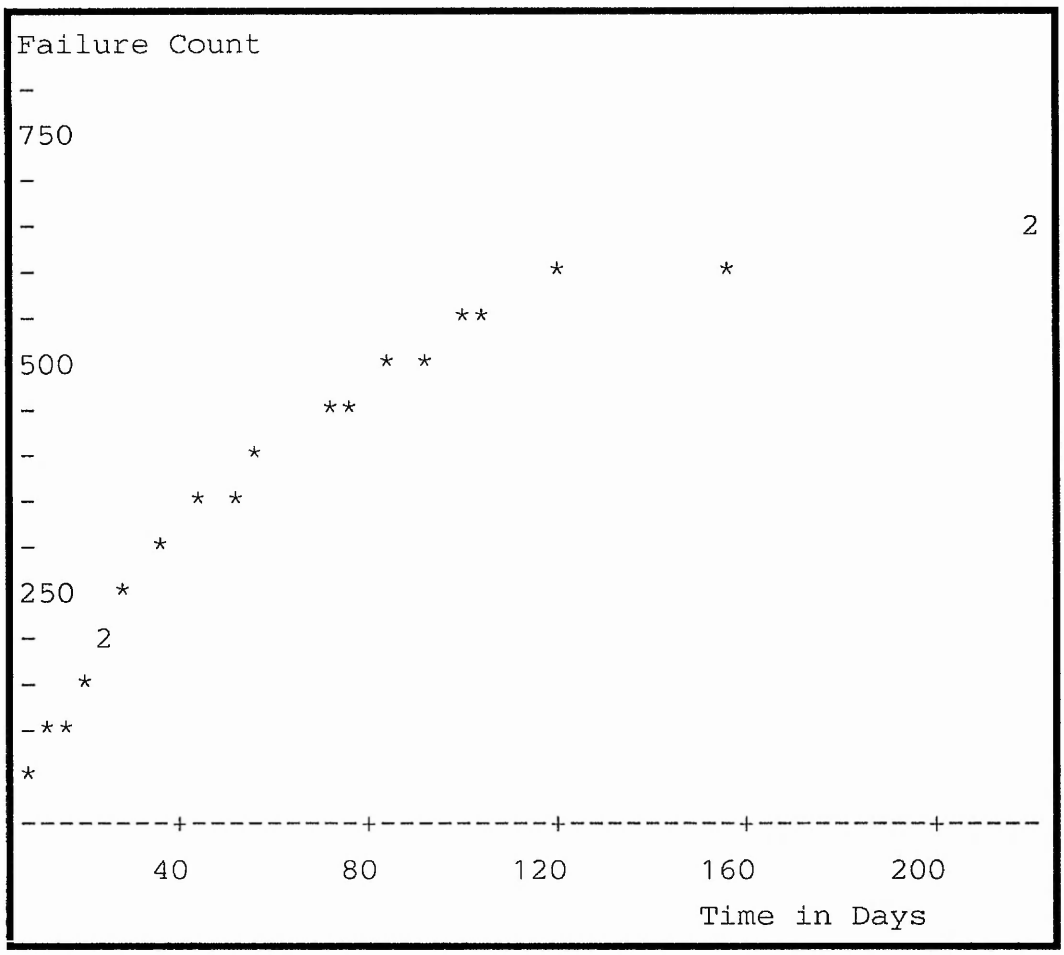
All versions to 16 inclusive were in the test phase, all from 18 inclusive were in the live phase. For version 17, there were 9 failures on 5 failure days and 1 non failure day in the test phase; and 26 failures on 8 failure days and 2 non failure days in the live phase.

There were 570 test failures occurring over 84 failure days and 26 non failure days (110 days in total). There were 100 live failures occurring over 41 failure days and 70 non failure days (111 days in total).

There is no relationship between number of failures and number of days per product version, the numbers of days being approximately constant.

The cumulative number of failures per product version is reducing over time. This can be seen in figure 3.2 with each asterisk indicating a new product version. Therefore statistical modelling of reliability growth is applicable.

FIGURE 3.2. PLOT OF FAILURE COUNT AGAINST TIME IN DAYS



The data was tabulated in different forms to investigate whether there was a Pareto effect.

TABLE 3.2. NUMBER OF TIMES SOURCES REPAIRED

Repairs	0	1	2	3	4	5	6	7	8
Sources	437	232	100	44	30	28	20	4	10
Repairs	9	10	11	12	13	14	15	20	32	51
Sources	3	6	3	1	2	2	1	1	1	1

It can be seen that 769 sources (83%) out of the 926 required two repairs or less which shows the Pareto principle. The twelve sources which failed the most number of times are analysed in detail within chapters 6 and 7 on proportional hazards modelling and generalised linear modelling as this will provide information on the minimal reliability of the software.

TABLE 3.3. NUMBER OF TIMES SOURCE VERSIONS REPAIRED

Repairs	0	1	2	3	4	5	6	7	8
Source Versions	639	289	111	50	32	27	16	8	8
Repairs	9	10	11	12	13	14	15	19	21
Source Versions	5	5	0	1	2	2	1	1	1

TABLE 3.4. NUMBER OF FAULTS AGAINST REPAIRS PER FAULT

Repairs per Fault	1	2	3	4	5	6	7	8	9	10
Number of Faults	397	89	32	29	22	11	2	2	3	3
Repairs per Fault	11	12	14	16	21	26	34	39	44	65
Number of Faults	1	3	1	1	1	1	1	1	1	1

From table 3.4, 518 out of a total of 602 faults (86%) required less than 4 repairs per fault. However, there were 15 faults (2.49%) which required more than 10 repairs per fault (again, a Pareto principle).

The incidence of a large number of faults occurs when a new set of source codes have been run for the first time. The number of repairs per fault may be regarded as a measure of severity of the fault and could be used as a covariate in proportional hazards modelling to explain the difference in hazard rates of a set of sources. There was no indication from the data supplier why there were such a high number of repairs to individual faults. A possible reason is that the programmers required a large number of trial repairs as the fault could not be isolated easily.

3.3.1 CONCLUSION

Analysis in this thesis has mainly centred on failures, however there is so much subsidiary information which has been and could have been collected on faults and repairs that there is no need to collect difficult or expensive to collect information for reliability analysis as there is a wealth of information available from a closed loop failure reporting scheme.

3.4 SIMPLE PLOTS OF THE DATA

Three plots from the Alvey data set 3 are shown below with suggested statistical techniques to model the structure.

Plotting the number of failures per day against day (figure 3.3) showed that there was a definite reduction in failures over time. On two occasions there were long sequences of failure free days during the live phase. One period of 13 days was identified as Christmas and New Year and the other of 11 days occurred after a very large number of failures (15) in a day. Possible reasons for these are:

system utilization was high.

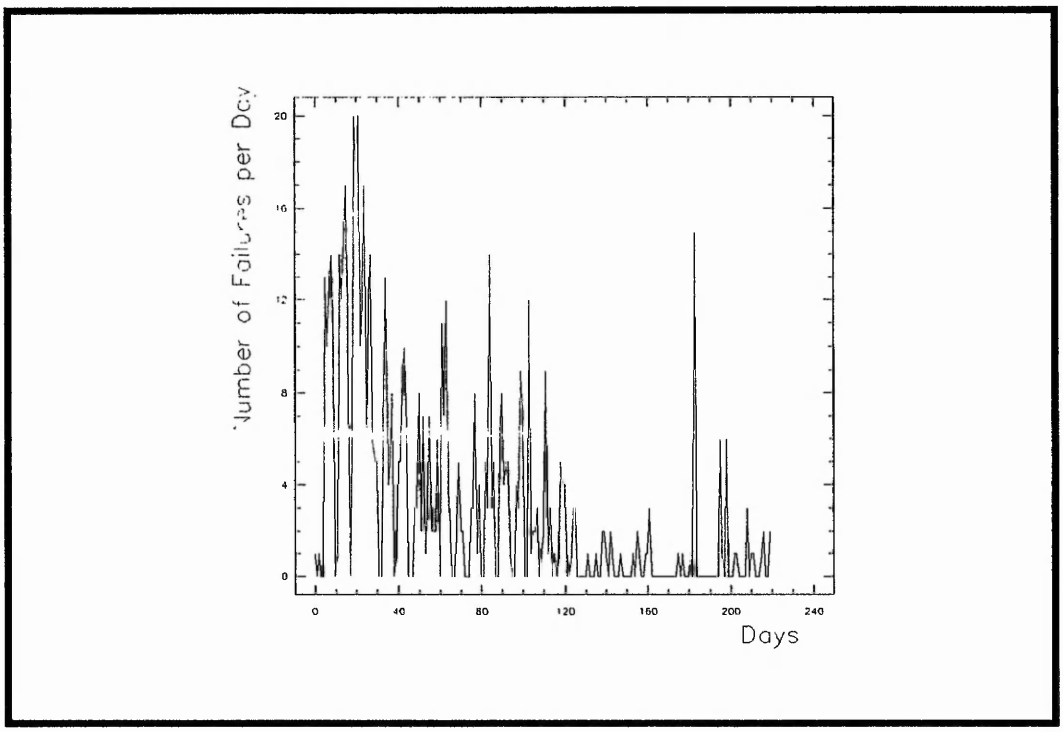
it was decided to raise reports against all minor failures which have been previously observed and ignored.

a new and enthusiastic repair programmer!

A time series approach was adopted in chapter 5 as the failure counts are fixed at one day intervals. Time series may be used to determine if reliability growth and/or seasonality have any effect on failure counts. The trend

and seasonality may also be modelled as covariate information in a proportional hazards model. The results are summarised in chapter 6.

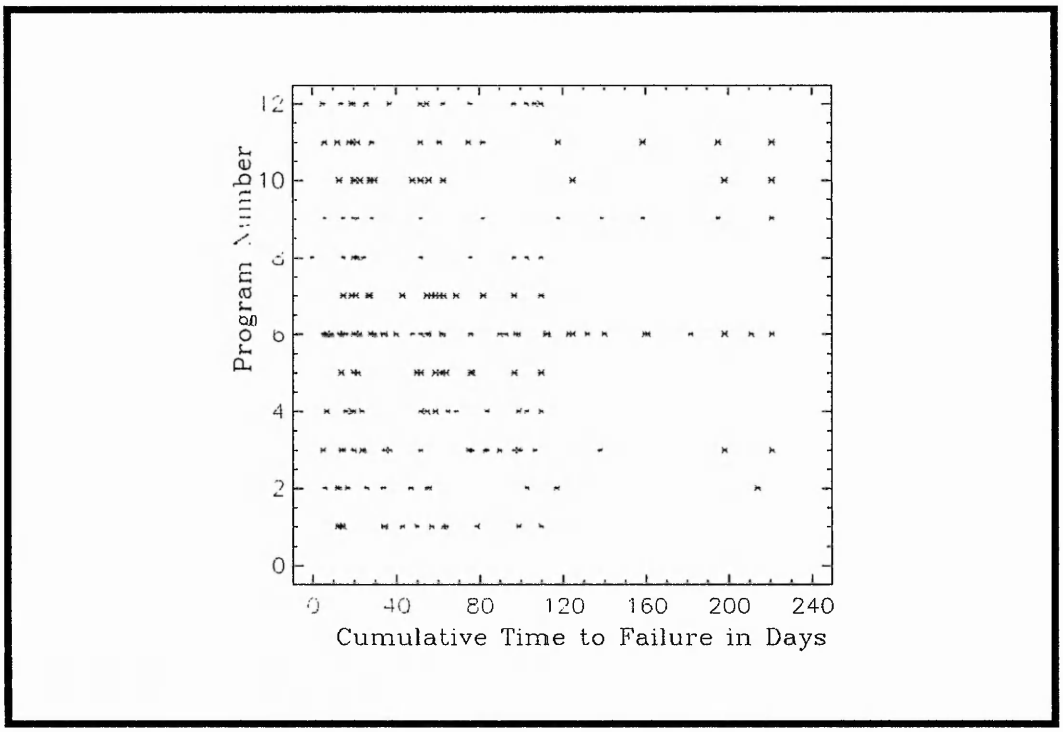
FIGURE 3.3. PLOT OF NUMBER OF FAILURES PER DAY AGAINST DAYS



The three sources, numbers 3, 6 and 10 which required most repairs were all Cobol include files. Of the remaining 9 sources which were repaired more than 10 times, eight were module main source codes, the other being a Cobol include file. These twelve sources were repaired 217 times out of a total 926 sources with 1356 repairs. The twelve sources were all Cobol files of size greater than 9 4K blocks of code and text of which for 10 of these, only one particular source version was repaired. Possible reasons for these

sources failing more often are that they are being used more heavily or are being tested more thoroughly. Figure 3.4 is a plot of the cumulative time to failure in days of the twelve sources. From the plot, it can be seen that the interfailure times are getting longer for all of the sources. The twelve least reliable sources provide a snapshot of the failure count in figure 3.3 above. Fagan (1976) suggests monitoring the least reliable sources (as they provide an indication of the software system reliability growth). The reliability growth of continuous time data (such as the days to failure) with explanatory information (such as source designation) may be modelled by a hazards or intensity formulation.

FIGURE 3.4. PLOT OF CUMULATIVE TTF AGAINST SOURCE NUMBER



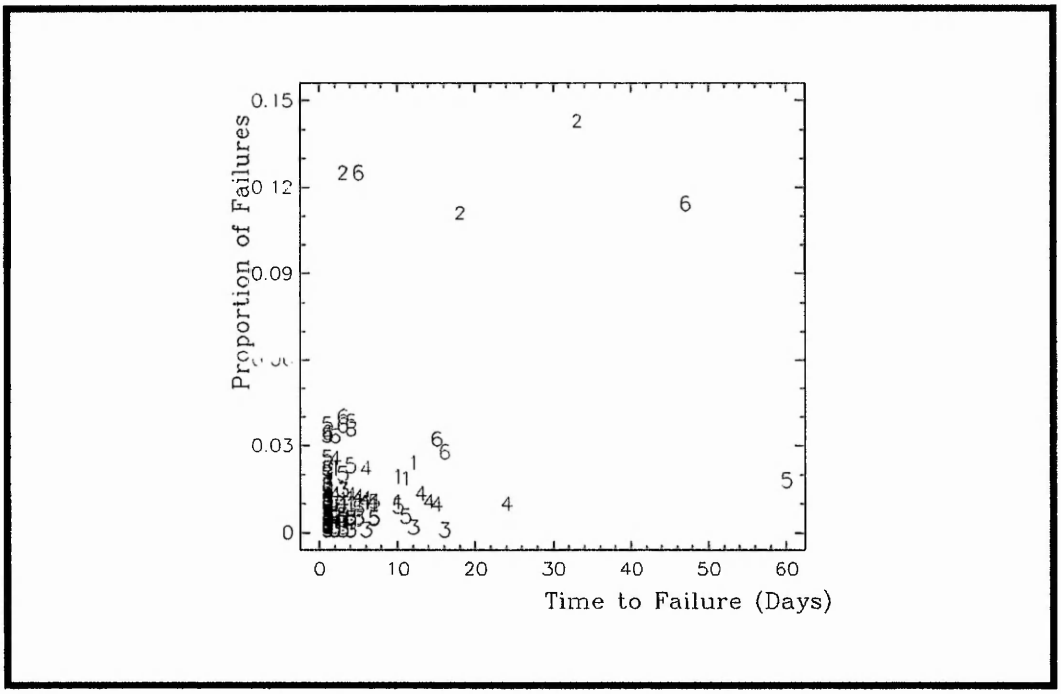
The twelve sources have been analysed using proportional hazards modelling, proportional intensity modelling and generalised linear modelling, the results of which are discussed in chapters 6 and 7 and McCollin, Wightman and Bendell (1989) and Wightman, McCollin and Dixon (1991). A number of explanatory factors could not be fitted together in PHM. This was due to the factors being collinear. The problem of multicollinearity of the covariates was investigated by applying regression to the number of failures and cumulative time to failure (the results being at the end of this chapter) and using multivariate techniques to search for relationships between failure count and source type and size. This is discussed in chapter 7.

It was known that the suite of source programs were run as a package and failures were recorded on a daily basis with the sources at fault. The total number of sources found at fault was 1356 for which 674 resulted in the package failing. The package was expanded throughout the development phase and on certain days the number of sources running increased without failures or faults occurring. This censoring information and the failure data with the type of source failed was collected over an eight month period and, for the six types of source, 269 failure and censoring points and 222 failure points were analysed.

The total number of sources running per day was recorded and the number of failures/faults per day to total number of sources running per day with number of failures/faults per day was calculated as a proportion. This adding of the top term (number of failures/faults per day) to the bottom of the expression made sure that the proportion always lay between zero and one. Figure 3.5 is a plot of

the proportion of failures against time to failure for each source type. From the figure, there are no immediate observations of any note. Further analysis of this proportion failed data is described in chapter 7.

FIGURE 3.5. PLOT OF THE PROPORTION OF FAILURES AGAINST THE TTF FOR EACH SOURCE TYPE



3.5 CORRELATION AND REGRESSION USED FOR EDA

An analysis of the twelve least reliable sources of Alvey data set number 3 was carried out. Plots of cumulative time to failure against failure number were graphed and if curvature was seen, then a logarithmic transformation was applied. The correlation coefficient was calculated for the original variables and for the transformed data. As can be seen from table 3.5, the correlation coefficients

are very close to one for every case of the original data. However, most of the plots of the cumulative time to failure against failure number are not linear and show a tendency for the times to increase with failure number, i.e. there is an upward curvature to the plot which signifies reliability growth. In all cases, the intercept was found to be nonsignificant. The initial linear model fitted is

$$CTTF = b * N$$

where CTTF is the Cumulative Time To Failure and N is the Failure Number.

When curvature was seen, initially cumulative time to failure was logged and if this transformed data was curved, then the failure number was also logged. The effect of the initial transformation is to create the model

$$CTTF = a * \exp(b * N)$$

where a is the intercept term.

which is the Logarithmic Non-homogeneous Poisson process (described in chapter 5).

The second transformation is the model

$$CTTF = a * N^{b}.$$

This model is the Non-homogeneous Poisson Process with Weibull intensity also described in chapter 5.

TABLE 3.5. CORRELATION AND REGRESSION INFORMATION OF ALVEY DATA SET NUMBER 3

Source No.	Corr. Coeff	Comments	1st Transform	2nd Transform
1	0.962	Upward Curvature	Linear	
2	0.936	Upward Curvature	Linear	
3 version 1	0.880	Upward Curvature	Linear	
3 version 2	0.944	Upward Curvature	Linear, two outliers	
4	0.987	Two Lines		
5 version 1	0.948	Too Few Failures		
5 version 2	0.899	Upward Curvature	Two Lines?	
6	0.967	Two Lines		
7 version 1	0.954	Linear, one outlier		
7 version 2	0.991	Linear		
8	0.912	Two Lines		

TABLE 3.5. (CONTINUED) CORRELATION AND REGRESSION INFORMATION OF ALVEY DATA SET NUMBER 3

Source No.	Corr. Coeff	Comments	1st Transform	2nd Transform
9	0.923	Two Lines	Linear, one outlier	Does not pass through origin
10	0.970	Two Lines?	Two Lines	Does not pass through origin
11	0.908	Upward Curvature	Linear	Does not pass through origin
12	0.987	Linear		

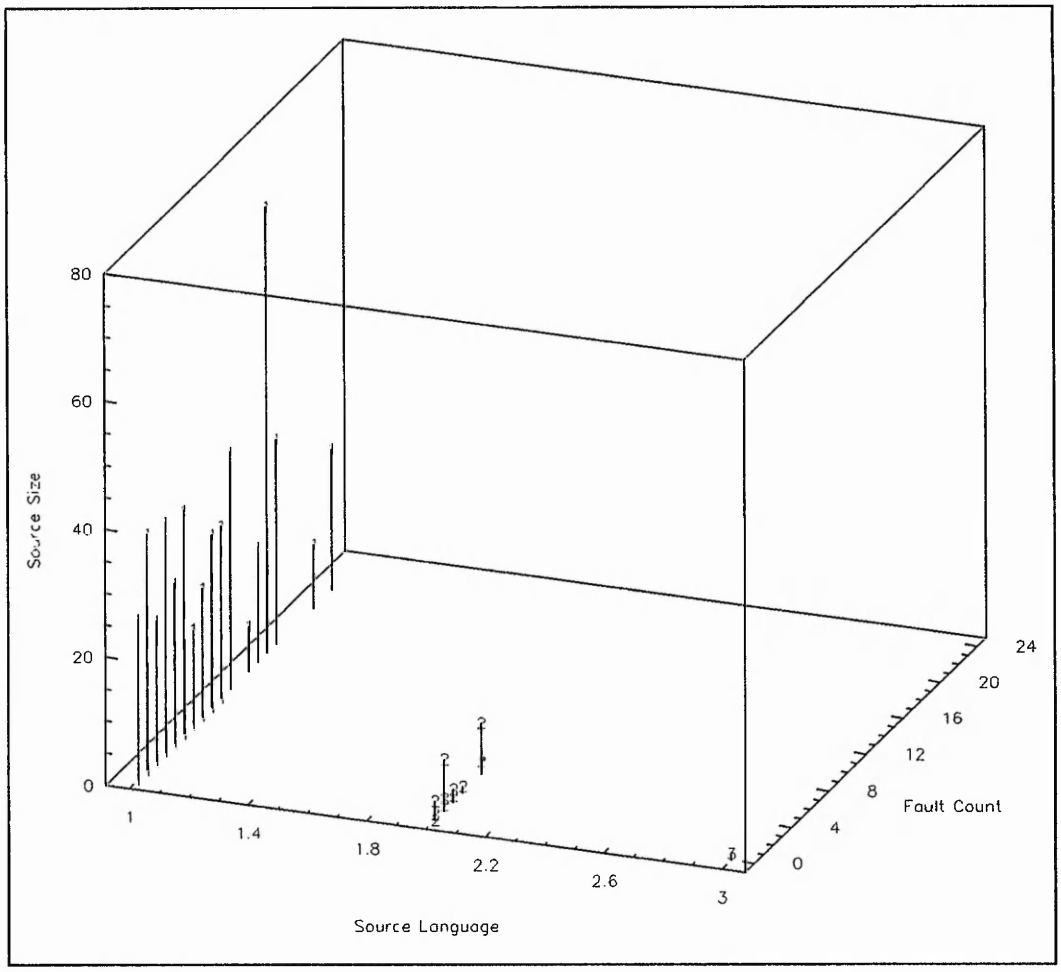
Assuming that the upward curvature and the two lines are due to change in the level of work carried out on the software, then only sources 7 and 12 are unaffected. Sources 9, 10 and 11 are separate from the rest in that they correspond to a more complex regression model. Further analysis using proportional intensity modelling in chapter 7.6.4 showed that this curvature may be modelled by the IBM model (Rosner (1961)). These results are confirmed by the proportional hazards modelling in chapter 6.3 and proportional intensity modelling in chapter 7.6.4.

3.6 RELATIONSHIPS BETWEEN FAULT COUNT AND SOFTWARE ATTRIBUTES

It was decided to incorporate as much information together from the twenty files as delivered by the data supplier into one file. The data on the software attributes such

as source type, size, language, first occurrence and final occurrence in a product, programmer, times between faults, times between failures and the number of faults were incorporated together. A number of plots of these variables were carried out and two are shown below.

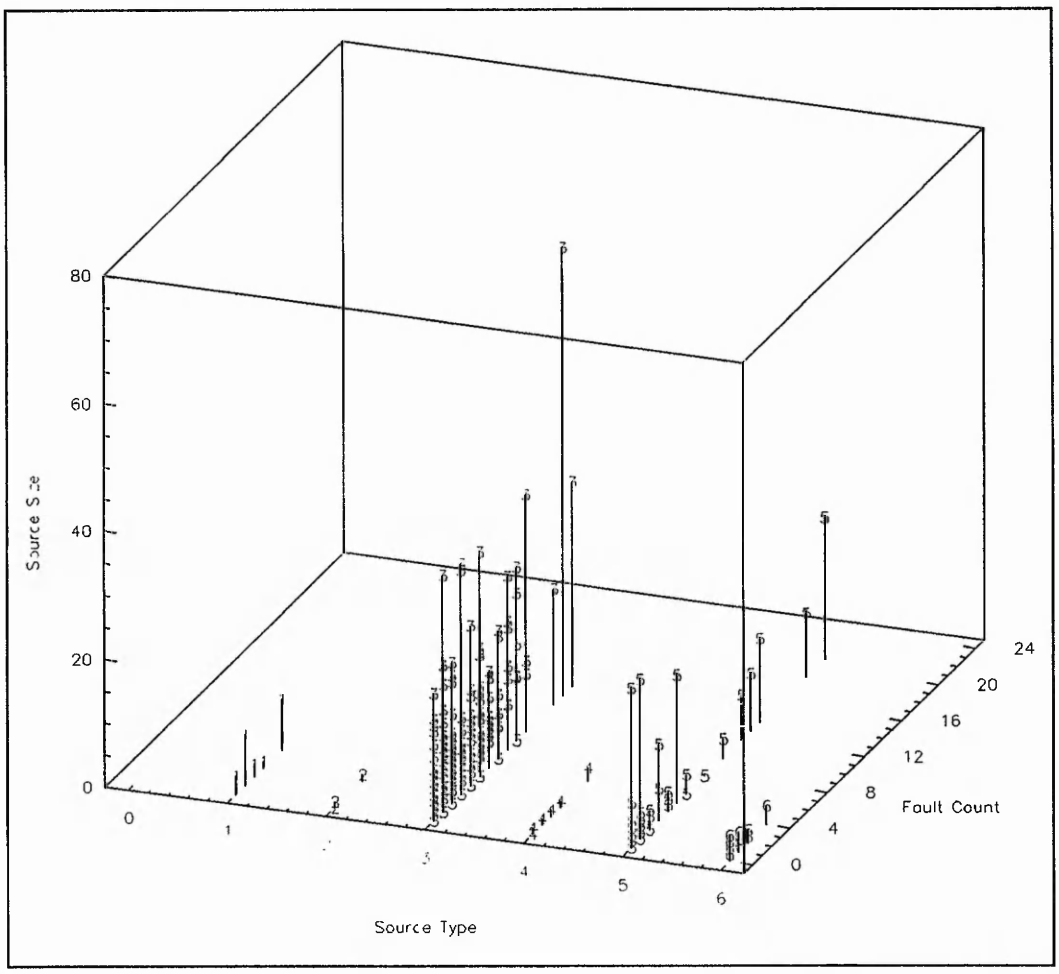
FIGURE 3.6. PLOT OF NUMBER OF FAULTS AGAINST SOURCE LANGUAGE AND SIZE



In the above figure 3.6, it can be seen that most of the sources are in language 1 and that the number of faults (in 4K bytes of code) is not related to size. The reason

for this is probably due to the product not being designed and written from scratch and so no substantial amount of new code was written.

FIGURE 3.7. PLOT OF NUMBER OF FAULTS AGAINST SOURCE TYPE AND SIZE



In this figure, there does not appear to be any significant relationship between source size and number of faults. In both plots, a zero count of faults against zero source size is not shown. The amalgamated file is analysed in

more detail in chapter 7 using discriminant analysis, principal components analysis and log-linear modelling using all the variables.

Further analysis of the software attributes is carried out in chapters 6 and 7.

3.7 SUMMARY

An exploratory approach has been applied to Alvey data set number 3 and relationships between failure count and cumulative time to failure were found. This means that naive application of reliability distribution theory is not immediately applicable. Modelling of a time metric where there is evidence of reliability growth may be carried out by time series or non-homogeneous Poisson processes. The explanatory information available (source type, designation, etc) may be analysed by models which can incorporate structure such as proportional hazards modelling, proportional intensity modelling and multivariate techniques.

4 TIME SERIES ANALYSIS

Time series methods in reliability have been implemented by many authors. The analysis of times to failure (TTF) of software and/or hardware has been documented by Singpurwalla and Soyer (1985), Horigome, Singpurwalla and Soyer (1984), Meinhold and Singpurwalla (1983), Crow and Singpurwalla (1984) and Davies et al (1987). Some references have included the application of traditional linear autoregressive integrated moving average (ARIMA) models of Box and Jenkins (1976) to TTFs. These include Bendell and Walls (1985) and Walls and Bendell (1987).

Time series has been applied to the failure count data of Alvey data set number 3 so that a forecast of when the software is failure free may be made and compared with when the software was actually delivered to the customer. The analyses of the logged data in chapter 4.3 are compared with a PHM specification for the same data in chapter 6.

4.1 STATE OF THE ART OF TIME SERIES

Although 'discoveries' of trend and cyclical features have been made using these time series techniques, the whole area was reviewed in the Alvey SRM project. Nottingham research concludes that invariably almost all the assumptions that are made in applying linear modelling techniques are violated by reliability data, and in particular software reliability data.

4.1.1 BAYESIAN ANALYSIS OF TIME SERIES

Violated assumptions are linearity, normality, constant parameters, change points and outliers, (Davies et al

(1987)). According to Davies et al, alternative, and more flexible model formulations are provided by the Dynamic Linear models and implementable using the BATS package, developed at Warwick University (Harrison and Stevens (1976), (West, Harrison and Pole (1988)). Typically, time between failures or time to failures (TTF's) are described by an observation equation

$$TTF(i) = m(i) + r(i) + v(i)$$

where i is the failure number, m(i) a level parameter that evolves with the failures, r(i) is a set of possible covariates (failure dependent) and v(i) is white noise. The extra flexibility is provided by allowing the evolving nature of m(i) and r(i) to be stochastic. The Bayesian (Kalman filter) recursion allows outliers to be handled/detected automatically, missing values, and user intervention with the model. Nottingham researchers have used these techniques to model the MUSA data sets and Alvey data set 8. The approach also allows flexibility in traditional Weibull and hazard modelling. Some results of the above work have been presented in the 1989 SARSS proceedings by Davies, Naylor and McCollin.

4.1.2 TIME SERIES AND PROPORTIONAL HAZARDS MODELLING

Gamerman and West (1989) explore time series methods within a proportional hazards framework. Following Gamerman and West, they have as a starting point the base-line hazard (from Breslow 1974)

$$\log(h_0(t)) = \alpha_t \qquad t \in I_t = (t_{t-1}, t_t) \quad -\infty < \alpha_t < \infty$$

where each I_t must be specified. Stating that it is not

realistic for α_i to be unrestricted from interval to interval they introduce a model that relates consecutive values of α_i,

$$\alpha_i = \alpha_{i-1} + w_i$$

where w_i (zero mean) determines stochastic movement in h_0 (the movement in h_0 is controlled by the variance of w_i which may be chosen to allow for a range of behaviours).

The next step in the development is to allow the $\underline{\beta}_i$ vector (covariate) to vary with time.

Defining
$\underline{\theta}'_i = (\alpha_i, \underline{\beta}'_i)$

$$\underline{\theta}_i = \underline{\theta}_{i-1} + \underline{w}_i$$

Letting $\underline{z}_i$ be the values of the covariates in I_i, the complete specification is

$$\log(h(t)) = (1, \underline{z}'_i)\underline{\theta}_i \qquad \underline{\theta}_i = (1 \times \alpha_i \; z \times \underline{\beta}_i)' \;, \; t \in I_i$$

However, at present there are a number of questions regarding estimation procedures.

4.2 ANALYSIS OF FAILURE COUNTS

This section describes possible methods of analysing numbers of failures per day of the software product irrespective of the source codes which have failed.
A large portion of the literature on failure analysis in the past has dealt with times between failures, Thompson (1981), and Ascher and Feingold (1984). However, pro-

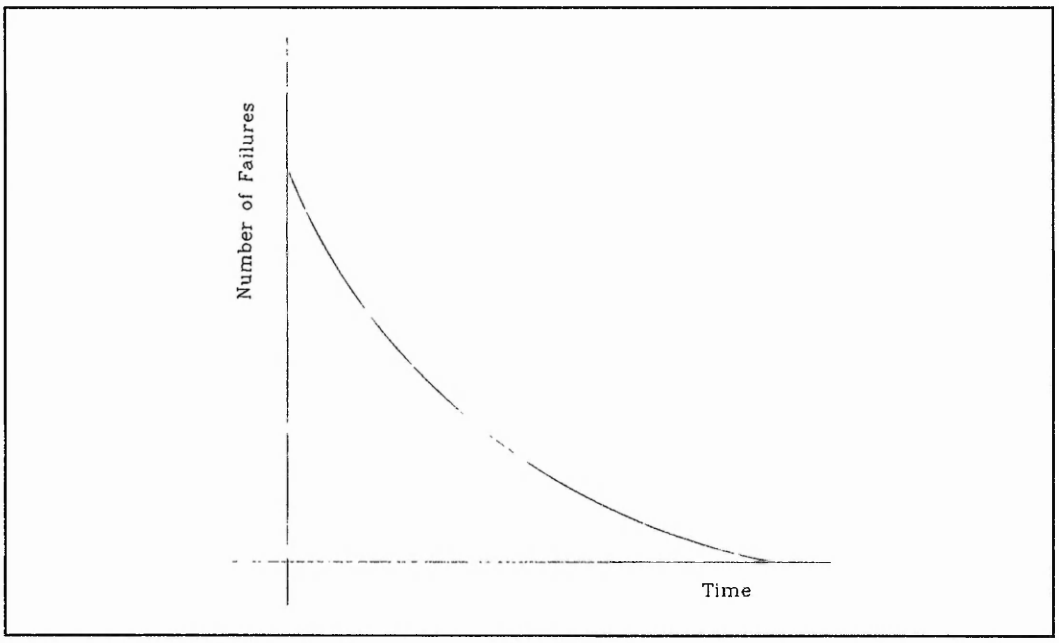
portional hazards modelling has been carried out in the past with a number of failures as a metric and also as a covariate by Kalbfleisch and Prentice (1980) and Lawless (1987). This latter approach is discussed in chapter 7. Ansell and Phillips (1989) have also analysed an event process with covariates.

A paper by Smith and Oren (1980) describes a Nonhomogeneous Poisson Process derived to analyse a large number of failures in a time interval which also may be applicable in this instance.

4.2.1 A THEORY OF SOFTWARE DEVELOPMENT

As a new large software system is developed, the number of failures found should reduce towards zero. Longbottom (1980) figures 5.6, 5.7 and 5.8 show that the reduction in failures over time is approximately exponential and this has been represented in figure 4.1 below.
The plot of number of failures per day of the Alvey data set number 3 is shown in figure 3.3 in this thesis and this compares favourably with figure 4.1.

FIGURE 4.1. EXPONENTIAL PLOT OF DATA AS PER LONGBOTTOM REFERENCE



Suppose we wish to derive a physical model of number of failures in a time interval.

The number of software failures in an interval x_t would be expected to be a constant fraction of the number of failures found in a previous interval x_{t-1} that have remained and not been removed by design change (repeat failures).

Thus ϕx_{t-1} is the number of failures remaining and $(1-\phi)x_{t-1}$ are those removed.

Now add a number of new failures per day C' which is defined as a fixed value C plus a random element α_t so that

$C' = C + \alpha_t$.

Thus

$x_t = C + \phi x_{t-1} + a_t$.

If the software design team were dealing with a large number of faults to repair after each failure (see table 3.4) then the number of failures model above may incorporate terms such as x_{t-2}, x_{t-3}, etc which would indicate a lower failure removal rate. If a number of failures are being left unattended as they have an insignificant effect on the software performance and/or failures are caused by the occurrence of just one software fault, then the model may incorporate a very low level trend of failures. The parameter for this low level trend θ and the ϕ parameters for each of the $x_{t-1}, x_{t-2}, \ldots$ should reduce over time for a software development project so for reliability growth, the values of the parameters ϕ, θ should lie between zero and one.

Also, if failures were being recorded at regular intervals, then a seasonal effect would be expected depending on the rate of compiling and coding.

Based on the above model, a time series approach appears appropriate. Time series represents a set of discrete equidistant points denoted x_t. Time series for counts is not very well developed but has been stated as an analysis method for software reliability data in Mellor and Bendell (1986) page 390. Recent references are by Harvey and Fernandes (1988), McKenzie (1988), Zeger (1988), Holden (1987), Phelps and MacCallum (1989) and Madiedo (1986) who uses the Box-Jenkins approach. The Box Jenkins (1976) approach was used for the analysis presented here as it is well developed, there is commercial software available (MINITAB, Statgraphics) and it is possible to forecast

future counts of failures which is extremely beneficial for determining the optimal release time for the software to the customer.

4.2.2 DERIVATION OF THE BOX-JENKINS APPROACH

The approach to select a suitable time series model is to first identify the structure within the data and suggest one or two candidate models for parameter estimation. After estimation has been carried out, then diagnostic checking takes place to determine goodness of fit of the model to the data.

The Box-Jenkins time series models are known as Autoregressive Integrated Moving Average Models denoted by ARIMA(p,d,q)(P,D,Q)s where

p is the order of the non-periodic autoregressive component
d is the degree of differencing required to remove some types of non-stationarity in non-periodic data.
q is the order of the non-periodic moving average component
P is the order of the periodic autoregressive component
D is the degree of differencing required to remove some types of trend in periodic data.
Q is the order of the periodic moving average component
s is the number of periods before the series repeats itself.

The non-periodic autoregressive component of order p is given by $x_t = \phi_0 + \phi_1 x_{t-1} + \ldots + \phi_p x_{t-p} + a_t$ where the current value of the variable x_t is linearly related to the p previous values of the variable in time and a random disturbance term which follows a normal distribution. The non-periodic moving average term of order q is given by

$x_t = \theta_0 - \theta_1 a_{t-1} - \ldots - \theta_q a_{t-q} + a_t$ which linearly relates the current value of the variable to the q random disturbance terms. Periodic terms of lag s of the form $x_t = \Phi_0 + \Phi_s x_{t-s} + \ldots + \Phi_{Ps} x_{t-Ps} + a_t$ and $x_t = \Theta_0 - \Theta_s a_{t-s} - \ldots - \Theta_{Qs} a_{t-Qs} + a_t$ are the autoregressive and moving average equations respectively. The integrated terms, d and D, define the degree of differencing required to remove certain types of non-stationary trend within the data. Estimation of the parameters is by the method of non-linear least squares using a combination of the Gauss-Newton method and the steepest descent method which is known as Marquardt's compromise (Pankratz (1983)).

The autoregressive term accounts for the failure trend due to a high fault incidence and low removal rate of failures in the physical model and the moving average term can be thought of as the low level trend in the physical model which indicates those failures which have a low priority for repair. The Box-Jenkins model incorporates an additive error term a_t which follows a normal distribution with zero mean and constant variance.

4.2.3 ANALYSIS OF SOFTWARE FAILURE COUNTS PER DAY FOR ALVEY DATA SET NUMBER 3

Initially, a parsimonious model was fitted to the original data set shown in figure 3.3, i.e one with the smallest number of parameters which provides good forecasts. The objective here was to define a physical model for the process in as few parameters as possible and determine the effect of the test strategy on the parameter values. The advantage of using a parsimonious specification is

that few parameter estimates need to be monitored and the model may be readily checked for any changes in the development cycle such as staff shortages, etc.

The values of ϕ_1, θ_1, Θ for an
ARIMA (1 0 1)(0 1 1)7 model were calculated to be 0.9208, 0.652 and 0.8765 respectively written as

$$x_t = C + \phi_1 x_{t-1} + \theta_1 a_{t-1} + x_{t-7} - \phi_1 \Theta x_{t-8} - \Theta a_{t-7} + a_t$$

which shows that the number of failures per day is decreasing and is tending towards zero. In this case, the constant value, C, was removed as it was significantly close to zero. The value of C is important to monitor in software development as it should be close to zero for perfect debugging. In this case, the debugging strategy appears to be effective.

However, the test phase accounts for most of the exponential trend in the data as seen when the data is plotted in figure 3.3. The analysis of the test phase data (110 values) produced three possible parsimonious models. The three candidates were
ARIMA (0 1 1)(0 1 1)7,
ARIMA(1 0 1)(0 1 1)7 and
ARIMA(2 0 0)(0 1 1)7 written as

$$x_t = C + x_{t-1} + x_{t-7} - x_{t-8} - \theta a_{t-1} - \Theta a_{t-7} + \theta\Theta a_{t-8} + a_t$$

$$x_t = C + \phi_1 x_{t-1} + \theta_1 a_{t-1} + x_{t-7} - \phi_1 \Theta x_{t-8} - \Theta a_{t-7} + a_t$$

and

$$x_t = C + \phi_1 x_{t-1} + \phi_2 x_{t-2} + x_{t-7} - \phi_1 x_{t-8} - \Theta a_{t-7} + a_t \ .$$

It has been shown by Harvey (1989) page 74 that these are

the only candidates which model this exponential trend. Two conditions that should be satisfied for the parameter estimates are stationarity and invertability of the estimates where more weight should be placed on more recent observations.

As long as

$\mathrm{mod}\,\hat{\phi}_1 < 1, \quad \mathrm{mod}\,\hat{\theta}_1 < 1, \quad \mathrm{mod}\,\hat{\Theta}_1 < 1$
and

$\phi_2 \pm \phi_1 < 1, \quad \theta_2 \pm \theta_1 < 1, \quad \Theta_2 \pm \Theta_1 < 1$
are true, then the conditions hold. For all the model fits in this chapter, these conditions have been checked and validated.

The selection of the best model may be carried out by the Modified Box-Pierce statistic (see later) available with the Minitab software package or by changing the order of each of the p, q, P, Q terms by one and checking for a closer fit. Alternatively, a check on the normality of the residuals may be investigated. Each of these methods were implemented.

This is one possible approach to the analysis. It may be useful to assume that the testing was not finished at the end of the test phase so that possibly other structure may be highlighted.

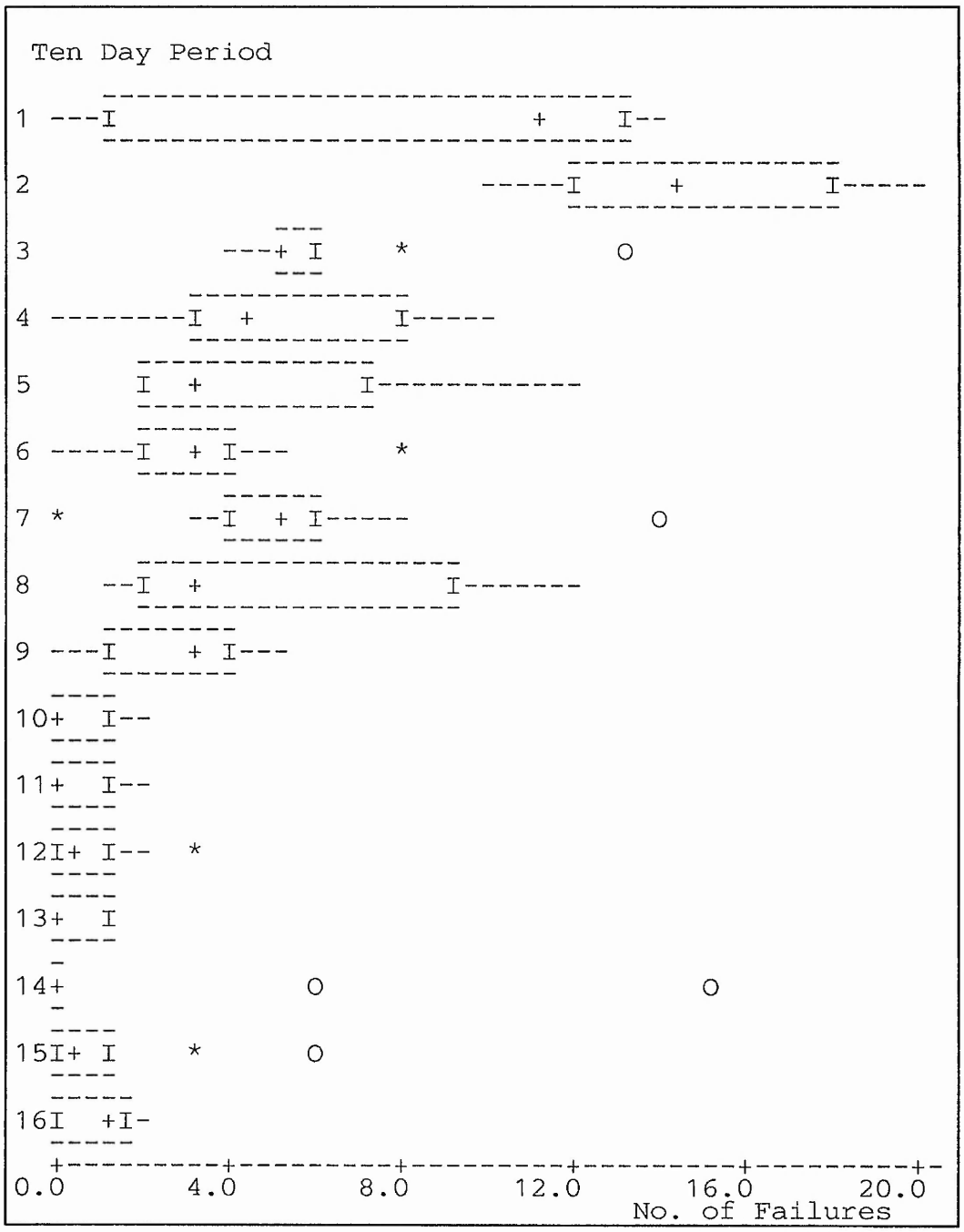
4.2.4 EXPLORATORY APPROACH

The seasonal effect was seen to be smaller at the weekends and this was having the effect of forecasting smaller values of the number of failures per day on the weekends

than the rest of the week. The possible reasons for this is that failures were left until Monday to record or the staff were not working at the weekend. To remove the effect of not working at the weekends, two approaches may be attempted. The first is to average out the Saturday, Sunday and Monday data and the second approach is to remove all data from the weekends from the data set. This is justified by the analysis of failure free days, most of which occurred on weekends. The second approach is documented below. This reduction in the data set did not remove the seasonality; it only changed it from a 7 day effect to a 5 day effect.

Before proceeding with further time series analysis, the data may be manipulated to show the structure that is being looked for. In the initial time series model, there was a 7 day seasonal effect. This will now be a 5 day effect since the weekends have been removed. Thus, if box and whisker plots are taken for each set of ten values, this will show the non-seasonal structure which the time series will model. The reason for showing the box and whisker plot is that outliers will be highlighted (* and O on the figure). This justifies the taking of logs of the data to remove the outliers so that further time series analysis may be carried out as described in chapter 4.3 with comparative PHM of the data in chapter 6.

FIGURE 4.2. BOX AND WHISKER PLOT OF ALVEY 3 WEEKDAY FAILURE COUNT DATA



This plot shows that the spread of the data is approximately constant from the third ten day period until after the ninth ten day period when only zeros and ones were occurring. After the first two periods, the medians of the periods 2 to 6 are reducing linearly and after a jump in period 7, the median for periods 6, 8 and 9 is constant at three failures until period ten (or after one hundred weekdays) which is one hundred and forty days from the project start.

Delivery to the customer occurred on day 110 which would account for the higher spread of data in period eight and the high outlier in week seven when there would be a higher workrate to provide a failure free product by the delivery date. The zero in week seven is a bank holiday. Hence most of the structure in the data occurs before the 90th weekday. This is about 131 days into the collection.

The reasons for the software failures reducing to a constant level of one or zero after period nine may be caused by one or more of the following reasons:

the reduced severity of the customer environment compared to the test environment of the supplier

the reduced effort being applied to find and/or report failures

the effect of removing most of the faults which cause failures.

An implication of these is the loss of the autoregressive term in the Box-Jenkins model (the trend in the physical

model).

4.2.5 FIVE DAY SEASONAL TIME SERIES

A time series analysis was applied to the amended data without Saturday's and Sunday's observations and two five day seasonal models were found. These were ARIMA(2 0 0)(0 1 1)5 : $x_t = C + \phi_1 x_{t-1} + \phi_2 x_{t-2} + x_{t-5} - \phi_1 x_{t-6} - \Theta a_{t-5} + a_t$ and ARIMA (1 0 1)(0 1 1)5 : $x_t = C + \phi_1 x_{t-1} + \theta_1 a_{t-1} + x_{t-5} - \phi_1 \Theta x_{t-6} - \Theta a_{t-5} + a_t$.

In physical terms, model (1) is the number of failures per day is a constant multiplied by the number of failures on the previous day + a constant multiplied by the number of failures on the second day before + a 5 day seasonal effect i.e., consecutive Monday's number of failures are related,
consecutive Tuesday's number of failures are related, etc. The weekly seasonal effect is probably due to the method of failure report collection and the autoregressive effect is due to the effect of repairing failures on finding them. The estimates of each of the ϕ, θ, Θ parameters were each less than unity so that reliability growth was taking place.

Model (2) was similar to model (1) except that the second autoregressive component was replaced by a first order moving average. The moving average denotes a reduction in the average number of failures over time being present in the data, the failure count for the lower severity failures (in terms of fault count) within the physical model.

The models were fitted to the data over a number of periods so that the parameter estimates could be monitored. These are listed in the following tables.

TABLE 4.1. TABLE OF THE ARIMA (2 0 0)(0 1 1)5 MODEL VALUES

No. of values	ϕ_1	ϕ_2	Θ	C	Goodness of Fit
80	.3425 (2.99)	.3094 (2.7)	.8906 (11.81)	-0.109 (-1.76)	15.51 71 fit bpg noc
90	.3197 (3.06)	.3149 (3.04)	.9458 (14.08)	-0.153 (-3.9)	13.05 81 rss bpg p
100	.3325 (3.33)	.3026 (3.04)	.9642 (17.48)	-0.169 (-5.74)	11.71 91 fit bpg p
110	.3223 (3.41)	.2989 (3.18)	.9516 (17.6)	-0.1666 (-6.23)	10.66 101 fit bpg p
120	.3227 (3.59)	.2990 (3.34)	.9591 (20.5)	-0.158 (-7.13)	9.76 111 rss bpg p

TABLE 4.2. TABLE OF THE ARIMA (1 0 1)(0 1 1)5 MODEL VALUES

No. of values	ϕ_1	θ_1	Θ	C	Goodness of Fit
80	.8211 (6.6)	.4405 (2.33)	.8902 (11.21)	-0.0514 (-1.52)	16.02 71 fit bpg noc
90	.8148 (6.73)	.4563 (2.56)	.9245 (13.54)	-0.0682 (-3.0)	13.93 81 fit bpg p
100	.7900 (6.46)	.4261 (2.41)	.9635 (18.01)	-0.0857 (-5.54)	12.16 91 fit bpg p
110	.8034 (7.19)	.4451 (2.73)	.9553 (17.56)	-0.0813 (-5.89)	10.94 101 fit bpg p
120	.8112 (7.95)	.4541 (3.00)	.9621 (20.72)	-0.0761 (-6.84)	10.06 111 fit bpg p

The meaning of the terms in the goodness of fit column are
First number - residual Mean Square
Second number - degrees of freedom
fit - Relative change in each estimate less than 0.001
rss - Unable to reduce sum of squares to meet convergence criterion

bpg - Not significant Modified Box-Pierce statistic
p - t values for parameter estimates >2
noc - constant is not significant in the model.

Most of the models are reasonable fits to the data, (see fit against rss in column 6), the number in the brackets after each parameter estimate being the Student's t values which should lie beyond ±2 (see p in column 6) approximately 95% of the time if the estimate is correct.

The parameter estimates remain reasonably constant over 40 observations although the constant is not significant at 80 values for both models (the significance value is in the brackets in the tables). The criteria for rejecting a parameter from the model is to compare the modulus of the significance value with 2 and if the significance value is less than 2, then reject the parameter from the model specification (Pankratz (1983)).

To forecast when no failures per day had been reached, the first occurrence of a negative number and when two consecutive negative numbers were noted. These forecasts become less accurate the further from the end of the data they are. Reasonable forecasts may be made up to two cycles away, i.e. ten weekdays. The seasonal effect of both models produced a first negative number with the rest of the week being positive counts. Looking for two consecutive negative numbers shows the other days coming into effect.

The forecasts $x_{t+l} = \hat{x}_t(l)$ are available within the Minitab software and may be calculated one step ahead at a time by replacing unknown values using the derived ARIMA

specification and assuming forecasted errors are zero. A 95% confidence interval is given by $\hat{x}_t(l) \pm 1.96\hat{\sigma}(a_t(l))$ assuming the forecasts are normally distributed.

TABLE 4.3. FORECASTS FOR THE ARIMA (2 0 0)(0 1 1)5 MODEL

No. of Values	First Forecast	First zero	Second consecutive zero
80	80	-	-
90	90	98	113
100	90	98	103
110	100	103	108
120	90		112

TABLE 4.4. FORECASTS FOR THE ARIMA (1 0 1)(0 1 1)5 MODEL

No. of Values	First Forecast	First zero	Second consecutive zero
80	80	-	-
90	90	108	128
100	90	108	118
110	100	108	113
120	90	108	123

For both models (1) and (2), the forecasts were nearly constant at six for 80 observations. Each successive

forecast for each model produced a consistent value for the occurrence of the first zero. Model (1) was more consistent for the forecasts of two consecutive failure free days with about 110 week days being the average forecast for model (1). The value of 110 weekdays equates to 154 days into the data collection which is 44 days after delivery to the customer.

4.3 TIME SERIES OF LOGGED DATA

As can be seen from the box and whisker plot of figure 4.2, the data has a number of outliers and neither of the time series models discussed stand out as best. It was decided to take logs of the number of failures per weekday plus one to reduce the variation and to show that the resulting model is similar to a PHM formulation in chapter 6. The problem of zero failures is alleviated by the addition of one to each data point. Hence $z_t = \ln(x_t + 1)$.

The best model fit for the logged data was the ARIMA(0 1 1)(0 1 1)5 model
which is

$$z_t = C + z_{t-1} + z_{t-5} - z_{t-6} - \theta a_{t-1} - \Theta a_{t-5} + \theta\Theta a_{t-6} + a_t.$$

After 100 days, the parameter values were estimated and a two week forecast was calculated. The Minitab software package was used to test the adequacy of the model fit.

The estimates of the parameters are as follows :

TABLE 4.5. TABLE OF ARIMA (0 1 1)(0 1 1)5 MODEL ESTIMATES FOR THE LOGGED DATA

Type	Estimate	St. Dev.	t-ratio
θ	0.6561	0.0805	8.15
Θ	0.9282	0.0604	15.37
C	-0.00552	0.003129	-1.76

where θ is the MA (1) term and Θ is the seasonal MA (5) term. The constant term is within two standard deviations of zero but the model gives better forecasts when it is included.

The Residual Mean Square is an estimate of the variance of the errors $\hat{\sigma}^2 = \sum_{t=1}^{n} \hat{a}_t^2/(n-m)$

where m is the number of coefficients being tested,
n is the number of observations
and
$\hat{a}_t$ are the estimated errors between the observed and the expected values.

For this model, $\hat{\sigma}^2 = 0.4445$ on 91 degrees of freedom. This value is the lowest for any of the models which were tried.

The Modified Box-Pierce (1970) chi-squared statistic is given as

$$Q_m = n(n+2)\sum_{k=1}^{m} r_k^2/(n-k)$$

where m is the number of coefficients being tested,

n is the number of observations in the series
and r_k $1 \leq k \leq m$ are the sample autocorrelations given by

$$r_k = \frac{\sum_{t=1}^{n-k}(z_t-\bar{z})(z_{t+k}-\bar{z})}{\sum_{t=1}^{n}(z_t-\bar{z})^2}$$

where $\bar{z} = \sum_{t=1}^{n} \frac{z_t}{n}$.

The null hypothesis of the test is that the population autocorrelations for the lags is equal to zero with the alternative that they are not equal to zero.

The Minitab printout gives values of the statistic at lags 12, 24, 36 and 48. The expected value of the statistic is X^2 and a quick ad hoc test is that if values of the statistic are between zero and twice the degrees of freedom in the Minitab printout, the statistic is not significant and the model is a reasonable fit to the data. This is the case for this data.

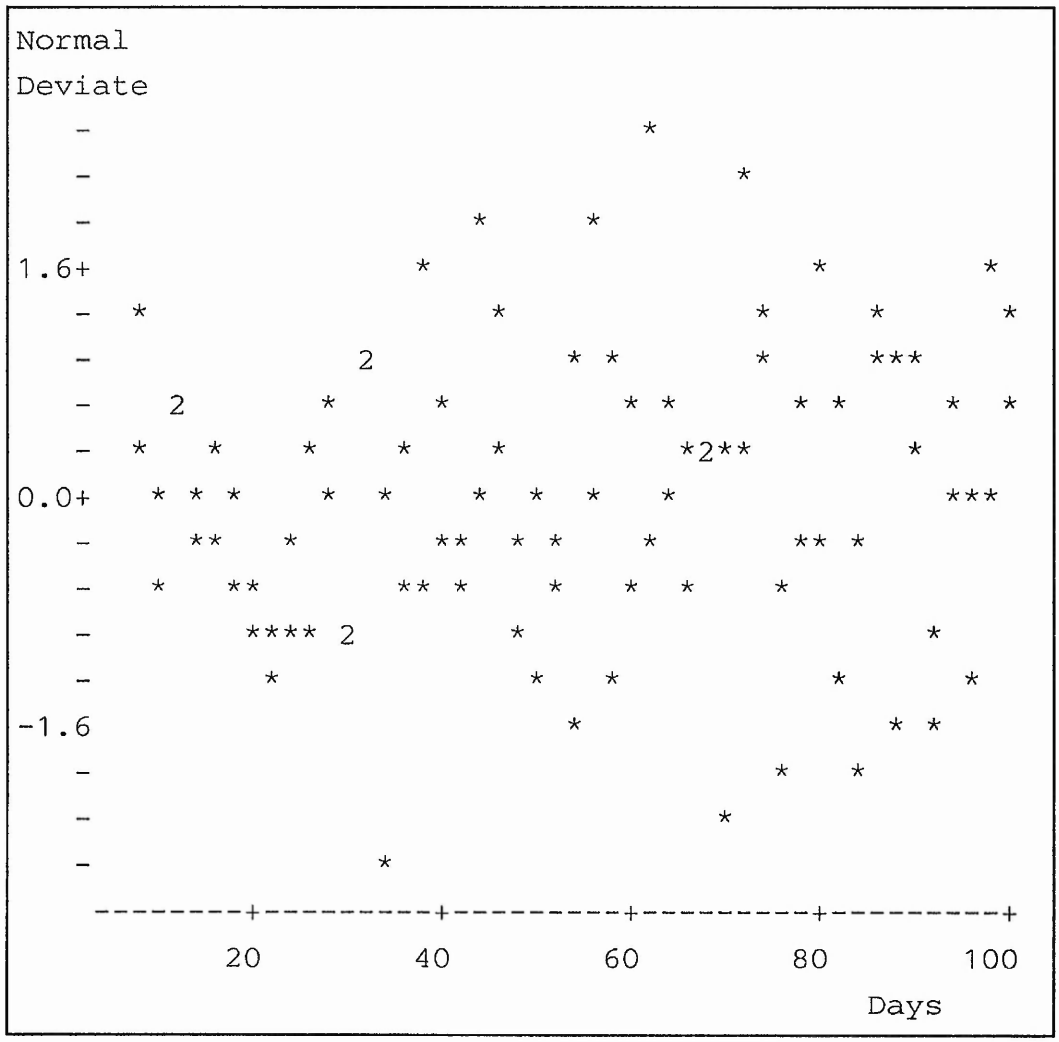
TABLE 4.6. TABLE OF BOX-PIERCE STATISTICS

Lag	12	24	36	48
X^2	14.3(DF=10)	23.3(DF=22)	42.2(DF=34)	54.0(DF=46)

4.3.1 RESIDUAL PLOTS

Various plots of the residuals are obtainable by manipulating simple Minitab commands and these are presented below. As can be seen, the model is a good fit to the data. It is also unique in it's parsimony.

FIGURE 4.3. PLOT OF THE NORMALISED RESIDUALS AGAINST WEEKDAY
(FIRST 6 VALUES REMOVED DUE TO DIFFERENCING)



This plot should show a random spread about a mean of zero with spread between ±2. This may be checked by plotting the autocorrelation function and the partial autocorrelation function of the residuals and both of these plots (not shown) show no appreciable underlying structure.

FIGURE 4.4. TIME SERIES PLOT OF THE RESIDUALS

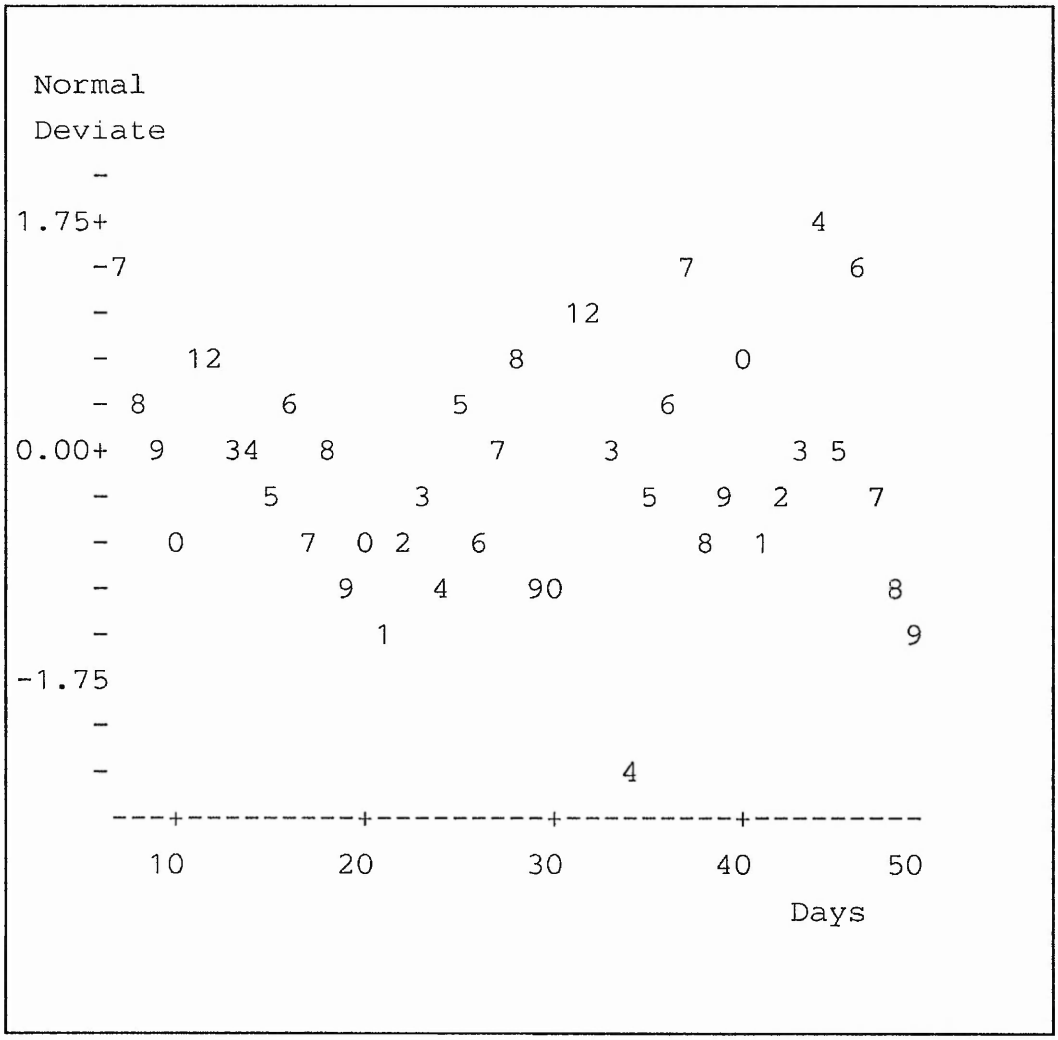
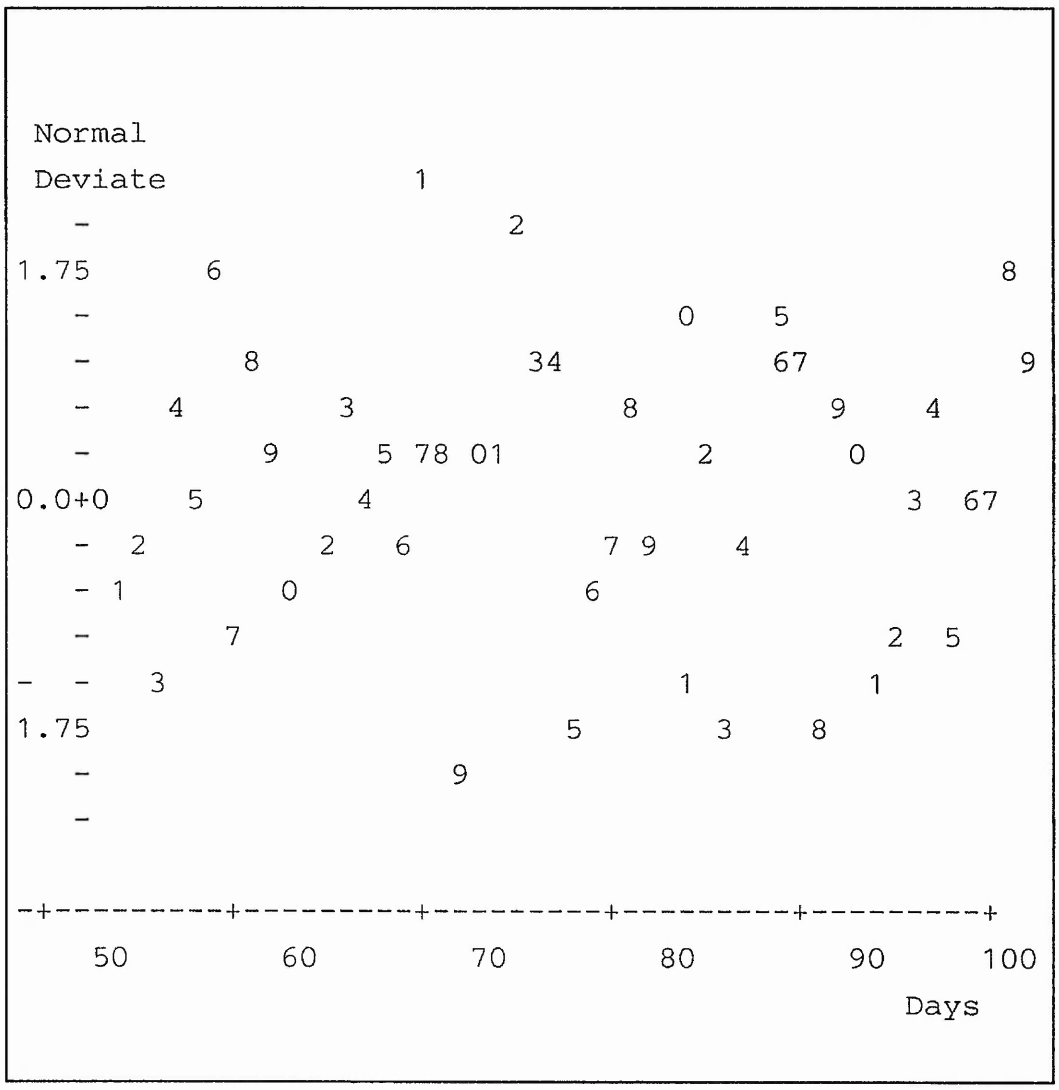
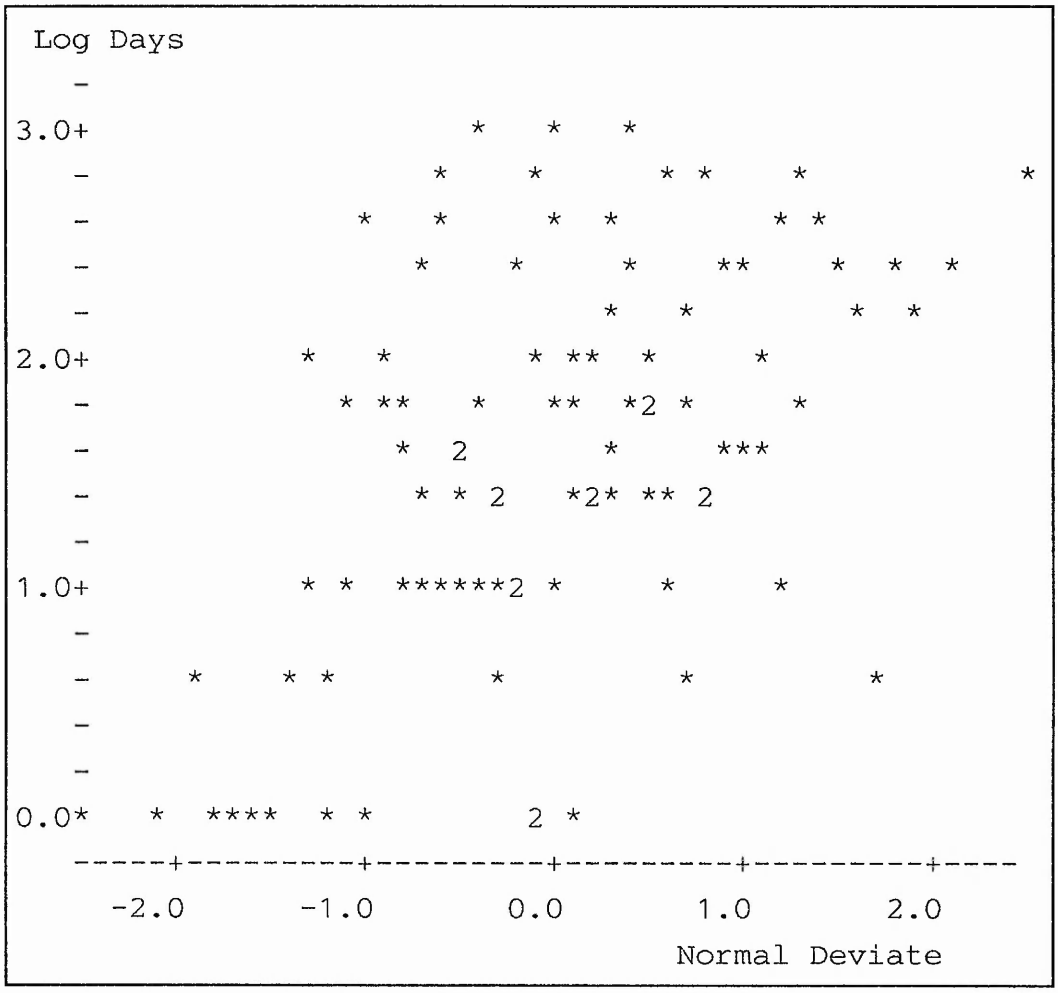


FIGURE 4.4. (CONTINUED) TIME SERIES PLOT OF THE RESIDUALS



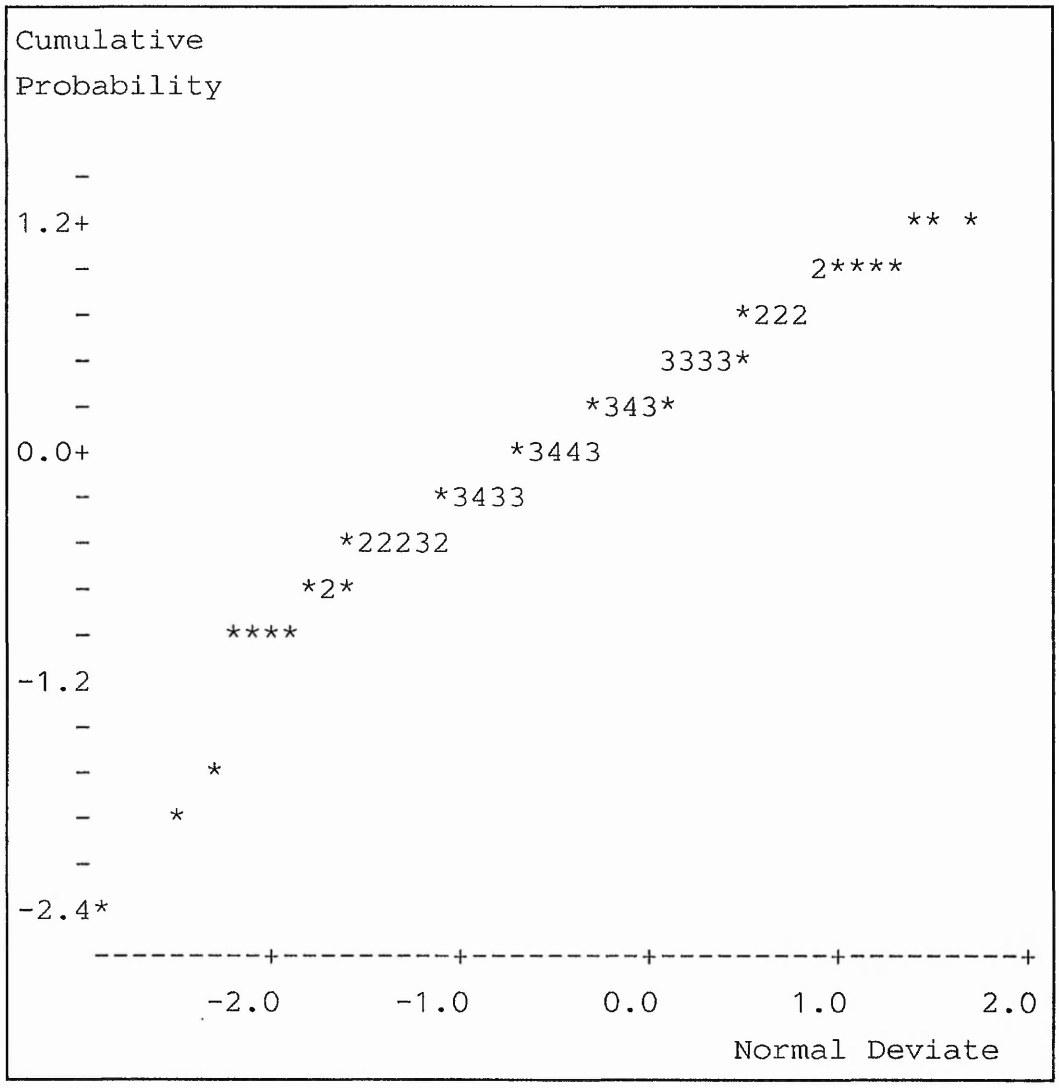
This is a plot of the normal deviate value of the residuals against days with each data point numbered and the plot should show no structure. There does appear to be a seasonal trend over thirty days however this trend dies away quickly and cannot be resolved into a model.

FIGURE 4.5. PLOT OF THE ORIGINAL DATA AGAINST THE RESIDUALS



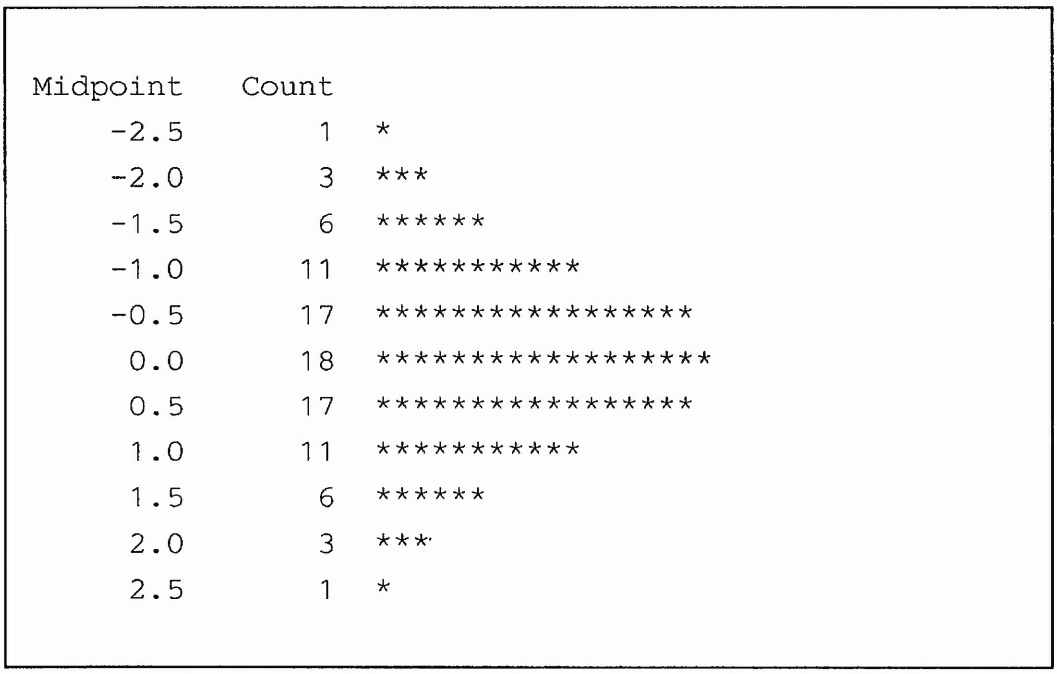
This plot should show no relationship between the two variables or else there is a serial correlation present. A correlation coefficient of the two variables was estimated and was close to zero. So we may conclude there is no significant serial correlation.

FIGURE 4.6. NORMAL PROBABILITY PLOT OF THE RESIDUALS



This should show a straight line if the normal distribution is appropriate for the residuals. This appears to be the case apart from the three lowest points.

FIGURE 4.7. BAR CHART OF THE RESIDUALS



This is a symmetrical plot about zero with most values lying between ±3 which confirms that the residuals are normally distributed.

4.3.2 FORECASTS FOR LOGGED DATA

The forecasts for the original data is performed by the transformation $\hat{x}_t(l) = e^{\hat{z}_t(l)+0.5\hat{\sigma}^2 a'_t(l)} - 1$ where $\hat{x}_t(l)$, $\hat{z}_t(l)$ are the estimated values of the original data and the logged data each forecasted one step ahead and $\hat{\sigma}^2 a'_t(l)$ is the estimated forecast-error variance of the log forecast. The 95% confidence intervals for the original data may be calculated from $\hat{w}_t(l) = e^{\hat{y}_t(l)} - 1$ where $\hat{w}_t(l)$, $\hat{y}_t(l)$ are the upper (or lower) confidence values for the original and the logged data respectively.

TABLE 4.7. FORECASTS FROM PERIOD 100

Period	Forecast	Lower 95% Value	Upper 95% Value	Change	95% Value	Actual Value
101	1.0218	-0.562	4.981	-61.7%	130%	1
102	0.8350	-0.641	4.700	-12.3%	138%	0
103	0.4900	-0.735	3.842	-23.5%	145%	2
104	1.0200	-0.674	5.843	27.8%	152%	0
105	1.3477	-0.654	7.267	12.4%	158%	0
106	0.4815	-0.783	5.208	-37.7%	168%	1
107	0.5166	-0.822	4.841	-12.9%	174%	0
108	0.2295	-0.869	3.904	-24.0%	180%	0
109	0.6641	-0.838	5.859	3.2%	187%	0
110	0.9310	-0.829	7.209	12.2%	193%	1
111	0.3755	-0.892	5.090	-38.6%	201%	0
112	0.2883	-0.912	4.679	-13.4%	208%	2
113	0.00773	-0.935	3.729	-24.5%	214%	1
114	0.3622	-0.920	5.563	26.7%	220%	1

TABLE 4.8. FORECASTS FROM PERIOD 105

Period	Forecast	Lower 95% Value	Upper 95% Value	Actual Value
106	0.3948	-0.699	3.135	1
107	0.2144	-0.762	2.769	0
108	0.1238	-0.799	2.641	0
109	0.3600	-0.778	3.584	0
110	0.5061	-0.774	4.268	1
111	0.2177	-0.839	3.479	0
112	0.0589	-0.872	3.038	2
113	-0.0212	-0.892	2.861	1
114	0.1832	-0.881	3.818	1
115	0.3087	-0.879	4.490	1

TABLE 4.9. FORECASTS FROM PERIOD 110

Period	Forecast	Lower 95% Value	Upper 95% Value	Actual Value
111	0.3727	-0.688	2.966	0
112	0.1118	-0.769	2.357	2
113	0.0389	-0.802	2.267	1
114	0.1646	-0.796	2.805	1
115	0.3862	-0.776	3.694	1
116	0.1014	-0.842	2.922	3
117	-0.1091	-0.883	2.287	0
118	-0.1687	-0.899	2.171	0
119	-0.0693	-0.897	2.662	0
120	0.1063	-0.887	3.484	0

Forecasts from a specific weekday for three periods from 100 weekdays into the project showed that the number of failures would be practically zero on day 113. Forecasts from the same weekday for two periods from 105 and 110 days for two periods showed that day 113 using the criteria of first zero and day 118 using the criteria of two consecutive zeros are the days when zero failures are forecasted. These results confirm the previous analysis using the original data.

The forecasts are highly dependent on the model fit so a low value of the residual variance is imperative. The percent changes in the forecasts of the actual values with their confidence interval is given for the 100 day data

and as can be seen the changes are drastic. The reasons for this are that the original data is discrete and the variance is large thus a change from zero to one failure in the original data is having a significant effect on the forecasted values. The theory on forecasting is taken from Pankratz (1983).

When explanatory information is available, the additional variation in the models may be reduced by incorporating it into a proportional hazards type time series model of Gamerman and West. Finally, it must be stressed that the forecast is dependent on there being no change in environmental conditions (which there probably were here) as the software had been delivered by the time zero failures was forecasted.

4.4 ANALYSIS OF THE LIVE PHASE

An analysis of the phase after delivery to the customer, i.e. from 110 to 220 days, the best model fit for the untransformed data is ARIMA (0 1 1)(0 1 1)7 which is

$$x_t = 0.0074 + x_{t-1} + x_{t-7} - x_{t-8} - \theta a_{t-1} - \Theta a_{t-7} + \theta\Theta a_{t-8} + a_t$$

where $\theta = 0.9267$, $(t = 16.39)$ and $\Theta = 0.8995$, $(t = 11.88)$.

Therefore the number of failures per day should decrease to a constant value of mainly zeros and a few ones, to maintain the constant of 0.0074, if no new product versions are installed.

As has been seen, time series models the software development and data collection process. The values of the time series coefficients may be influenced by the levels and types of software development testing. The test levels of hardware reliability growth programmes influence

the shape parameter of the Duane or Weibull intensity model as described in MIL-HDBK-189 (1989). The values of the shape parameter for a NHPP with Weibull intensity are given in Jaaskelainen (1982) and O'Connor (1981) for specific development strategies in hardware reliability growth testing. Values of the time series coefficients may be defined likewise for specific strategies of software development.

In conclusion, time series has been used to model the data collection process to highlight trend and serial correlation and may be used for forecasting of when the software is fault free. These forecasts were highly dependent on how well the data fitted the ARIMA model specification. In chapter 6, further analysis of this data set of failure counts is carried out using PHM to determine the effect of the seasonal component and trend on the hazard rate.

5 SOFTWARE RELIABILITY THEORY

Since Jelinski and Moranda published their paper on software reliability modelling in 1972, there have been put forward many software reliability modelling formulations. Mellor (1987) classified these into either structural or black-box models where the first considers the internal structure of the system whereas the second considers the system failure process. The rest of this section will cover the black-box methods. Mellor further classifies black-box methods as either interfailure time models and those due to fault manifestation. The most well-known of the interfailure time models is the Littlewood-Verrall model (1973). Again, this class of models is not considered here. The fault manifestation class of models are order statistics processes such as the order statistics process described in chapter 1.4, Gray (1986), Miller (1986) and Keiller and Miller (1991).

General Order Statistic Models assume each time to manifestation of a fault follows a distribution, for example, the exponential distribution gives rise to the class of Exponential Order Statistics (EOS) models. The times to failure of the whole system are then order statistics from an independent distribution. The majority of software reliability models are EOS models and the following assumptions (taken from Mellor (1987)) apply.

The system contains a set of faults each of which cause a single failure independently. Each fault has it's own rate and on manifestation is immediately and perfectly removed from the system. The failures occur as a Poisson process.

Miller (1986) and Thompson (1988) have shown that an EOS model, a NHPP with appropriate rate and the pdf of the nth interval are indistinguishable from one another. For example, the Jelinski-Moranda model defined with a total number of faults, a, each with an independent exponential pdf

$$f(w_i) = b e^{b w_i}$$

and hazard rate b is equivalent to a Poisson process with intensity
$\lambda(t_i) = abe^{-bt_i}$ (the Goel-Okumoto model)

if the number of faults is Poisson distributed with mean a.

This was shown by Lewis and Schedler (1976).

The pdf $f_i(t) = (a-i+1)be^{-(a-i+1)bt}, \quad 1 \le i \le a$ defines the same model.

Another instance is the Littlewood (1981) model where the fault distribution is a gamma random variable which may be viewed as a NHPP (see later) if the number of faults is Poisson distributed.

It is proposed to use the NHPP formulations for each of the well-known models so that they may be classified in a proportional hazards framework. The integration of models into classes has been carried out before by Gray and Miller (EOS models), Langberg and Singpurwalla (1985) (shock models) and Kremer (1983) (birth-death models). The advantage of the PHM approach is that extra covariate information may be included such as the effect of a design

change (see Davies et al (1987)). The PHM formulations derived below model most of the NHPP's used in software and hardware reliability irrespective of whether the number of failures is finite or not as time tends to infinity. The main advantage of the PHM formulations over NHPP's is that a diagnostic approach to reliability data modelling may be used. If the covariates or hazard functions which go to make up the NHPP's are not significant in the PHM formulations, then those NHPP's are not appropriate to the data under consideration.

5.1 THE NON-HOMOGENEOUS POISSON PROCESS

The Non-homogeneous Poisson Process (NHPP) is used to model cumulative times to failure of repairable systems.

The expected number of failures in the period $(0,t_i)$ is given by $E(N(t_i)) = M(t_i)$ where $M(t_i)$ is known as the mean value function where i is the actual number of failures.

A review of NHPP's is given below. They are classified as those where the expected number of failures in time t_i, $E(N(t_i))$, is bounded and used mainly for software reliability modelling and those which may be used for hardware and software reliability where $E(N(t_i))$ may increase without limit.

TABLE 5.1. TABLE OF NON-HOMOGENEOUS POISSON PROCESSES

Model	$M(0)$	$M(\infty)$	$\lambda(0)$	$\lambda(\infty)$
Weibull Model	0	a	∞, $c<1$ ab, $c=1$ 0, $c>1$	0
Littlewood NHPP	0	a	abc	0
Musa Basic Model	0	a/b	a	0
Ohba	0	a	$ab/(c+1)$	0
Goel-Okumoto	0	a	ab	0
S-Shaped	0	a	0	0
Duane	0	∞	0	∞, $b>1$ a^{-1}, $b=1$ 0, $b<1$
Cox-Lewis	0	∞	e^{a}	∞
IBM Model	0	∞	c+a	c
Bounded Intensity Model	0	∞	0	b
Logarithmic Model	0	∞	a	0
Square Root Model	0	∞	∞	0

5.2 THE RELATIONSHIP BETWEEN NHPP'S TO PHM

A number of proportional hazards models (PHM) are formulated which incorporate some of these NHPP's notably the binomial and Poisson type exponential models. References which describe the PHM approach are (Cox (1972) Kalbfleisch and Prentice (1980), Lawless (1982) and Cox and Oakes (1984)). The application of PHM within a software context has been undertaken by Nagel and Skrivan (1981), Font (1985), Wightman and Bendell (1986), McCollin, Bendell and Wightman (1989) and Davies et al (1987). Also, under the Alvey Software Reliability Modelling Project, Nottingham undertook the analysis of a number of software reliability data sets using PHM.

Note that other explanatory variables $z_2,\ldots,z_n$ can be used to model (in the same model) the effects of other factors thought to influence the performance of the software.

The incorporation of NHPP's into the PHM formulation has been shown from the expression $h(w_i/t_{i-1})=\lambda(t_{i-1}+w_i)$ taken from Musa (1987), page 260 from which we may relate the intensity function to the Proportional Hazards Model (PHM) where w_i and t_{i-1} are the waiting time since last failure and cumulative time to the (i-1)th failure respectively. Provided the expression may be factorised to $\lambda(t_{i-1}+w_i)=\lambda(t_{i-1})\lambda(w_i)$ so that there is a term with cumulative time to failure and a term with waiting time. By splitting the intensity function into these two separate terms, we may relate the cumulative time term to a covariate structure and the waiting time to failure term to the baseline hazard function in the PHM formulation given by

$$h(w_i;z_1,\ldots,z_n)=h_0(w_i)e^{(\beta_1 z_1+\ldots\ldots+\beta_n z_n)} \qquad \ldots(1).$$

The procedure for parameter estimation is to first calculate the covariate values $\beta_1,\ldots,\beta_n$ and then use these to estimate the other parameters within the baseline hazard function.

5.3 DESCRIPTION OF MODELS FOR SOFTWARE RELIABILITY GROWTH

5.3.1 BINOMIAL TYPE MODELS OF THE EXPONENTIAL CLASS

Wightman, McCollin and Dixon (1991) considered a PHM formulation which allows binomial type models of the exponential class (as classified by Musa et al (1987)) to be incorporated within a proportional hazards framework. The exponential part of the classification refers to the failure distribution of each fault (assumed to be common) with the binomial part referring to the distribution of the number of faults experienced by time t_i. Examples of this type of software reliability model are Jelinski-Moranda (1972) and Shooman (1972).

The formulation of the proportional hazards model for the binomial type models of the exponential class is as follows. Following Musa et al (1987), page 276, the source hazard rate for this class of model is

$$h(w_i/t_{i-1}) = (a-i+1)b$$

where a is the total number of faults present at time zero,
b is the constant value of the hazard for each fault,
w_i is the time from the (i-1) th failure, with t_0 =0, i.e., the waiting time
t_{i-1} is the time of the (i-1) th failure.

This may be rewritten as

$$h(w_i/t_{i-1}) = ab\left(1 - \frac{(i-1)}{a}\right).....(2)$$

The proportional hazards formulation with the metric (w_i in equation (1)) taken as time since last failure is

$$h(w_i; z_1, \ldots, z_n) = h_0(w_i)e^{(\beta_1 z_1 + \ldots + \beta_n z_n)}.....(1)$$

where β_j , j=1,...,n are the parameters of the model, z_j, j=1,...,n the values of the explanatory variables and $h_0(w_i)$ is the baseline hazard.

Now if $h_0(w_i) = ab$, a constant, i.e., the well known exponential distribution and

$$z_1 = log e\left(1 - \frac{(i-1)}{a}\right)$$

(with an appropriately chosen value for a); then a value of β_1 approximately equal to one obtained when PHM is applied indicates that a binomial type exponential model is appropriate for the data under investigation. The hypothesis that $\beta_1 = 1$ may be tested, as β_1 is asymptotically normal, (Tsiatis (1981), Anderson and Gill (1982)).

An alternative formulation is to use the approximation for small βz in equation (1) so that

$$e^{\beta_1 z_1} \approx 1 + \beta_1 z_1 = 1 - \frac{(i-1)}{a}$$

with $h_0(w_i) = ab$. Thus $z_1 = i - 1$ with $\beta_1 = -\frac{1}{a}$.

If the covariate value β_1 is estimated first within the

PHM formulation, then this provides the inverse of minus the estimate of the total number of faults at time zero.

5.3.2 THE WEIBULL MODEL

The intensity function of this model is given by $\lambda(t_i) = ae^{-bt_i^c} bct_i^{c-1}$ $a,b,c>0$ with a mean value function of $M(t_i) = a\left(1 - e^{-bt_i^c}\right)$. It is discussed in Musa and Okumoto (1984) and Miller (1986) among others. When c=1, the model becomes the Goel-Okumoto model (see below). It cannot be transformed into a proportional hazards model due to the power term of the time metric.

5.3.3 THE LITTLEWOOD NHPP

This is discussed in Rook (1990) and Littlewood (1981) with an intensity function of the form $\lambda(t_i) = abc(1+bt_i)^{-c-1}$ $a,b>0$. The mean value function is $M(t_i) = a(1-(1+bt_i)^{-c})$. This model may also be viewed as an order statistics process in the reference of Littlewood (1984). When the b parameter is large, the intensity function for this model reduces to the Duane model discussed later. If $a \to \infty$, $b \to 0$ and $ab = k$, a constant, then this model becomes the Goel-Okumoto/ Jelinski-Moranda NHPP model.

5.3.4 POISSON TYPE MODELS OF THE EXPONENTIAL CLASS

A proportional hazards model has been formulated which incorporates Poisson type models of the exponential class (Wightman, McCollin and Dixon 1991). In this formulation, the exponential distribution is again the assumed per fault distribution with the Poisson distribution

referring to the number of faults experienced by time t. Examples of this type of model are Musa (1975), Scneidewind (1975), Moranda (1975) and Goel-Okumoto (1979).

The formulation of the proportional hazards model for the Poisson type models of exponential type is as follows. From Musa et al (1987) page 276, the source hazard rate

$$h(w_i/t_{i-1}) = abe^{-bt_{i-1}}e^{-bw_i}$$

Let t_{i-1} equal the time of the (i-1)th failure with a-i+1 faults left. From Musa et al (1987), for this model the number of faults left at t_{i-1} is $ae^{-bt_{i-1}}$ so that

$$h(w_i/t_{i-1}) = (a-i+1)be^{-bw_i}$$

which may be written as

$$h(w_i/t_{i-1}) = ab\left(1-\frac{(i-1)}{a}\right)e^{-bw_i} \qquad ...(3)$$

In the PHM formulation (1), let

$h_0(w_i) = abe^{-bw_i}$, the Gumbel hazard of chapter 1.7.5 and $z_1 = loge\left(1-\frac{(i-1)}{a}\right)$.

When applying PHM, if an estimate of β_1 approximately equal to one with a form of the baseline hazard shown above, then Poisson type exponential models are appropriate for the data under investigation.

This extreme value baseline hazard function in this formulation is considered by Lloyd and Lipow (1977). The hypothesis that β_1 is approximately equal to one may be tested (in the same way as the binomial class of models)

as the β's from PHM are asymptotically normal. The form of the baseline hazard for this software reliability model type may be investigated by plotting the logarithm of the cumulative baseline hazard against time t.

Alternatively, by using the approximation

$$e^{\beta_1 z_1} = 1 + \beta_1 z_1$$

within the PHM formulation and equating it to $1 - \frac{(i-1)}{a}$ in equation (3) then $z_1 = i - 1$ and $-\beta_1^{-1}$ provides an estimate of the initial number of faults in the source.

If we choose to fit the Musa basic execution time model (Musa 1975) and the model of Goel-Okumoto (1979) into a PHM formulation, then the covariate structure is somewhat easier to calculate for data analysis. This Musa formulation appears in McCollin et al (1990).

5.3.5 THE MUSA BASIC EXECUTION TIME MODEL

Font (1985) derived a proportional hazards model with the Musa model as the hazard function. The following formulation of the Musa model within a PHM framework is useful as a goodness of fit test for the Musa model in that if the number of software failures is not a significant explanatory variable in the PHM formulation then the Musa model is not appropriate for the data analysis.

The Musa basic execution time model (Musa (1975)) takes the form

$$\lambda(t_i) = a e^{-bt_i}$$

where t_i is the total execution time, a is the initial failure intensity, $\lambda(t_i)$ is the failure intensity function and b denotes the "constant hazard which characterises any individual failure"

The expected number of failures in time t_i is given by

$M(t_i) = \int_0^{t_i} \lambda(w)dw$ which is

$\frac{a}{b}(1 - e^{-bt_i})$.

From Musa (1987), the cumulative hazard function is

$$H(w_i/t_i) = M(t_{i-1} + w_i) - M(t_{i-1})$$

Letting t_{i-1} = cumulative failure time up to time i-1 and hence w_i = time since last failure.

Thus, $H(w_i/t_i) = -a(e^{-bt_{i-1}+w_i} - e^{-bt_{i-1}})/b$.

If we differentiate $H(w_i/t_i)$ with respect to w_i,

$$(dH(w_i/t_i)/dw_i = a e^{-bw_i} e^{-bt_{i-1}} \quad , \quad -(4))$$

then $dH(w_i/t_{i-1})/dw_i = h(w_i)$
Now, the PHM formulation is

$$h(w_i, z) = h_0(w_i) e^{\beta_1 z_1} \quad , \quad -(1)$$

where w_i is the time since last failure,
z_1 is an explanatory variable,
β_1 is a parameter of the model and
$h_0(w_i)$ is the baseline hazard.

Now comparing (1) and (4),

i) If $h_0(w_i)$ from PHM = ae^{-bw_i} (a Gumbel hazard) and

ii) $e^{\beta_1 z_1} = e^{-bt_{i-1}}$:- $\beta_1 = -b$, $z_1 = t_{i-1}$
then the basic execution model is a sub model of PHM.

5.3.6 THE S-SHAPED INFLECTION MODEL (OR OHBA MODEL)

Ohba (1984) describes an inflection s-shaped software reliability growth model defined by

$$\lambda(t_i) = \frac{ab(c+1)e^{bt_i}}{(1+ce^{-bt_i})^2}$$

where $\lambda(0) = \frac{ab}{(c+1)}$, $\lambda(\infty) = 0$.

The parameters a b and c are, respectively, the initial error content, a>0, the error detection rate, 0<b<1, and the inflection parameter, $c = \frac{(1-r)}{r}$

where r is the ratio of the number of detectable faults to the total number of faults in the software. When r=1, the model becomes the Goel-Okumoto model described later. Kapur and Garg (1991) supply an optimal release policy for this model.

The mean value function is $M(t_i) = a\frac{(1-e^{-bt_i})}{(1+ce^{-bt_i})}$

which may be written as $M(t_i) = a\frac{(c+1)}{c}\left(\frac{1}{\left(1+ce^{-bt_i}\right)} - \frac{1}{(c+1)}\right)$.

We may rewrite the above equation as

$$\frac{1}{\left(1+ce^{-bt_i}\right)} = \left(\frac{1}{c+1}\right)\left(1 + \frac{cM(t_i)}{a}\right).$$

Now $e^{\frac{cM(t_i)}{a}} \approx 1 + \frac{cM(t_i)}{a}$,

We may use the actual cumulative number of failures by time t_i as a covariate in a PHM formulation which will result in an "Ohba" type proportional hazards model. We may write the intensity function as

$$\lambda(t_i) = \left(\frac{ab}{c+1}\right)e^{-bw_i}e^{-bt_{i-1}}e^{2\frac{ci}{a}}.$$

Now choose

$h_0(w_i) = \frac{ab}{c+1}e^{-bw_i}$, (a Gumbel hazard) and $z_1 = t_{i-1}$

so that $\beta_1 = -b$ and $z_2 = i$ so that $\beta_2 = 2\frac{c}{a}$. Thus the "Ohba" type model is a submodel of PHM. As both of these covariates are increasing, there may be problems of monotonicity and collinearity if these two covariates are fitted together in the same PHM formulation. If fitting is possible, the estimate of β's may be determined first and then by plugging in this estimate, appropriate values of a and c may be found.

5.3.7 THE GOEL-OKUMOTO MODEL

As has been mentioned, the Jelinski-Moranda model (1972) has a similar physical interpretation to the Goel-Okumoto (1979) model. The Jelinski-Moranda (J-M) model has been discussed prior to this software application by Bazovsky (1961), Cozzolino (1968) and is similar to the Cox-Lewis model (1966) described later. Bayesian formulations of J-M have been carried out by Raftery (1988), Meinhold and Singpurwalla (1983) and Littlewood and Sofer (1987).

The intensity function of the Goel-Okumoto model is

$$\lambda(t_i) = abe^{-bt_i} \qquad a, b > 0$$

which is similar in form to the Cox-Lewis model.

The expected number of failures is $M(t_i) = a(1 - e^{-bt_i})$.

By writing the intensity function as $\lambda(t_i) = abe^{-bw_i}e^{-bt_{i-1}}$ then we may choose $h_0(w_i) = abe^{-bw_i}$ (a Gumbel hazard) and $z_i = t_{i-1}$ and thus the Goel-Okumoto model is a sub model of PHM.

5.3.8 THE S-SHAPED MODEL

This model is described by Yamada, Ohba and Osaki (1983) and has a S-shaped mean value function given by

$M(t_i) = a\left(1 - (1 + bt_i)e^{-bt_i}\right)$ with an intensity function of the form

$$\lambda(t_i) = ab^2 t_i e^{-bt_i} \qquad a, b > 0.$$

The maximum value of the intensity function may be

calculated by equating it's differential to zero which is abe^{-1} when time $t=\frac{1}{b}$.

The intensity function may be written as

$$\lambda(t_i)=ab^2e^{\ln t_i}e^{-bt_{i-1}-bw_i}$$

which becomes

$$\lambda(t_i)=ab^2e^{-bw_i}e^{\ln t_i}e^{-bt_{i-1}}.$$

This cannot be written as a PHM formulation as the two covariates $z_1=\ln t_i$ and $z_2=t_{i-1}$ are monotonic and collinear.

5.4 DESCRIPTION OF MODELS FOR HARDWARE AND SOFTWARE RELIABILITY GROWTH

5.4.1 THE DUANE MODEL

In 1962, J.T. Duane of General Electric published a report (Duane (1964)) in which he presented a plot on log-log paper of the observed cumulative failure rate against cumulative time for a number of complex systems. These plots all closely followed a straight line. Many papers have been written since on this model notably Crow (1974), who showed that the Duane model is a NHPP. The model has also been called the AMSAA model (MIL-HDBK-189) and more recently the Weibull Process and the Power Law Model and the theory is now the most developed of all NHPP's.

The expected number of failures in the period $(0,t_i)$ is given by $E(N(t_i))=M(t_i)=\left(\frac{t_i}{a}\right)^b$.

The intensity function is given by

$$\lambda(t_i) = b\frac{M(t_i)}{t_i} = b t_i^{b-1} a^{-b} \qquad a, b > 0.$$

When the parameter b<1, then the system is undergoing reliability growth and when b=1 the Homogeneous Poisson Process is formed where each interfailure time is exponentially distributed. Only the time to first failure is Weibull distributed when b is not equal to one. When b>1, the system under analysis is undergoing a period of reliability decay where the interfailure times are becoming shorter as cumulative time increases.

For the Duane model, by using $M(t_i) = \left(\frac{t_i}{a}\right)^b$ then $t_i = a M(t_i)^{\frac{1}{b}}$

and on substitution into the intensity function, $\lambda(t_i) = b\, e^{\ln(M(t_i))} e^{-\ln t_i}$ which becomes $\left(\frac{b}{a}\right) e^{\left(1-\frac{1}{b}\right)\ln(M(t_i))}$.

Now, by using the log of the actual cumulative number of failures up to time t_i instead of $\ln(M(t_i))$ as a covariate z_1 and $h_0(w_i) = \frac{b}{a}$, then a similar model to the Duane model is formulated and this "Duane" type model is a sub model of PHM.

Smith and Oren (1980) have described a modified version of the Duane model for large counts of failures.

5.4.2 THE COX-LEWIS MODEL

The Cox-Lewis Model or log-linear model was first derived in Cox-Lewis (1966) and subsequently derived from a number of assumptions in Cozzolino (1968).

The mean value function of this NHPP is $M(t_i)=\left(\frac{e^a}{b}\right)(e^{bt_i}-1)$ and the intensity function is $\lambda(t_i)=e^{a+bt_i}$ $a,b>0$ where t_i is the cumulative time to fault i in a source.

Using Musa (1987), the model may be derived as a sub model of PHM as follows.

The intensity function for the cumulative time to failure may be written as $\lambda(t_{i-1}+w_i)=e^{a+bt_{i-1}+bw_i}$.

By letting $z_i=t_{i-1}$ and $h_0(w_i)=e^{a+bw_i}$, a Gumbel hazard rate within the PHM formulation of (1), then the Cox-Lewis model is a sub model of PHM.

5.4.3 THE IBM MODEL

The IBM model was first derived by Rosner (1961) and subsequently has been formulated by Ascher (1968) and also called the Smith's Industries Model (1977). The mean value function is

$$M(t_i)=ct_i+\left(\frac{a}{\ln b}\right)(b^{t_i}-1)$$

where 0<b<1, a<0 for increasing hazard rate and c+a>=0 and a>0 for decreasing hazard rate. Unlike the Power Law model and the log-linear model, the intensity function tends to a constant over time and not infinity. The intensity function is $\lambda(t_i)=c+ab^{t_i}$.

There are problems with estimating the parameters of this model. The techniques used for parameter estimation estimation are by non-linear least squares or maximum likelihood. By choosing the incorrect initial estimates

for the parameters in the iterative estimation procedure, the parameter estimates will not tend towards the required values as the likelihood function has inflexion points as well as the optimal solution. The solution should be checked by ensuring that all the second differentials of the log likelihood for the three parameter estimates satisfy the conditions for a maximum.

5.4.4 THE BOUNDED INTENSITY MODEL

The bounded intensity function (Hartler (1989)) is given by

$\lambda(t_i) = b(1-(1+bt_i)^{-1}) \qquad b>0$

which has the mean value function

$M(t_i) = bt_i - \ln(1+bt_i).$

This is similar to the IBM model above in that the intensity function tends towards a finite value. It is not possible to incorporate the IBM model or the bounded intensity model into a PHM formulation, however in chapter 7, the IBM model is used as a baseline intensity in a data set analysis.

5.4.5 THE LOGARITHMIC MODEL

This model was first put forward by Musa and Okumoto (1984) as part of the Logarithmic Poisson execution model. The intensity function is

$\lambda(t_i) = a(1+abt_i)^{-1} \qquad a,b>0$

and it's mean value function is $M(t_i) = \left(\frac{1}{b}\right)\ln(1+abt_i).$

Substituting $M(t_i)$ into the intensity function gives $\lambda(t_i) = ae^{-bM(t_i)}$.

By choosing $h_0(w_i) = a$ and $z_i = i - 1$, then a "logarithmic" type model is a sub model of PHM.

5.4.6 THE SQUARE ROOT MODEL

This model is discussed by Kremer (1983). The intensity function is $\lambda(t_i) = \frac{a}{\sqrt{t_i}}$ and the mean value function is $M(t_i) = 2a\sqrt{t_i}$. The model is also the Duane model with shape parameter equal to one half.

By applying the formula for the mean value function into the intensity, and using the Musa relationship between hazard and intensity, the hazard function for a "square root" type model may be written as the proportional hazards formulation using the actual number of failures up to time t_i not the expected number $M(t_i)$:

$\lambda(t_i) = \frac{2a^2}{i}$.

On choosing $z_i = log e(i)$, the corresponding β should be tested for a value of -1. The baseline hazard is $h_0(w_i) = 2a^2$ which is an exponential hazard.

5.5 APPLICATION OF PHM FORMULATIONS TO SOFTWARE RELIABILITY DATA

Most of the reliability growth models described above are given in the list below and may be formulated as a proportional hazards models with a baseline hazard which is either extreme value or exponentially distributed. All except one model have one covariate in their respective PHM formulations. For the "Ohba" type model, the effect

of the model S shape may be due to external influences such as reduced severity of testing or change in staff levels. If such information is available, it may be more useful to apply as a covariate since the covariate would then have a physical meaning and the complex S model may reduce to a simpler form. The PHM formulation for this S-shaped model is very susceptible to model fitting as both the covariates are monotonic in the model and so they are likely to be linearly related to one another which will affect convergence of the parameter estimation.

Five models which do not fit into the framework are the Littlewood, Weibull, IBM, S-shaped and bounded intensity models. Apart from the bounded intensity model and the S-shaped model, these are all three parameter models. It may be shown that the maximum likelihood estimate of a (the number of failures parameter) is a linear combination of the other two parameters and due to this additivity, PHM formulations are not possible. Also, in all four of these models, the intensity function is a sum of terms rather than a product and therefore the model structure cannot be reduced to a hazard term multiplied by a covariate. The S-shaped model cannot be formulated as a proportional hazards model as it requires two covariates which are collinear.

The other four models which could not be formulated into a PHM framework were the Duane, Ohba, square root and logarithmic models however similar models to these have been formulated as proportional hazards models.

The baseline hazard for each of these formulations is either the exponential or the Gumbel and the covariate term is either a function of the accrued failure time or

number of failures.

The PHM approach has the advantage over the physical structure approach in that it may be used as a diagnostic tool to determine which is the most appropriate NHPP to model reliability growth. The approach is described in chapter 6 and is applied to part of Alvey data set number 3.

TABLE 5.2. TABLE OF PHM FORMULATIONS

Model	Covariate z_i	Coeff. β	Hazard
Binomial type (1)	$loge(1-(i-1)/a)$	1	ab
Binomial type (2)	$i-1$	$-1/a$	ab
Poisson type (1)	$loge(1-(i-1)/a)$	1	abe^{-bw_i}
Poisson type (2)	$i-1$	$-1/a$	abe^{-bw_i}
Musa Basic	t_{i-1}	$-b$	ae^{-bw_i}
"Ohba" type	$z_1=t_{i-1}$ $z_2=i-1$	$\beta_1=-b$ $\beta_2=2c/a$	$abe^{2\frac{c}{a}}e^{-bw_i}/(c+1)$
Goel- Okumoto	t_{i-1}	$-b$	abe^{-bw_i}
Cox-Lewis	t_{i-1}	b	e^{a+bw_i}
"Duane" type	$loge(i)$	$1-1/b$	b/a
"Square Root" type	$loge(i)$	-1	$2a^2$
"Log" type	$i-1$	$-b$	ae^{-b}

6 PROPORTIONAL HAZARDS MODELLING

Cox (1972) presented proportional hazards modelling in a seminal paper and suggested it would be useful in reliability studies as well as other activities such as medicine and actuarial studies. The formulation takes the form

$$h(t; z_1, z_2, \ldots, z_k) = h_0(t) e^{(\beta_1 z_1 + \beta_2 z_2 + \cdots + \beta_k z_k)} \quad , \quad t > 0, -\infty < \beta_i, \quad z_i < \infty \qquad -(1)$$

where the z_i's are the values of the covariates and the β_i's are the unknown parameters of the model representing the effect on the overall hazard of each of the values of the covariates. A list of covariates or explanatory factors applicable to software reliability are given in McCollin, Bendell and Wightman (1989). Software metrics may also be used as covariates. The values of the β_i's are unknown and they represent the effect of the z_i's on the hazard. The $h_0(t)$ term is a baseline hazard function. The only assumption of the model is that the hazards are common, stable and proportional which may be checked by plotting the respective hazard functions for different covariate values with each other against time.

The following approach is used to analyse waiting times to failure using PHM and provides an objective selection criteria for the application of the appropriate software reliability growth models to a given data set. This contrasts with the previous approaches where arbitrary fitting of a number of models were compared.

1. Initially, choose covariates. The covariates selected for the waiting times to failure will be cumulative time since last failure, number of failures up to last failure

and log number of failures. These are the covariates which were shown to be part of the formulation of the well known NHPP's presented in chapter 5.

2. Calculate the β_i's and test them being zero. The available software (see chapter 6.1) attempts to fit all the covariates and successively removes each non-significant covariate by using the assumed normality of the parameter estimates. The p - values of individual covariate values (the probability of observing a value more extreme than the test statistic value) determined by assuming a univariate normal distribution are an approximation to the actual multivariate normal p - values of the covariate values.

PHM software may encounter two problems with data. These are multicollinearity or partial collinearity where the covariates are a linear combination of each other. Collinearity between covariates is discussed in chapter 3.5 in which there is reported a very strong linear relationship between number of failures and cumulative time since last failure for the twelve least reliable sources in Alvey data set number 3.
The second problem with data is that the values of the time metric have the same rank as the corresponding covariate values (monotonicity). PHM may be used as a diagnostic tool to determine if either of these problems exist however correlation and regression are well documented tools for this. Covariates found to be monotonic or collinear by proportional hazards modelling were analysed in more detail by multivariate techniques and this is presented in chapter 7.

Once the β_i's have been calculated, the software calculates an estimate of the baseline cumulative hazard (equation 7 in chapter 6.2).

3. For the particular covariate, fit the appropriate cumulative hazard formulations to determine which of the NHPP's in table 5.2 are applicable to the data set.

4. Apply diagnostic checks for goodness of fit of the data to the models.

This 4 step procedure has been carried out by myself for the twelve least reliable sources of Alvey data set number 3 and the results are presented in chapters 6.3 and 6.4.

A proportional hazards model has been derived in chapter 6.1 for comparison with the time series analysis presented in chapter 4. The covariates chosen are those which correspond to trend, moving average and seasonality in a time series context.

6.1 ANALYSIS OF FAILURE COUNTS

An analysis using PHM was applied to the number of failures per day data of Alvey data set number 3 to determine the effect of the seasonality and trend and to compare with the results of the Box-Jenkins time series analysis.

As this data is a counting process, the appropriate model is the multiplicative intensity model of Fleming and Harrington (1991). This may be expressed as the proportional hazards model of Cox (1972) however the function

h(t;z) inherits none of the usual properties of the traditional hazard function and so cannot be estimated from the data.

An alternative approach is to use the discrete model of Kalbfleisch and Prentice (1980). The model is

$$h(t;z)dt = 1-(1-h_d(t)dt)^{\exp(\beta_1 z_1+\ldots+\beta_k z_k)}$$

where $h_d(t)dt = \sum_{i=1}^{n} h_i \delta(t-x_i)dt$ and $\delta(x)=1, \quad x=0$; otherwise.

There are problems with the estimation procedure for discrete PHM so Kalbfleisch and Prentice (1980) page 101 suggest using the continuous model as an alternative since the relative risk parameter $\exp(\underline{\beta}\underline{z})$ is exactly the same as in the continuous model.

An alternative formulation is as follows. If the data was assumed to be continuous with each time to failure less than a day (which for software execution times to failure is nearly always true) then Kalbfleisch and Prentice (1980) show that by grouping continuous data into disjoint intervals $[0=a_0,a_1),[a_1,a_2),\ldots,[a_{k-1},a_k=\infty)$, the hazard of failure in the i'th interval for an individual with covariate z is

$$P\{T\in[a_{i-1}a_i)/T\geq a_{i-1}\} = 1-(1-h_i)^{\exp(z\beta)}$$

where $h_i = \exp\left(-\int_{a_{i-1}}^{a_i} h_0(u)du\right)$.

In the case of Alvey data set 3, each time to failure is unity with different frequencies so it is not possible to estimate the model parameters.

The approach using the continuous model is adopted and the analysis will be compared with the previous time series analysis of counts.

Software has been written by Dr. D. Wightman at Nottingham adapted from the routines in Kalbfleisch and Prentice (1980) and described in his Ph.D thesis (1987) to estimate the β_i's. The method is described in chapter 6.2. The covariates used in the PHM analysis of the failure count data were chosen to be similar to those in a time series context. These were day of the week, previous counts per day over the previous six days and the cumulative number of failures. The results are as follows.

TABLE 6.1. PHM RESULTS FOR FAILURE COUNTS

Covariate	Value	Normal Deviate	p value (1-sided)
Failure Count z1	0.002619	4.8268	0.0000
Previous Day No. of Fail-ures z2	-0.046036	-2.2649	0.0118
Sunday z3	0.919899	4.262	0.0000
Saturday z4	0.740664	3.6110	0.0002

The analysis shows that the count hazard rate is increasing on weekends compared to the rest of the week. Also, the hazard rate is increasing as failures accumulate and decreasing with the number of failures on the previous day.
These covariates produce in the proportional hazards model a similar structural model to the
ARIMA(1 0 1)(0 1 1)7 model where the time series moving average term corresponds with the PHM covariate cumulative time to failure (age) and the time series trend corresponds with the PHM covariate previous count. These similarities are shown below.

The PH model formulation

$$h(t;z) = h_0(t)e^{(\beta_1 z_1+\beta_2 z_2+\beta_3 z_3+\beta_4 z_4)}$$

may be integrated with respect to t to give
$H(t;z) = H_0(t)e^{\beta_1 z_1+\beta_2 z_2+\beta_3 z_3+\beta_4 z_4}$ where H is the cumulative hazard function.

On taking logs, the formulation becomes
$\log H(t;z) = \log H_0(t) + \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3 + \beta_4 z_4$ which is similar to the formulation for the logged time series data in chapter 4 :

$$z_t = C + z_{t-1} + z_{t-7} - z_{t-8} - \theta a_{t-1} - \Theta a_{t-7} + \theta\Theta a_{t-8} + a_t$$

An exact statistical relationship between the two formulations does not exist however Whitehead (1980) develops a Poisson model using GLIM which is equivalent to the PHM formulation and a time series proportional hazards structure has been discussed in chapter 4.

6.2 ANALYSIS OF FAILURE TIMES USING PHM

Proportional hazards modelling has been carried out on the twelve sources which were repaired the greatest number of times given in tables 3.2 and 3.3. These twelve least reliable sources were analysed together and then individually. The plot of this data is in figure 3.4.

The continuous model is given by

$$h(t;z) = h_0(t)e^{\sum_{i=1}^{k}\beta_i z_i}$$

Prentice, Williams and Peterson (1981) considered the following formulations of PHM :

$$h(t/N(t),\underline{z}(t)) = h_{0S}(t)e^{\beta_S \underline{z}(t)} \quad -(1)$$

$$h(t/N(t),\underline{z}(t)) = h_{0S}(t)e^{\beta \underline{z}(t)} \quad -(2)$$

$$h(t/N(t),\underline{z}(t)) = h_{0S}(t-t_{N(t)})e^{\beta_S \underline{z}(t)} \quad -(3)$$

$$h(t/N(t),\underline{z}(t)) = h_{0S}(t-t_{N(t)})e^{\beta \underline{z}(t)} \quad -(4)$$

where N(t) is the number of failures on an item prior to time t,

s is a stratification variable which varies over a number of strata,

t is the time from the start of the study period and

$t-t_{N(t)}$ is the time since the last observed failure of the item.

The likelihood function for the Cox continuous proportional hazards model is

$$L(\underline{\beta};h_0(t_i)) = \prod_{i=1}^{n} R_0(t_i)^{\exp(\beta z_i)} \prod_{i\epsilon D(t_i)} h_0(t_i) e^{\beta z_i}$$

which may be rewritten as

$$L(\underline{\beta};h_0(t_i)) = \{ \prod_{i\epsilon D(t_i)} \frac{e^{\underline{\beta} z_i}}{\sum_{z_j \epsilon G(t_i)} e^{\underline{\beta} z_j}} \} \prod_{i\epsilon D(t_i)} h_0(t_i) \sum_{z_j \epsilon G(t_i)} e^{\beta z_i} \prod_{i=1}^{n} R_0(t_i)^{\exp(\beta z_i)}$$

where $D(t_i)$ is the set of items which have failed at t_i,

$G(t_i)$ is the set of items still at risk of failure just before time t_i,

$R_0(t_i)$ is the items associated baseline survivor function,

n is the total sample size

β is a row vector of k parameters

z_i is a column vector of k measured covariate values.

Cox (1972 and 1975) showed that the baseline hazard function may be left distribution free within the above likelihood construction and developed a term

$$\{ \prod_{i\epsilon D(t_i)} \frac{e^{\underline{\beta} z_i}}{\sum_{z_j \epsilon G(t_i)} e^{\underline{\beta} z_j}} \}$$

known as a 'conditional likelihood' or Cox's partial likelihood. Cox showed that the $\underline{\beta}$ parameter estimates may be determined by maximum likelihood estimation of this partial likelihood as it contains all the information of the risk set for estimating the covariate values. A method of estimation employs the Newton-Raphson iterative pro-

cedure on the first differentials of the log likelihood. The problem of ties (equal life lengths) is dealt with in Breslow (1974).

The Kalbfleisch and Prentice (1980) software routines estimate the baseline hazard by expressing the likelihood as $L(\alpha,\beta)=\prod_{i=1}^{n}\left\{\prod_{z_j\in D(t_i)}\left(1-\alpha_i^{\exp(\beta z_j)}\right)\prod_{z_h\in(G(t_i)-D(t_i))}\alpha_i^{\exp(\beta z_h)}\right\}$ -(5)

where $(1-\alpha_i)$ is the hazard contribution at t_i. Since the covariate values have already been estimated from the partial likelihood, then (5) may be maximised to estimate the α_i's.

The baseline survivor function is estimated by

$\hat{R}_0(t)=\prod_{i/t_{(i)}<t}\hat{\alpha}_i$ where

$$\hat{\alpha}_i=\left(1-\frac{\exp(z_{(i)}\beta)}{\sum_{h\in G(t_i)}\exp(z_h\beta)}\right)^{\exp(-z_{(i)}\beta)}$$

which provides an estimate of the baseline cumulative hazard of

$\hat{H}_0(t)=-\sum_{i/t_{(i)}<t}\ln\hat{\alpha}_i$ -(6).

Lawless (1982) derives the first order approximation of (6) as

$\hat{H}_0(t)=\sum_{i/t_{(i)}<t}(1-\hat{\alpha}_i)$ -(7)

and this is used in the routines of Kalbfleisch and Prentice (1980) and Wightman (1987). The computer routines for fitting PHM written by Dr Wightman successively removes

each insignificant covariate one at a time in the model until all the remaining covariates are significant. A number of diagnostic plots are available in the literature to determine goodness of fit and outliers (Schoenfeld (1982), Cox and Snell (1968), Cain and Lange (1984), Reid and Crepeau (1985)).

Cain and Lange (1984) and Reid and Crepeau (1985) show that the influence of an event at time t_j upon the estimate of the β value may be calculated by taking the first order approximation based on a Taylor series expansion of the difference between the estimate of the β value with all the observations included and the estimate of the β value with the observation at t_j omitted. This is then transformed into a normal deviate and compared with ± 1.96 to determine if the event alters the significance of the covariate if it is omitted.

Cox and Snell (1968) obtained residual quantities which should be roughly exponentially distributed if the proportional hazards model is a good fit. Plotting a product limit survivor function estimated from the set of residuals against the residual estimates produces a graphical goodness of fit test for the model since the plot should result in a straight line with gradient 1.

Examples of these plots and others are presented in Wightman and Bendell (1986) and McCollin, Wightman and Bendell (1989) among others. The latter reference shows some of the diagnostic plots of the analyses presented in chapter 6.4. Diagnostic plots are not presented in this thesis as only one or two of a number of covariates have been applied in the PHM specifications to show how the specific software reliability growth models fit in. Any diagnostic plots

would therefore be misleading as, for example, they may show covariates are missing when, in fact, they have been deliberately left out.

Problems with monotonicity and multicollinearity highlighted by PHM are dealt with by applying multivariate techniques in chapter 7.

6.3 ANALYSIS OF THE TWELVE LEAST RELIABLE SOURCES AS A GROUP

If all the twelve least reliable source failure times in Alvey data set number 3 are used with a covariate which designates each source number, then a comparison may be made of the reliability of these sources. This is model (3) in chapter 6.2 above. Steps 1 and 2 of the 4 step procedure is to select appropriate covariates and find the β values. The between sources variation was analysed by combining all the data of the individual sources and using covariates: source size or type and source designation (a binary covariate), age, number of failures, log number of failures and type of use. Information pertaining to 'source language' could not be incorporated as a covariate in the proportional hazards analysis as the twelve sources analysed were all in Cobol. The time metric used was time since last failure. Four formulations were found to be significant and they are listed in the table below.

TABLE 6.2. PHM RESULTS FOR TIMES SINCE LAST FAILURE

Model	Covariates	β Value	Significance, (Likelihood Ratio)	Baseline Intensity
1	Cumulative number of failures	-0.02658	0.0006 (12.563)	Binomial and Poisson models, "Log"
2	Cumulative time to failure	-0.0110	0.0000 (40.977)	Goel, Musa models
3	Log number of failures	-0.3412	0.00016 (12.798)	"Square root", "Duane"
4	Source no.10	-0.5403	0.0367 (6.784)	
	Source no.11	-0.5919	0.0383 (6.784)	

where the baseline intensities denote which of the NHPP's in table 5.2 are the most appropriate to the data based on the covariates fitted. On fitting the hazard function detailed in table 5.2, the appropriate NHPP's in table 5.2 will be identified.

The analyses showed that the hazard rate of all the sources decreased as age increased or as number of failures (or it's log) increased. The hazard rate of the sources denoted 10 and 11 were significantly less than the other sources. This is confirmed in the proportional intensity analysis of this data in chapter 7.6.4.

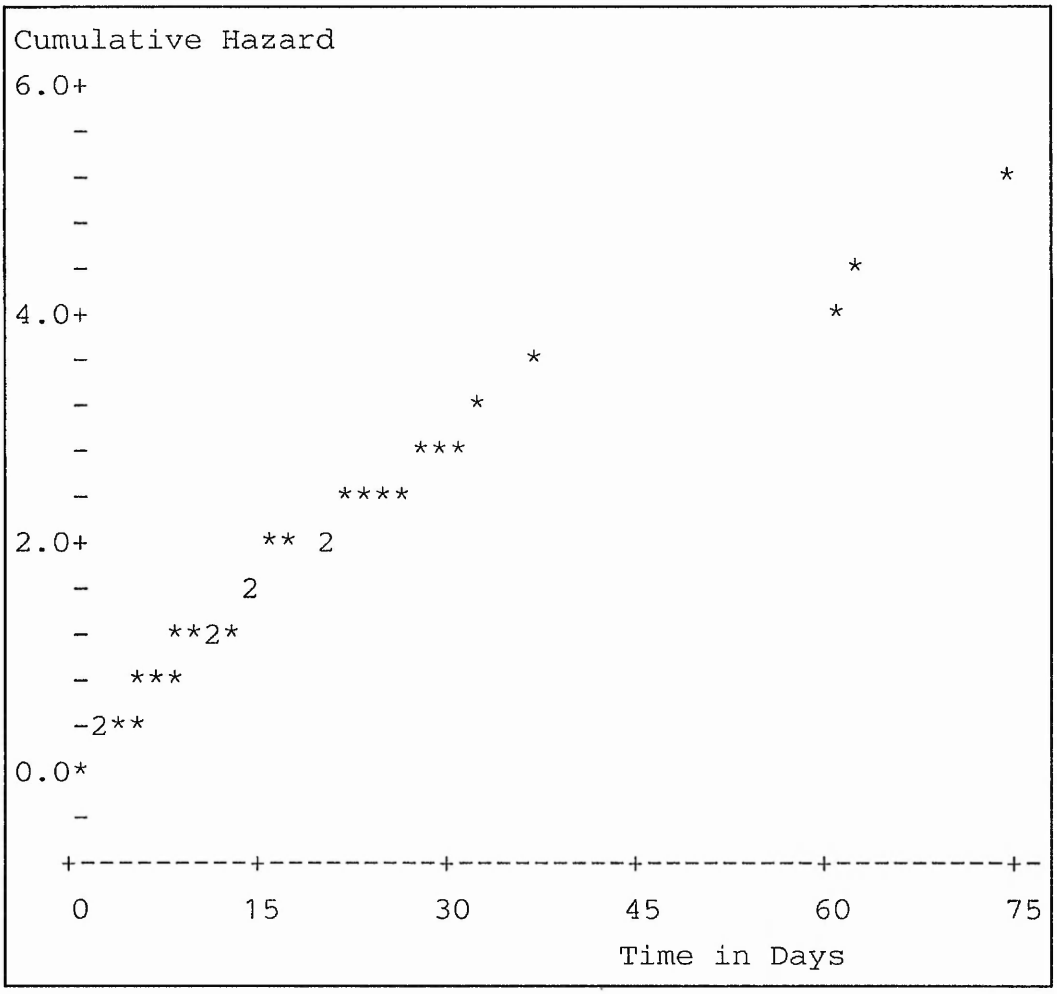
For a specific covariate structure, e.g. cumulative time to last failure and the appropriate cumulative hazard rate, e.g. the Gumbel hazard, will result in a listed NHPP in table 5.2, e.g. the Cox-Lewis model. Step 3 of the 4 step procedure is to model the appropriate hazards. Hazard analyses for these four proportional hazards structures were undertaken within the MINITAB package after the baseline cumulative hazard given in equation (7) in chapter 6.2 was estimated.

6.3.1 HAZARD ANALYSIS OF FORMULATION (1)

For formulation (1), possible NHPP's which may fit the data are the binomial and Poisson type exponential models and the "logarithmic" type model. The appropriate hazard functions to fit to the baseline in these cases are the Gumbel distribution for the Poisson and the exponential distribution for the binomial and "logarithmic" type models.

The cumulative hazard functions for these models are $H(w_i) = a(1 - e^{-bw_i})$, $H(w_i) = abw_i$ and $H(w_i) = ae^{-b}w_i$ respectively. A plot of the cumulative baseline hazard against time for this formulation is shown below.

FIGURE 6.1. PLOT OF ESTIMATED CUMULATIVE HAZARD AGAINST TIME SINCE LAST FAILURE



The binomial and Poisson models may be appropriate for this data as the $-\beta^{-1}$ value calculated from the PHM formulation represents the initial number of failures in the software for this particular model and is 37.62 which is extremely pessimistic. By plotting time since last failure w_i against $loge\left(1-\frac{H(w_i)}{a}\right)$, the slope parameter $-\frac{1}{b}$ of the Gumbel model may be estimated as long as the intercept

is zero. However this is not the case. If the best least squares model is fitted without the intercept, then b=0.00246

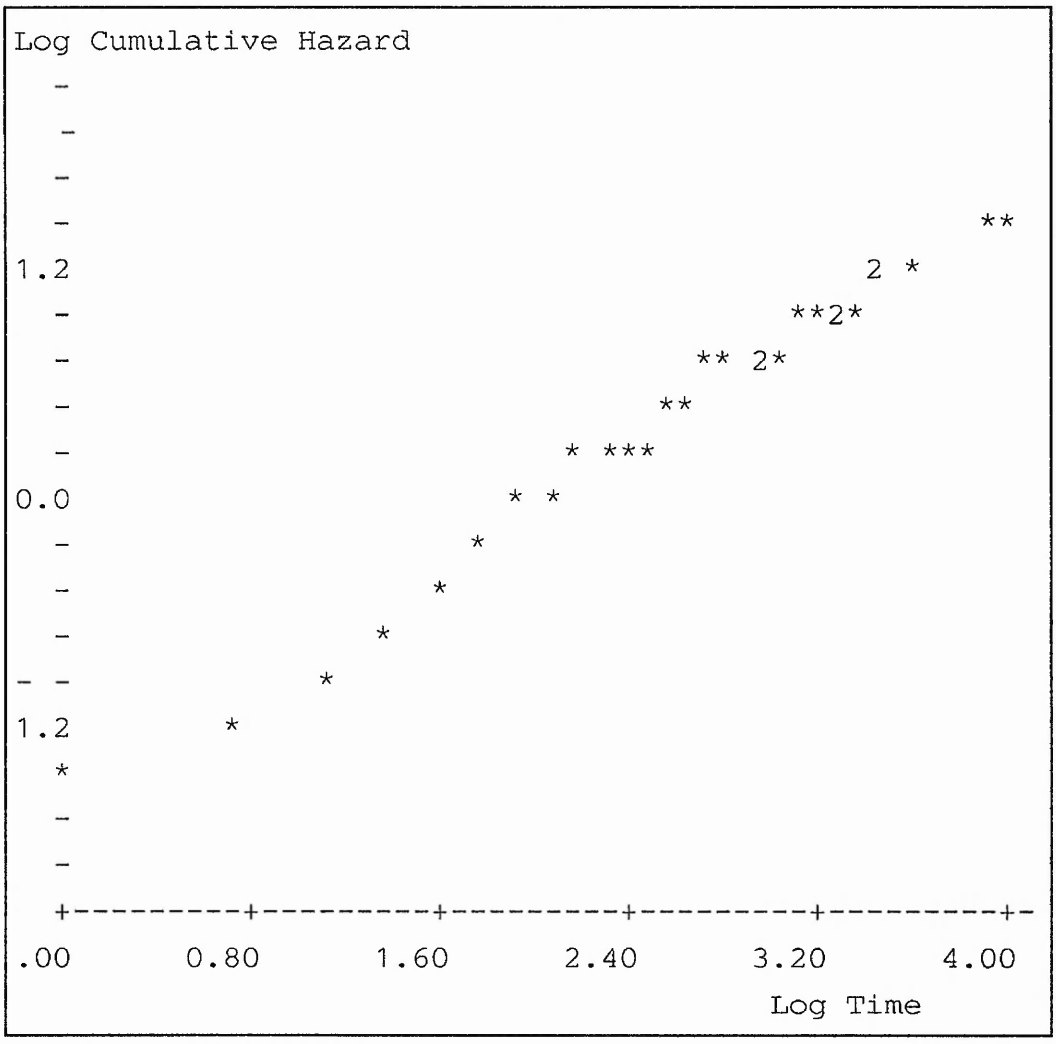
For the "logarithmic" type and binomial type models, the appropriate plotted relationship of cumulative hazard against time for an exponential distribution is a straight line through the origin. The coefficient of determination is 0.942 for this data so the data is almost linear. By regressing cumulative hazard on time since last failure, the model is $H(w_i) = 0.52059 + 0.070485 w_i$. The estimates of the parameters follow a normal distribution so given that the standard deviations of the estimates are 0.0862 and 0.003241 respectively, the estimate of the intercept is more than three standard deviations away from zero. This means that the exponential distribution is not appropriate.

There are four unusual observations, numbers 28R, 29RX, 30X and 31X where R denotes an observation with a large standardised residual and X denotes an observation whose X value gives it a large influence. Each of these observations occur in the live phase and cannot be resolved into a proportional hazards structure with the covariate 'type of use' as the observations are collinear with the covariate cumulative number of failures.

As the proposed hazard rate distributions do not fit the data, then either the three unusual observations may be removed from the linear plot and the models refitted or alternatively, hazard models such as the linear model (Gross and Clark (1975), the quadratic model of Gaver and Acar (1979) or the Weibull distribution may be fitted to

the data. By taking logs of the estimated cumulative hazard and the time since last failure, the plot shows that the Weibull distribution is appropriate.

FIGURE 6.2. PLOT OF ESTIMATED LOG CUMULATIVE HAZARD AGAINST LOG TIME SINCE LAST FAILURE



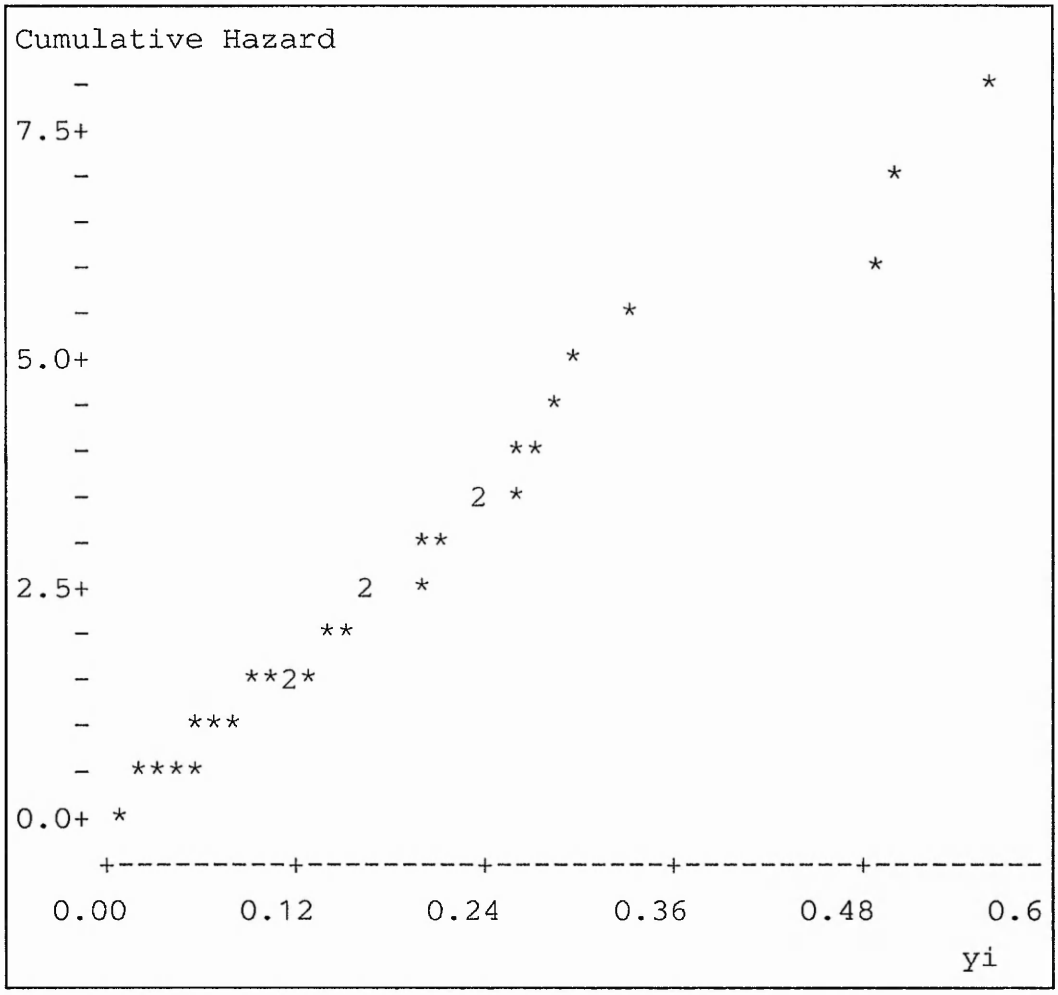
The coefficient of determination is 0.989 which denotes a strong linear relationship. The slope of the regression line is 0.7783 with a standard deviation of 0.01543 so that the parameter estimate is not within three standard

deviations of unity, i.e. the exponential distribution. The intercept of the regression line is -1.6047 with a standard deviation of 0.04328 with three unusual observations; 2X, 4R and 29R. Thus the Weibull cumulative hazard function is given by $H(w_i) = 0.2009 w_i^{0.7783}$.

6.3.2 HAZARD ANALYSIS OF FORMULATION (2)

The possible NHPP's which may fit the data for formulation (2) are the Goel-Okumoto and Musa models. The estimate of -b within the cumulative hazard $H(w_i) = a(1 - e^{-bw_i})$ produces the following plot of $H(w_i)$ against $y_i = 1 - e^{-bw_i}$.

FIGURE 6.3. PLOT OF CUMULATIVE HAZARD AGAINST Yi FOR FORMULATION (2)

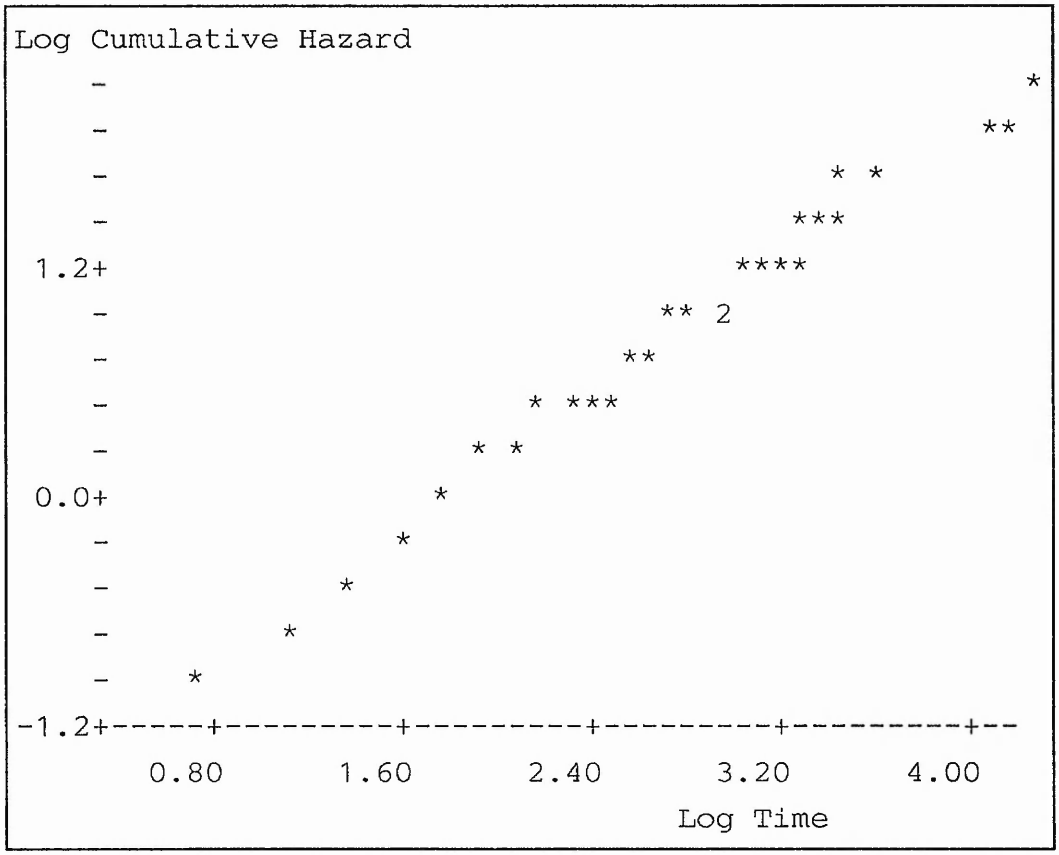


On regressing cumulative hazard on y_i, the intercept should be zero for the Goel-Okumoto and Musa NHPP's to fit the

data and it equals 0.137±0.158 assuming normality of errors so these two models are applicable.

The regression equation is $H(w_i) = 14.6y_i$ with the five unusual observations, 27R, 28R, 29RX, 30X and 31X and a coefficient of determination of 0.983. On taking logs of the original values, the Weibull distribution may be applicable. The logged data is shown in figure 6.4.

FIGURE 6.4. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR FORMULATION (2)

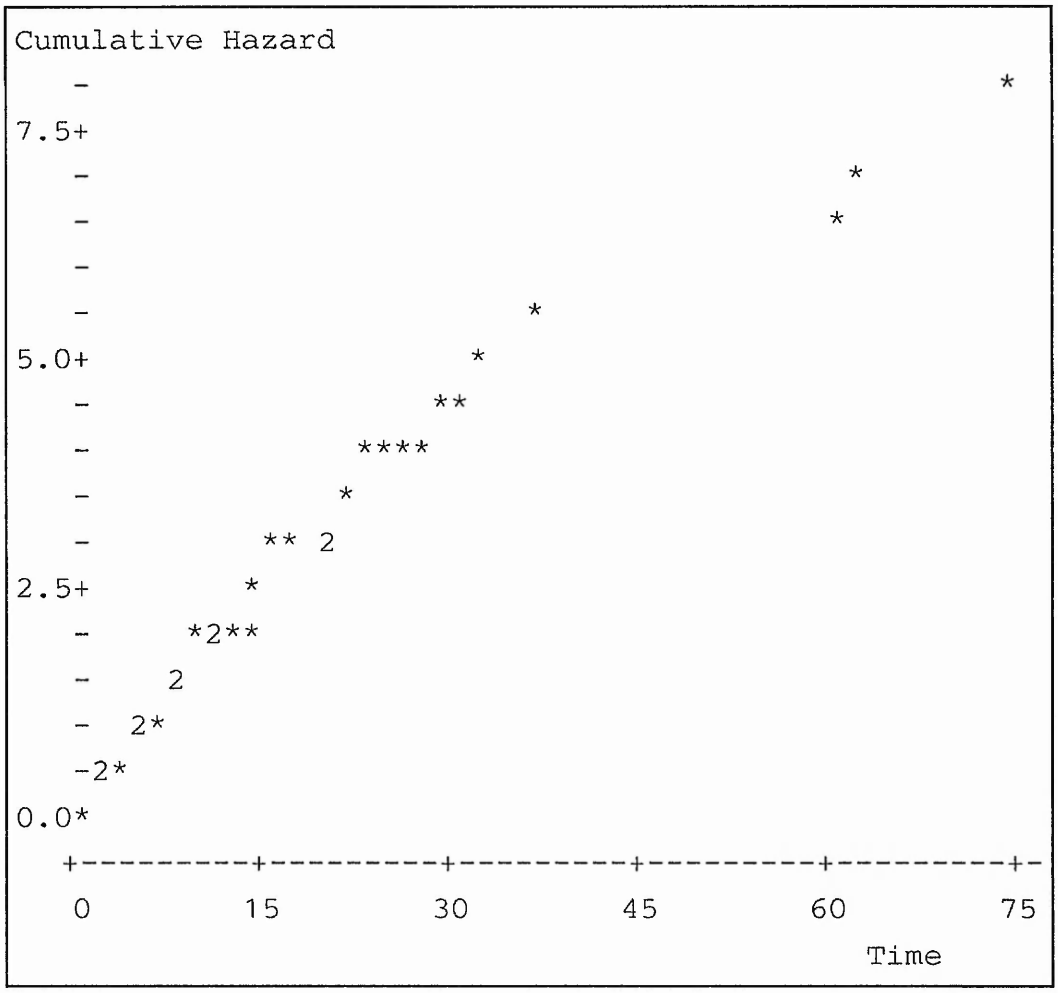


The regression equation is $\log(H(w_i)) = -1.488 + 0.847\log(w_i)$ (not an exponential distribution); the coefficient of determination is 0.988 and there is one missing zero time value and one unusual observation, 2RX.

6.3.3 HAZARD ANALYSIS OF FORMULATION (3)

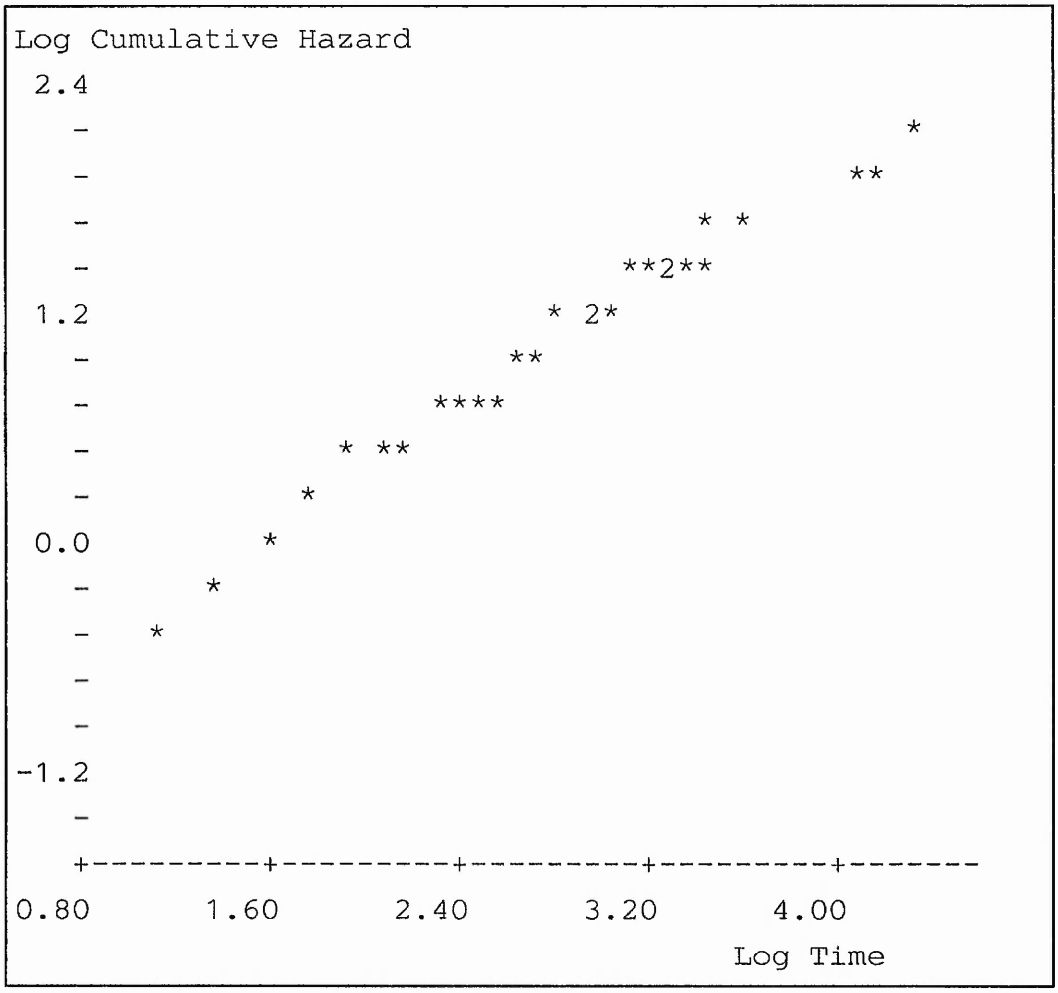
Formulation (3) suggests that the "square root" type model or the "Duane" type model may be valid for the data. The covariate value is -0.3412 and should be unity for the "square root" type model. This indicates the "Duane" type model as the only possible PHM formulation as long as the distribution of the hazard is exponential. A plot of the baseline cumulative hazard against the waiting time indicates a linear plot apart from the three outliers, see figure 6.5.

FIGURE 6.5. PLOT OF CUMULATIVE HAZARD AGAINST TIME FOR FORMULATION (3)



On taking logs, the Weibull distribution was fitted.

FIGURE 6.6. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR FORMULATION (3)



The regression equation (one missing value) is $\log(H(w_i)) = -1.19 + 0.785\log(w_i)$ with a coefficient of determination of 0.988 and two unusual observations, 2X and 29R so that the cumulative hazard is $H(w_i) = 0.3042 w_i^{0.785}$ so the Duane model is not applicable. The removal of the three outliers in the linear plot may change the formulation to the "Duane" type model.

6.3.4 HAZARD ANALYSIS OF FORMULATION (4)

A number of different distributions may be attempted to fit the hazard of formulation (4). In this instance, the plot of the hazard against time is almost linear except for the three last points as described previously, (figure 6.7). By taking logs of both axes, the plot produced is in figure 6.8.

FIGURE 6.7. PLOT OF CUMULATIVE HAZARD AGAINST TIME FOR FORMULATION (4)

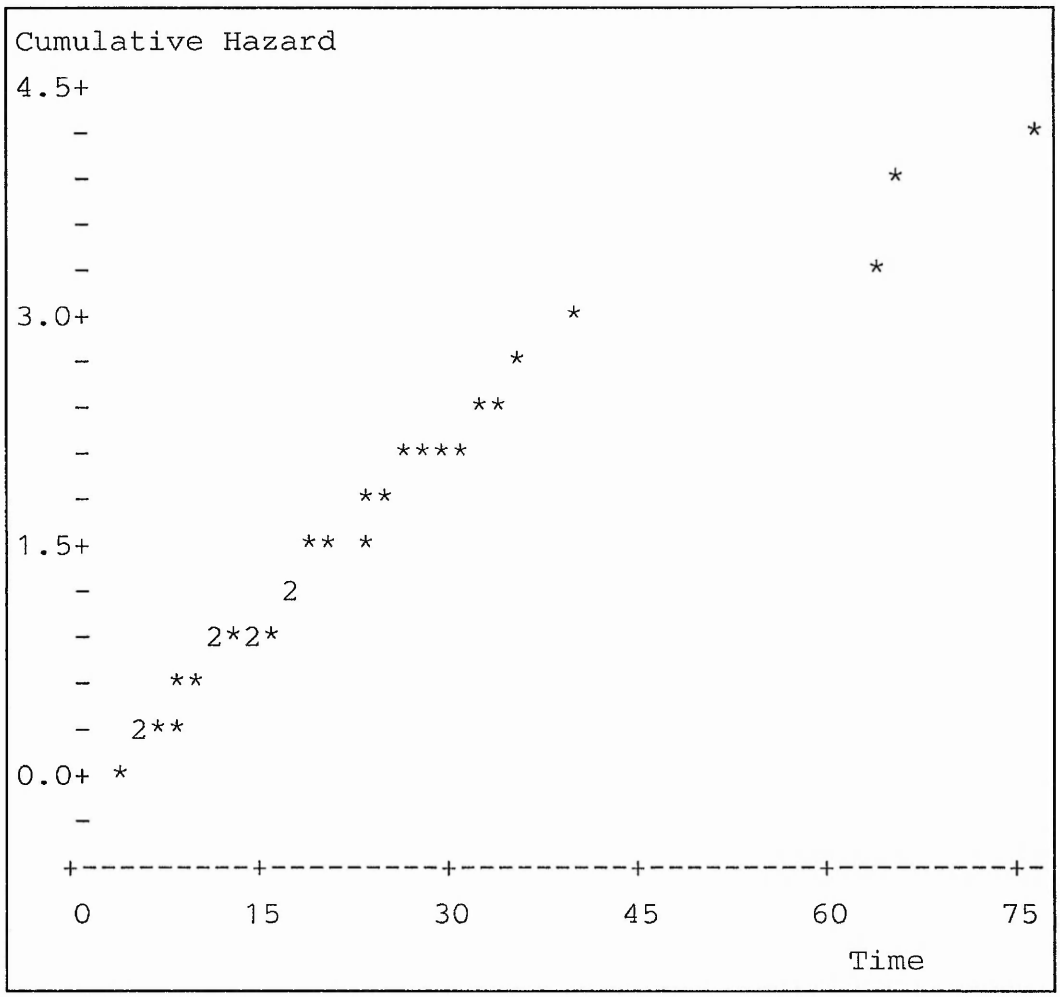
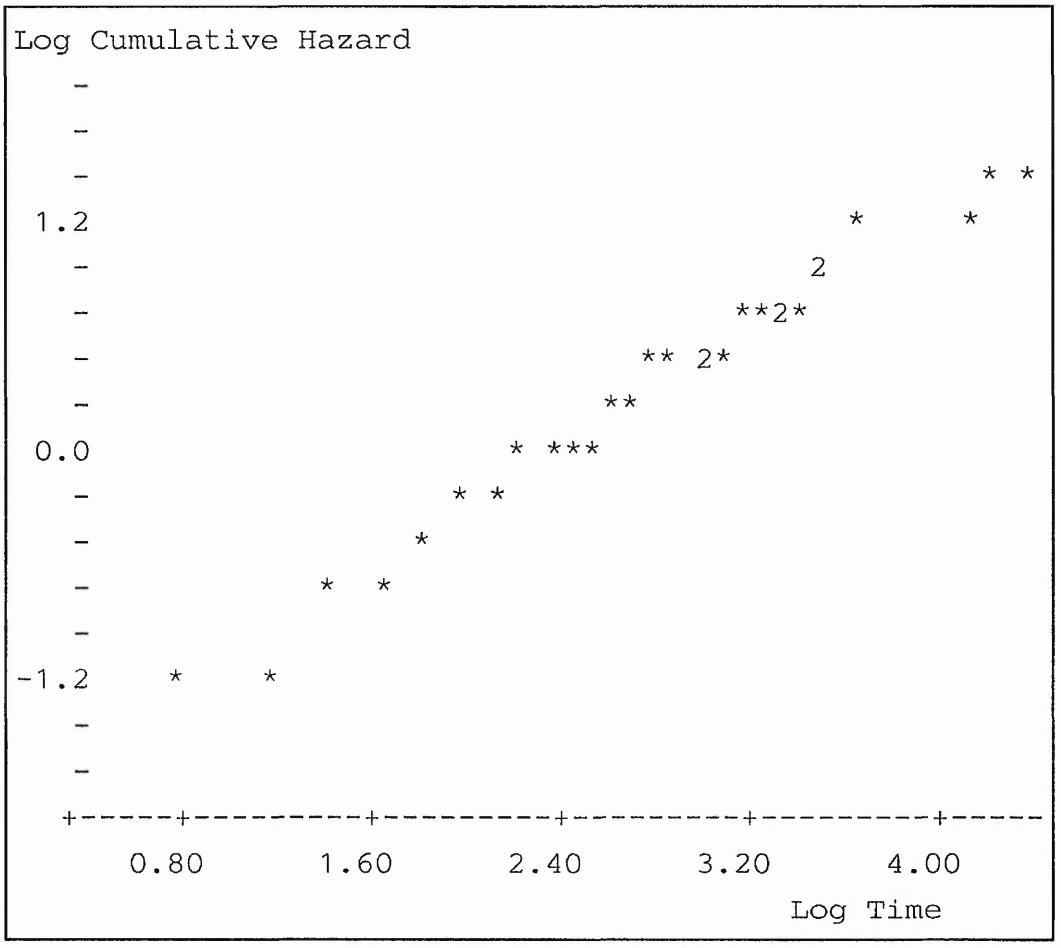


FIGURE 6.8. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR FORMULATION (4)



The estimates of the slope and intercept parameters when the log cumulative hazard is regressed against time are the parameter estimates for the Weibull distribution. The slope estimate of 0.77744 with a standard deviation of 0.01474 (not an exponential distribution) describes a wear-in phenomenon (i.e. when there are a number of long and very short waiting times). The regression equation is

$\log(H(w_i)) = -1.80649 + 0.77744\log(w_i)$ with one missing zero time value and three unusual observations, 2RX, 4R and 29R. The coefficient of determination is 0.99.

Step 4 of the 4 step procedure is to carry out diagnostic plots. This has not been done for the reasons outlined at the end of chapter 6.2.

It has been shown by the above analyses that the most appropriate well known NHPP which applies to this data set is the Goel-Okumoto model and that various PHM formulations also fit the data well. Also, sources 10 and 11 were found to be more reliable than the other sources.

6.4 STRATIFICATION OF WAITING TIMES TO FAILURE OF SOURCES

Proportional hazards modelling (PHM) was applied to the individual sources using time since last failure in days as the metric, i.e. the waiting time to next failure for each of the sources individually. This corresponds to model (4) in the Prentice, Williams and Peterson (1981) paper.

Step 1 of the 4 step diagnostic procedure is to determine appropriate covariates. Analysis was carried out using the covariates; age t_{i-1} as in the Goel-Okumoto and Musa models (GO and MU in table 6.3), number of failures $i-1$ as in the Binomial and Poisson Exponential Order Statistic and "logarithmic" type models (BE, PE, LOG in table 6.3), log of the number of failures $\log e(i)$ as in the "square root" type model and the "Duane" type model (SQ and DU in table 6.3), source version change and type of use for this within-sources variation. Information pertaining to programmer, repair date and repair programmer could not

be utilized as too much data was missing. The covariate, "source language", could not be incorporated as the twelve sources analysed were all in Cobol.

As has been shown previously in the EDA section, there was a high correlation between age and previous number of failures. Further analysis of this and other collinearity was carried out by multivariate techniques shown later in chapter 7. Multicollinearity usually occurs because of the data collection method, constraints on the model or in the population and/or model misspecification.

The covariates associated with the NHPP's discussed earlier were attempted to be modelled separately into a proportional hazards structure for each of the twelve sources analysed and in every case, even though some of the covariates were nonsignificant at the 5% level, the hazard decreased with increasing cumulative time to failure and also decreased with the increasing previous number of failures. The significant covariates are listed below. For three of the sources 274, 422 and 546 (labelled as number 3, 5 and 7) there were two versions of the software being tested at once, one on the test facility and one on the customer site. There was no significant difference between the hazard rates of versions one and two of each of the sources. The covariate "type of use" was also not significant. The reason for nonsignificance of the covariates for the sources is most probably attributable to the low sample sizes involved. This completes step 2 of the procedure.

TABLE 6.3. COVARIATE INFORMATION OF WITHIN SOURCES VARIATION

Source Number	No. of failures, faults, censors, source versions	Covariate	Value	p - value
489 (no. 6)	29,20,23,23	$i-1$ (BE, PE, LOG)	-0.036	0.00429
		t_{i-1} (GO, MU)	-0.012	0.00426
		$loge(i)$ (SQ)	-0.383	0.0109
606 (no. 9)	11,0,3,3	$i-1$ (BE, PE, LOG)	-0.292	0.0068
		t_{i-1} (GO, MU)	-0.013	0.0319
		$loge(i)$ (SQ)	-0.874	0.0166
737 (no. 12)	11,0,3,4	$i-1$ (BE, PE, LOG)	-0.484	0.0033
		t_{i-1} (GO, MU)	-0.024	0.010
		$loge(i)$ (SQ)	-1.222	0.0166

After determining the covariate structure, the baseline cumulative hazard was calculated using equation (7) in chapter 6.2 (step 3 of the procedure). The baseline cumulative hazard was then analysed in MINITAB to determine

whether the appropriate hazard to model a known NHPP in table 5.2 fitted the data and whether other hazard functions fitted the data well.

The following tables contain the coefficients of determination for each of the models which were fitted to the three source data sets in table 6.3 above. The models were

$\log(H_0(w_i)) = b\log w_i - b\log(a)$: (the Weibull distribution);

$H_0(w_i) = a + bw_i$: (the exponential distribution when a=0);

$\frac{H_0(w_i)}{w_i} = a + bw_i + cw_i^2$: (the quadratic hazard function) and

$H_0(w_i) = c + a(1 - e^{-bw_i})$: (the Gumbel distribution when c=0)

where $H_0(w_i)$ is the cumulative baseline hazard.

The goodness of fit of the models to the data was determined by the closeness of the coefficient of determination (the square of the correlation coefficient) to unity. This is shown below.

TABLE 6.4. TABLE OF COEFFICIENTS OF DETERMINATION FOR THE FOUR HAZARD MODELS FOR SOURCE NUMBER 6

Model	$i-1$	t_{i-1}	$\log(i)$
Exponential	0.944	0.941	0.958
Weibull	0.967	0.966	0.972
Quadratic	0.643	0.649	0.741
Gumbel	0.925	0.936	-

TABLE 6.5. TABLE OF COEFFICIENTS OF DETERMINATION FOR THE FOUR HAZARD MODELS FOR SOURCE NUMBER 9

Model	$i-1$	t_{i-1}	$\log(i)$
Exponential	0.968	0.938	0.962
Weibull	0.972	0.965	0.972
Quadratic	0.276	0.192	0.151
Gumbel	0.529	0.917	-

TABLE 6.6. TABLE OF COEFFICIENTS OF DETERMINATION FOR THE FOUR HAZARD MODELS FOR SOURCE NUMBER 12

Model	$i-1$	t_{i-1}	$\log(i)$
Exponential	0.899	0.934	0.905
Weibull	0.937	0.939	0.936
Quadratic	0.407	0.343	0.410
Gumbel	0.472	0.967	-

In every case, the Weibull provided a better fit than the exponential which is to be expected as the Weibull is a

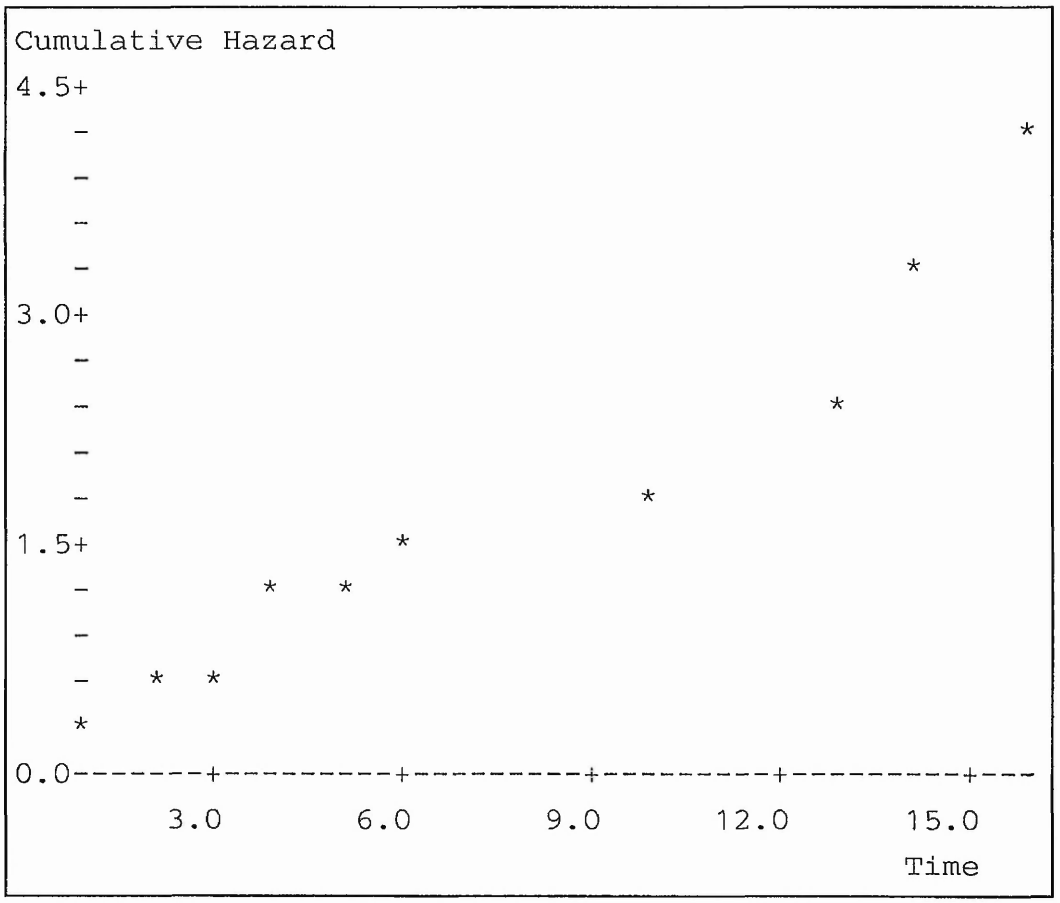
more flexible model. Where the Gumbel distribution was fitted (for the covariate t_{i-1} to see if the data fitted the Goel-Okumoto model), only source number 12 had a better Gumbel fit. The quadratic hazard function provided the worst fits in every case.

On developing the full proportional hazards model, the "log" type model for the failure count covariate, the Goel-Okumoto model for the total time covariate and the "Duane" type model for the log failure number covariate were the appropriate formulations to fit the data structure.

6.4.1 HAZARD ANALYSIS FOR THE COVARIATE $i-1$ PHM FORMULATION

As the estimate of the β value for the covariate failure count was negative, the appropriate models in table 5.2 are the "logarithmic" type model and the binomial type models where the baseline hazard is exponential and the Poisson type models where the baseline hazard is Gumbel. By plotting cumulative hazard against time and seeing whether the plot is linear and goes through the origin, the exponential distribution is shown to be appropriate. This was carried out for the three sources and then the cumulative hazard was regressed against time to see if the constant in the regression was significantly close to zero assuming normal errors. In each case this was true, so regression was undertaken with the constant removed to provide an estimate of the constant hazard rate of 0.233, 0.175 and 0.195 failures per day for each of the sources 6, 9 and 12 respectively. A plot of cumulative hazard against time for source 6 is shown in figure 6.9.

FIGURE 6.9. PLOT OF CUMULATIVE HAZARD AGAINST TIME FOR SOURCE NUMBER 6 AND COVARIATE $i-1$



6.4.2 HAZARD ANALYSIS FOR THE COVARIATE t_{i-1} PHM FORMULATION

For the time covariate model, the appropriate NHPP's are the Cox-Lewis model, the Musa model and the Goel-Okumoto model. The Goel-Okumoto model and Musa model provide estimates of the initial number of failures in the sources by the values calculated for parameter, a and a/b, and so further analysis is restricted to these models.

The a and b parameters for both models $E(N(t_i)) = a y_i = a(1 - e^{-bt_i})$ and $E(N(t_i)) = \left(\frac{a}{b}\right) y_i = \frac{a}{b}(1 - e^{-bt_i})$ for the three sources are :

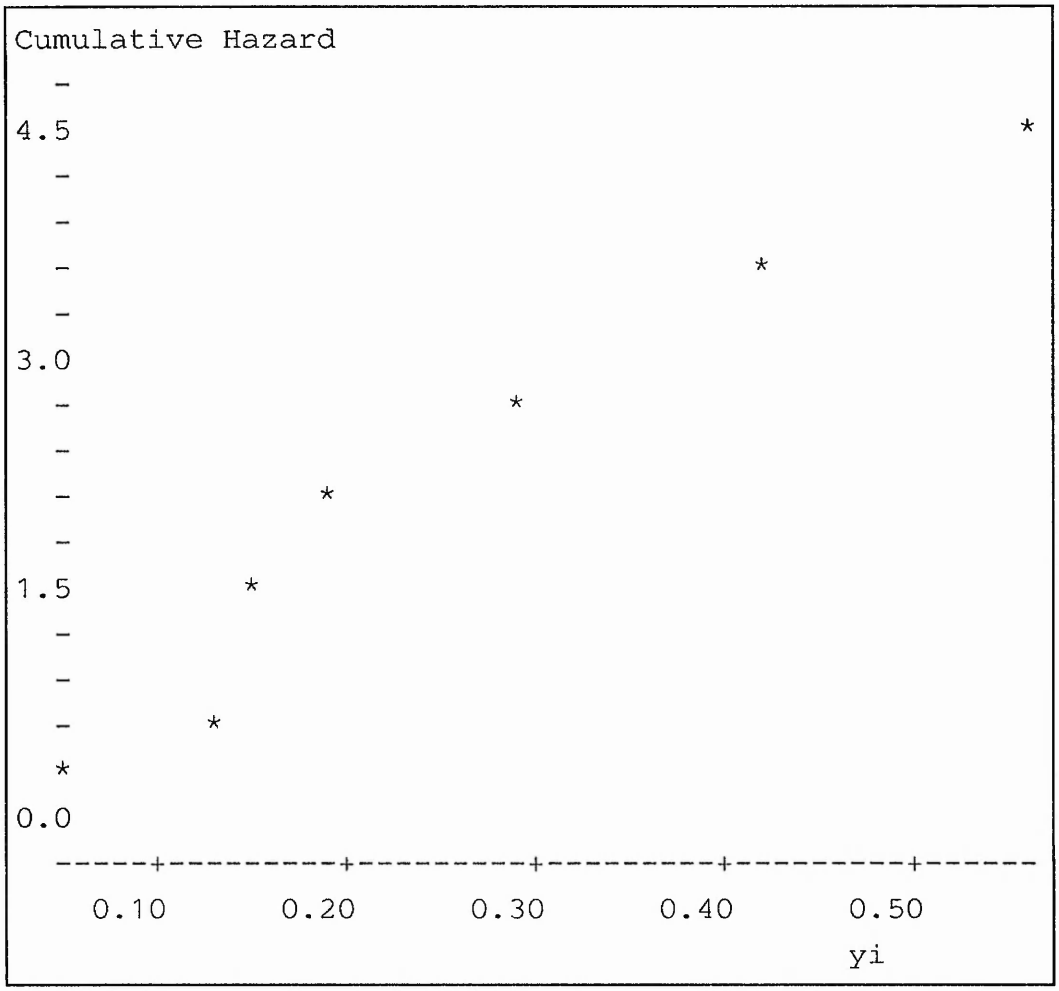
TABLE 6.7. TABLE OF ESTIMATES FOR THE GOEL-OKUMOTO MODEL

Parameters / Source Number	a for GO	b	a for MU
6	21.034	0.01177	0.2475
9	9.9082	0.01281	0.1269
12	8.4179	0.02401	0.2021

It can be seen that in both models, the initial number of failures is very optimistic as each data set have already experienced more than these initial number of failures and indicates that this is a bad model for the data.

A plot of cumulative hazard against y_i for source number 12 is shown in figure 6.10.

FIGURE 6.10. PLOT OF CUMULATIVE HAZARD AGAINST Yi FOR SOURCE NUMBER 12

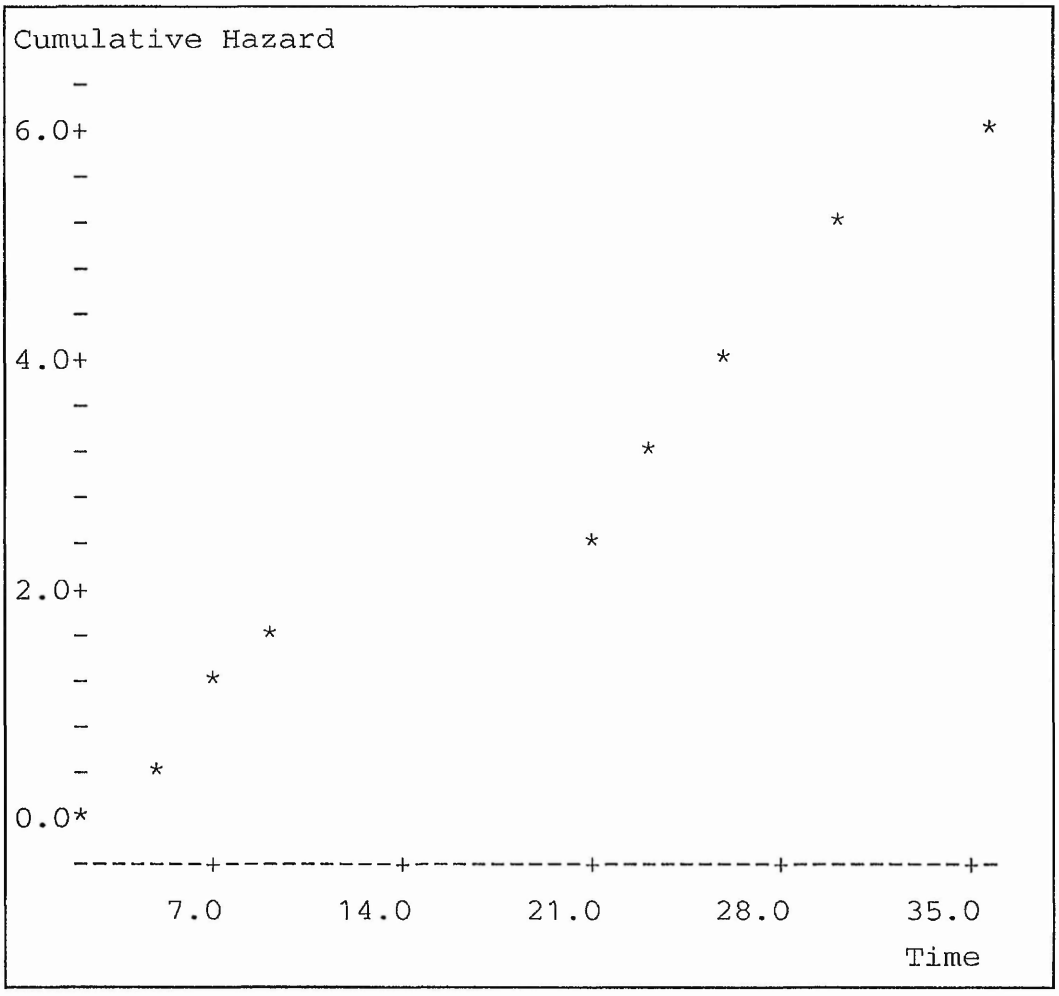


6.4.3 HAZARD ANALYSIS FOR THE COVARIATE $\log(i)$ PHM FORMULATION

In table 5.2, the appropriate PHM formulations for the log covariate are the "Duane" type model and the "square root" type model which is a special case of the "Duane" type model. The covariate value for the log covariate takes the form $1-\frac{1}{b}$ and equals -0.3832, -0.874 and -1.223

for the sources 6, 9, 12 respectively. As the β values are asymptotically normal, then confidence intervals may be calculated and in the latter two cases, the "square root" type model was an appropriate PHM formulation to model the data. On regressing cumulative hazard against time, the constant was found to be not significantly different from zero assuming normal errors and the estimate of the hazard rate was calculated as 0.265, 0.157 and 0.186 for the three sources 6, 9, 12 respectively. The "Duane" growth rates, b, for the three models were calculated to be 0.7230, 0.5336 and 0.4499 which indicated reliability growth and the estimates of the scale parameter in the "Duane" type model are 2.733, 3.469 and 2.418 for the respective three sources. A plot of the cumulative hazard against time for source number 9 is shown in figure 6.11. Step 4 of the 4 step procedure has not been carried out for the previous reasons given.

FIGURE 6.11. PLOT OF CUMULATIVE HAZARD AGAINST TIME FOR SOURCE NUMBER 9



6.4.4 WEIBULL HAZARD PLOTTING

Two parameter Weibull hazard plots were carried out on each of the formulations. The estimates of the Weibull parameters for each of the formulations is given in table 6.8. As can be seen, the Weibull parameters of the same sources do not vary much with covariate. A Weibull plot for each source covariate for one of the sources appears in figures 6.12, 6.13 and 6.14.

TABLE 6.8. TABLE OF TWO PARAMETER WEIBULL PARAMETER ESTIMATES FOR THE NINE MODELS

Covariate / Model	$i-1$	t_{i-1}	$\log(i)$
6 a	0.3272	0.3187	0.4415
6 b	0.8449	0.8484	0.7848
9 a	0.0972	0.0665	0.1038
9 b	1.1836	1.1414	1.1267
12 a	0.3543	0.2428	0.3413
12 b	0.8258	0.8489	0.8217

FIGURE 6.12. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR SOURCE 6 AND COVARIATE t_{i-1}

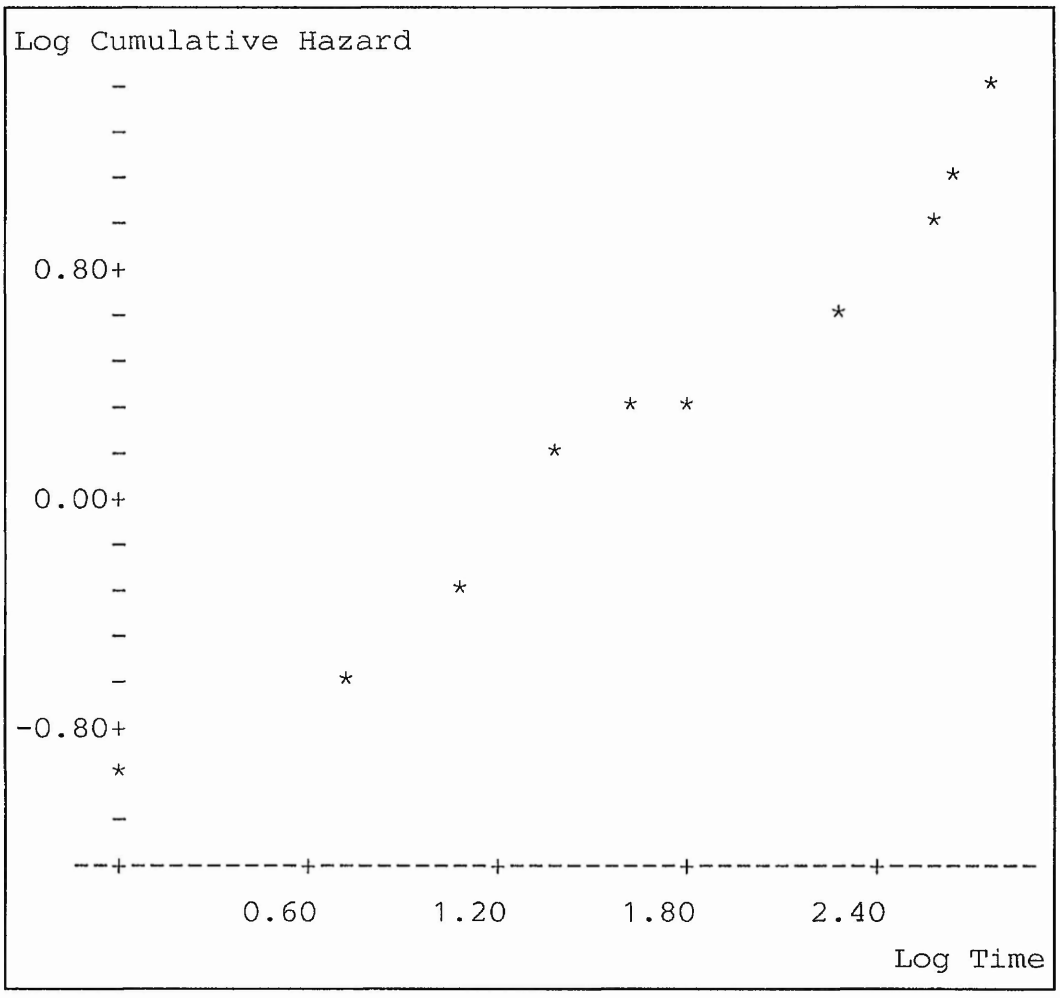


FIGURE 6.13. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR SOURCE 12 AND COVARIATE $i-1$

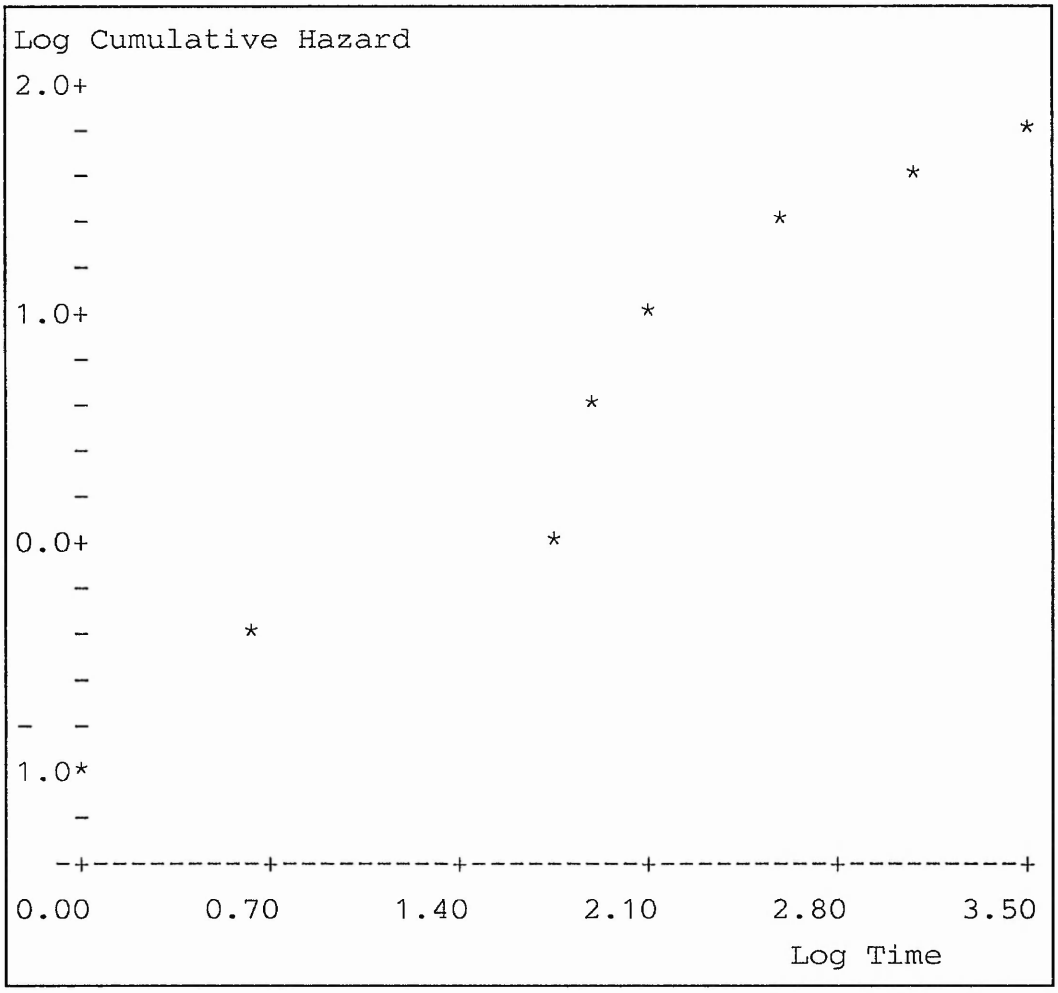
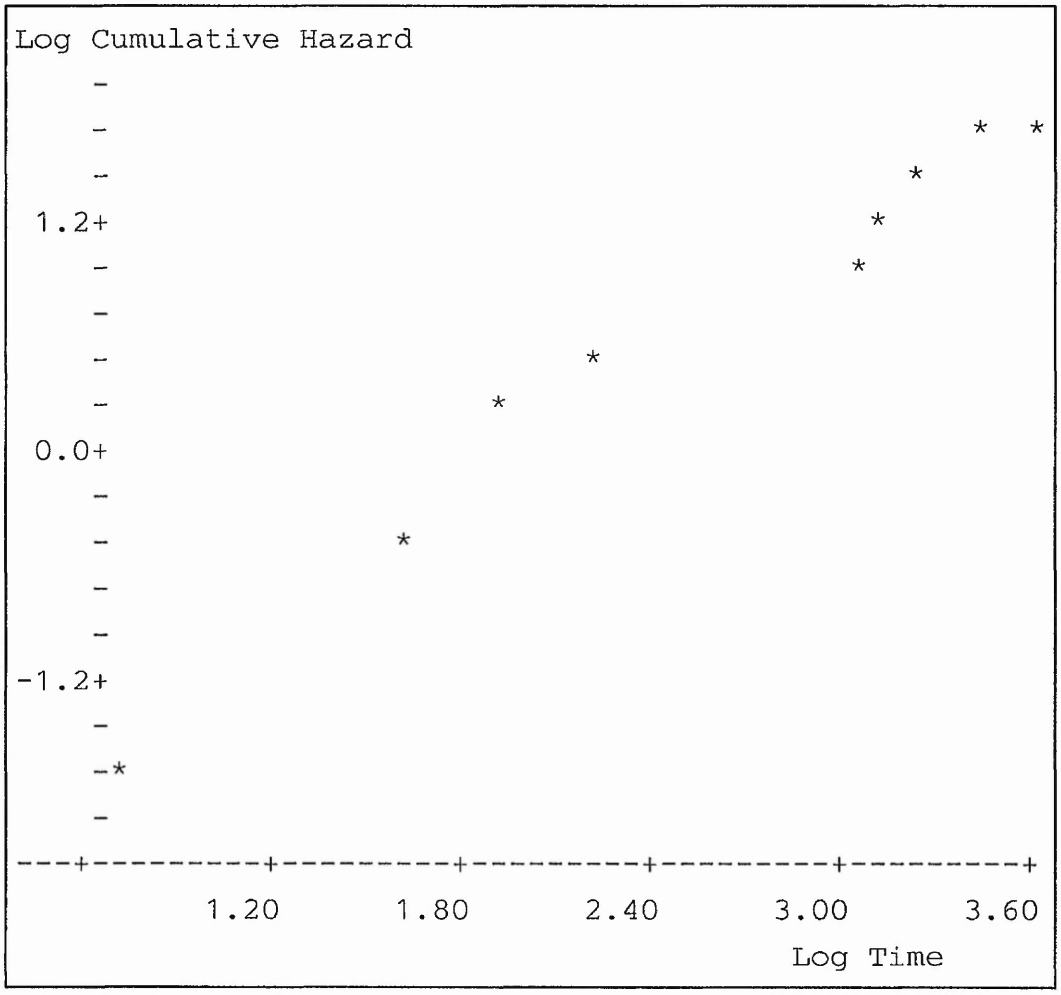


FIGURE 6.14. PLOT OF LOG CUMULATIVE HAZARD AGAINST LOG TIME FOR SOURCE 9 AND COVARIATE log(i)

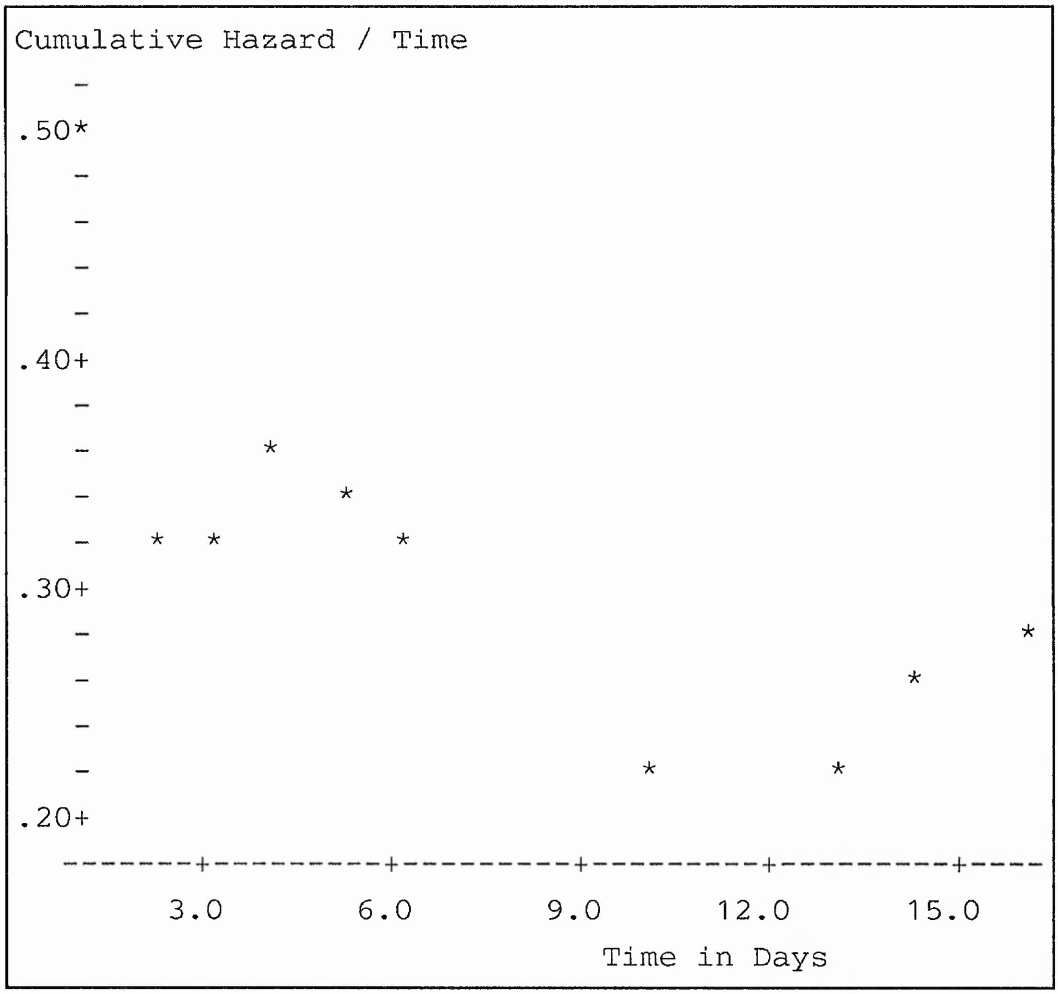


6.4.5 QUADRATIC HAZARD FITTING

The quadratic hazard model provided one reasonable fit to the data for source 6, covariate log(i). The plot of cumulative hazard over time is plotted against time and shows a reasonably quadratic shape to the data. The

regression equation is
$H(w_i) = 0.4717w_i - 0.0410w_i^2 + 0.00181w_i^3$ which results in a hazard formula $h(w_i) = 0.4717 - 0.0820w_i + 0.00543w_i^2$.

FIGURE 6.15. QUADRATIC HAZARD PLOT FOR SOURCE NUMBER 6 AND COVARIATE $\log(i)$



6.5 PHM ANALYSES USING SOFTWARE ATTRIBUTES

In an analysis discussed in McCollin, Wightman and Bendell (1989), the hazard rate was found to be not significantly different for different source sizes. A reason for this may be that the time metric (i.e. days to failure) is inappropriate for this covariate. Different source sizes affect the cpu time directly and the calendar time only indirectly by the cpu time. It was stated by the data supplier that the software was continuously operating for the duration of the project so that the calendar time between failures was the same as the cpu time between failures accumulated for the complete software package. However, it is not possible to relate the hazard function based on cpu times to source failure to size unless certain conditions hold true for the usage of the individual sources (i.e. all the sources are being run for similar lengths of time which is not very likely).

As PHM uses the ranking of the failure times and not the failure times themselves, it can be shown that as long as the ranking of the days between failures remains the same for cpu time, execution time or operating time between failures, then the conclusions concerning the hazard for the metric days to failure are valid for the other time metrics. For example, if execution time can be controlled so that it is always the same function of calendar time, e.g calendar time equals a constant multiplied by execution time, then conclusions about the calendar time hazard function will apply to the execution time hazard function.

In this analysis, the covariate size was not significant possibly because the assumption of continuous operation of the whole software package was not a function of the

twelve sources time to failures.

PHM was applied to the twelve sources with the six different types of source as a covariates. These covariates were found to be significant in the formulation and the hazard rate for source type 3 was found to be less than the other five types. Further modelling of the software attributes is given in chapter 7.

6.6 SUMMARY

A 4 point diagnostic procedure has been described and applied to Alvey data set number 3. The procedure aids the statistical analyst in determining the most appropriate NHPP to model a data set.

In conclusion, proportional hazards modelling has been used to model the software collection process, the software attributes and the software product and has provided valuable insight of the structure of diverse data sets in terms of the comparative reliability of sources and the ability to model the data sets taking into account age, previous number of failures and different hazard rates.

7 MULTIVARIATE TECHNIQUES

The analysis of failure counts using Proportional Hazards Modelling (PHM) showed that a number of explanatory factors were collinear.

The problem of multicollinearity of the covariates was investigated by applying multivariate techniques to the data set and the results of this are described under the task 4 work heading of the Alvey SRM project and in McCollin et al (1990).

The purpose of this section is to describe attempts to analyse Alvey software data set number 3 by multivariate methods. The data for the analyses was expressed in the form for suitable multivariate analysis however the work in chapters 7.1, 7.2 and 7.3 was carried out by Peter Dixon and the work in chapters 7.6.1 to 7.6.4 was mainly carried out by Dr. David Wightman.

The following variables were considered (how this set of data was created is described in chapter 3.6) :

X_1 = source	X_6 = type of source
X_2 = source version	X_7 = first appearance
X_3 = programmer	X_8 = final appearance
X_4 = language	X_9 = number of faults
X_5 = size of source	X_{10} = time

Data screening and editing were necessary to overcome idiosyncrasies and to render the data meaningful and suitable for analysis. The screening and editing were undertaken using MINITAB. Subsequent analysis was undertaken using MINITAB and GLIM.

7.1 DISCRIMINANT ANALYSIS

Multivariate discriminant analysis is a technique which allows the multivariate response for

$\underline{X}_*^T = \{X_1, X_2, \ldots, X_j, \ X_{j+2}, \ldots, X_p\}$

to be attributed to known groups according to X_{j+1} provided X_{j+1} is a group indicator, via discriminating functions. The discriminating functions then may be used to assign further observations on $\underline{X}_*^T$, not so far identified on a X_{j+1}, to a X_{j+1}.

A feature of the software data is that, in a number of cases, the multivariate response has not been identified by programmer (X_3). It is of interest to use the data on cases where the programmer is known as a "learning set" for discriminating between programmers, thereby making it possible for cases in the "prediction set", with programmer unknown, to be identified with a programmer.

The procedure is to calculate

$$w_i = \underline{L}_i^T \underline{X}_* - 0.5 \underline{L}_i^T \overline{\underline{X}}_{*j} + \ln(\pi_i) \ ; \quad i = 1, 2, \ldots\ldots, j, j+2, \ldots\ldots m$$

where m is the number of distinct groups (programmers) indicated by X_{j+1}, $\overline{\underline{X}}_{*_i}$ is the mean vector for group i, S_* is the pooled within groups estimate of Σ_*, the variance-covariance matrix of $\underline{X}_*$ and $\underline{L}_i = \underline{S}_*^{-1} \overline{\underline{X}}_{*i}$, π_i is the prior probability that a case belongs to group i, and to allocate the individual to that group for which the w_i is the greatest, (Chatfield and Collins (1980)).

Unfortunately, the success rate for correctly identifying the multivariate response on $\underline{X}_*$ by known programmer in

the training set was found to be low, with only 25.5% of cases correctly identified for the 30 programmers working on the project.

However, the success rate varied from programmer to programmer, ranging from no cases correctly identified to 88.9% of cases correctly identified. With a low overall success rate it is inappropriate to attempt to identify programmers for cases in the prediction set.

It is possible that the failure of the technique to achieve a reasonable success rate may be attributed to a violation of the theoretical assumptions of discriminant analysis, that the discriminating variables have a multivariate normal distribution and have equal variance - covariance matrices within groups, (programmers); (Chatfield and Collins (1980)). The data under study, consisting mainly of variables having a discrete or categorical nature, do not conform to these requirements. Goldstein and Dillon (1978) give a discussion on techniques of discrete discriminant analysis applied to data not conforming to the multivariate normal, homoscedastic (equal variances) groups pattern. Bishop, Fienberg and Holland (1975) give a similar treatment.

7.2 PRINCIPAL COMPONENTS ANALYSIS (PCA)

A commonly used multivariate technique is that of Principal Components Analysis, where p correlated variables are combined to obtain a new set of uncorrelated variables, called Principal Components (PC's). This method is well documented and appears as a standard technique within the MINITAB statistical package. It is regarded as a sub-model of correspondence analysis (by Hill (1973)) and this

reference shows that contingency tables, canonical correlation analysis and principal components analysis are a special case.

The new variables are linear combinations of the original variables and are derived in decreasing order of importance so that PC(1) accounts for as much as possible of the variation in the original data. If the first few components account for most of the variation in the original data, the effective dimensionality of the problem is less than p.

Let $\underline{X}^T = \{X_1, X_2, \ldots, X_p\}$ be a p-dimensional random variable with variance-covariance matrix Σ and let

$$Y_j = a_{1j}X_1 + a_{2j}X_2 + \ldots\ldots + a_{pj}X_p = \underline{a}^T\underline{X} \quad , \qquad (j = 1, 2, \ldots, p)$$

where $\underline{a}^T = \{a_{1j}, a_{2j}, \ldots, a_{pj}\}$ such that $\underline{a}_j^T\underline{a}_j = \sum_{k=1}^{p} a_{kj}^2 = 1$ and

$\underline{a}_i^T\underline{a}_j = 0, \quad (i <> j)$

Y_1 is found by choosing a_1 so that Y_1 has the largest possible variance, Y_2 is found by choosing a_2 so that Y_2 has the next largest variance and is uncorrelated with Y_1; Y_3 is found by choosing a_3 so that Y_3 has the next largest variance and is uncorrelated with Y_1 and Y_2, ... and so on.

Thus obtained, $Y_1, Y_2, \ldots, Y_p$ are the Principal Components (PC) of x having variance equal to the eigenvalues of the sample variance - covariance matrix S ($= \hat{\Sigma}$); (Chatfield and Collins (1980)).
In the case of the software data it is wise to base PCA

on the sample correlation matrix P rather than S, thus rendering the variables, which are heteroscedastic (different variance), equally important.

The MINITAB results were:

(i) Examination of the correlation matrix P showed a sufficiency of non-zero elements to warrant the PCA worthwhile.

(ii) Eigenanalysis of P.

TABLE 7.1. TABLE OF EIGENANALYSIS RESULTS FOR PCA

i	1	2	3	4	5	6
Eigenvalue λ_i	2.25	1.52	1.12	0.98	0.64	0.42*
Proportion $\lambda_i/\Sigma\lambda_i$	0.32	0.22	0.16	0.14	0.09	0.06
Cumulative Proportions	0.32	0.54	0.70	0.84	0.93	0.99

(* denotes that subsequent eigenvalues exist but account for only 1% of the variation).

Note that as many as five PC's are required before more than 90% of the variation in the data is explained. Ideally, it is desirable that the majority of the variation in the data should be explained by two or three components at the most. Unfortunately no such reduction of the effective dimensionality was obtained. Reduction to two or three components is useful in that 2D or 3D plots of component score might be examined for patterns or clusters and that attempts at reification (a physical relationship between

number of faults and source size, type, etc) might be made. However, it is of some interest that the effective dimension of the data reduces to about five, with this technique.

MINITAB also supplies the coefficients $\underline{a}_j^T$ from the eigenvectors corresponding to each eigenvalue.

7.3 LOG-LINEAR MODELS

It is possible to obtain from the software data multi-way tables containing number of faults as response corresponding to variables such as X_4 = source language, X_5 = source size and X_6 = source type. With such categorical data it is appropriate to fit log-linear models, beginning with the no-association model.

$$E(F_{ijk}) = N \pi_{i..} \pi_{.j.} \pi_{..k} \qquad (1)$$

where F_{ijk} = number of faults in the cell of the multi-way table corresponding to the i'th language, j'th source size, k'th source type;

N = grand total of faults in the multi-way table;

$\pi_{i..}$ = marginal probability in the i'th category of X_4 (language) irrespective of X_5 and X_6 (size and type of source), $\pi_{.j.}$ = marginal probability in the j'th category of X_5 (size) irrespective of X_4 and X_6 (language and type); $\pi_{..k}$ = marginal probability in the k'th category of X_6 (type) irrespective of X_4 and X_5 (language and size).

Taking logarithms in (1)

$$\ln E(F_{ijk}) = \ln N + \ln \pi_{i..} + \ln \pi_{.j.} + \ln \pi_{..k} \qquad (2)$$

With a little manipulation it is possible to write (2) in the form

$$\ln E(F_{ijk}) = u + u_{1(i)} + u_{2(j)} + u_{3(k)} \ , \qquad (3)$$

where the u's are functions of the theoretical marginal fault counts.

Now, (3) is reminiscent of a three-way ANOVA model, with no interaction. It is possible to fit (3) using GLIM, employing the deviance statistic equal to $-2\log(l_c/l_f)$ as the goodness-of-fit criterion, where l_c = likelihood of the data under the current model and l_f = likelihood of the data under the fullest possible model, following the notation of Baker and Nelder (1978).

Failure of the no-association model to fit the data encourages the inclusion of further model terms, firstly the two-way associations

$$u_{12(ij)}, \quad u_{13(ik)}, \quad u_{23(jk)}$$

corresponding to first-order interaction in ANOVA, and then, if necessary, the three-way association $u_{123(ijk)}$, corresponding to second-order interaction in ANOVA, (see Everitt (1977)).

TABLE 7.2. TABLE OF LOG-LINEAR MODELLING RESULTS

Model	Scaled deviance	change	residual df	change in df
A (3)	89.75			
B A+Size.Type	77.78	11.97	11	5
C B+Size.Language	26.94	50.84	10	1
D C+Type.Language	0.44	26.50	5	5

7.3.1 CONCLUSIONS OF LOG-LINEAR MODELLING

The scaled deviance (or change in scaled deviance) is approximately X^2 - distributed with the residual degrees of freedom (or change in degrees of freedom) from which it can be concluded that

(a) there is a significant association between size and type of source,

(b) there is a significant association between size of source and language,

(c) there is a significant association between type of source and language,

(d) there is no significant three-way association, suggesting that

(i) the association between size of source and type of source is the same for all languages,

(ii) the association between size of source and language is the same for all source types,

(iii) the association between type of source and language is the same for all source sizes.

Resulting from (b), close examination of the model parameters suggests that a negative association between Size (2) and Type (5) variables is indicative of a tendency for a lower fault count with medium to large sources than with small sources of the type "Include file".

Also, resulting from (c), a negative association between Type (4) and Language (2) variables suggests a tendency for a lower fault count with system operating language sources than with COBOL source programs of the type "Find control file".

The results (a) to (d) are to be expected since some of the types of source in a given language are used for specific common activities, e.g. calling routines to another source and so these will be the same sources (and hence the same size) because they have been written to be re-useable.

The multivariate procedures described earlier in this section revealed relatively little. However this should not malign the power and usefulness of techniques such as PCA and discriminant analysis, and they should be used if appropriate on other examples of software data in attempts to reveal data structure.

Log-linear modelling, a useful example of which is discussed above, has a very positive usefulness in

investigating data of the type considered and is recommended as an important tool in future work.

7.4 GENERALISED LINEAR MODELLING

Generalised linear modelling is a commonly applied approach within the area of medical statistics when determining the proportion of subjects affected by different amounts of a drug. A standard textbook for generalised linear modelling is by McCullagh and Nelder (1983). A paper in the area of exposure to a disease using the models described here is by Kleinbaum, Kupper and Chambless (1982). Also, a report was written for the REQUEST project on the application of generalised linear modelling to one of their collected data sets (Hufton, Quinn and McInnes (1989)). However, generalised linear modelling has not been utilised to any great extent previously in the analysis of software reliability. The methods are used here to analyse the proportion of a package of sources of a certain type which have failed by a given time. The background to this analysis is in chapter 3.4 with the plot of the proportion in figure 3.5.

A generalised logistic regression model is formulated as follows. The proportion of sources of a certain type failing at a particular instant in time is calculated and a transformation is taken so that when plotted against a linear combination of continuous variables (e.g; time since last failure, cumulative time to failure), a straight line results. The analyses carried out with the data under the model assumption are then tested for goodness of fit.

7.4.1 DESCRIPTION OF MODELS

Three model formulations which have been applied to the described data are illustrated in figures 7.1, 7.2 and 7.3. Each figure shows a function of the proportion of failures/faults per day, g(p), (where p is the proportion), plotted on the vertical axis with the explanatory variable of interest, x, on the horizontal axis. The function g(p) is chosen so that the plot should be linear. In figure 7.1, model (1) is

$$g(p) = \beta_0 + \beta_1 x + \epsilon$$

irrespective of the type of source, which can be tested for goodness of fit. The term ϵ is an error term which explains any variation not already described by the parameters of the model. Model (2) is

$$g(p) = \beta_0 + \beta_1 x + \alpha_i + \epsilon \qquad i = 1, 2, \ldots, k.$$

If the model is not significant on the factor, α_i = type, for the 6 types of source, then the plot is similar to figure 7.1. However if the proportion of failures/faults is affected by the type of source, then figure 7.2 is more appropriate. Model (3) is given by

$$g(p) = (\beta_0)_i + (\beta_1)_i x + \alpha_i + \epsilon$$

and, in this case, if the interaction between the type of source and the explanatory variable is not significant then figure 7.2 applies and if there is an interaction between the type of source and the explanatory variable, the plot will be similar to figure 7.3.

FIGURE 7.1. ILLUSTRATIVE PLOT OF g(p) AGAINST x WITH MODEL (1) FITTED

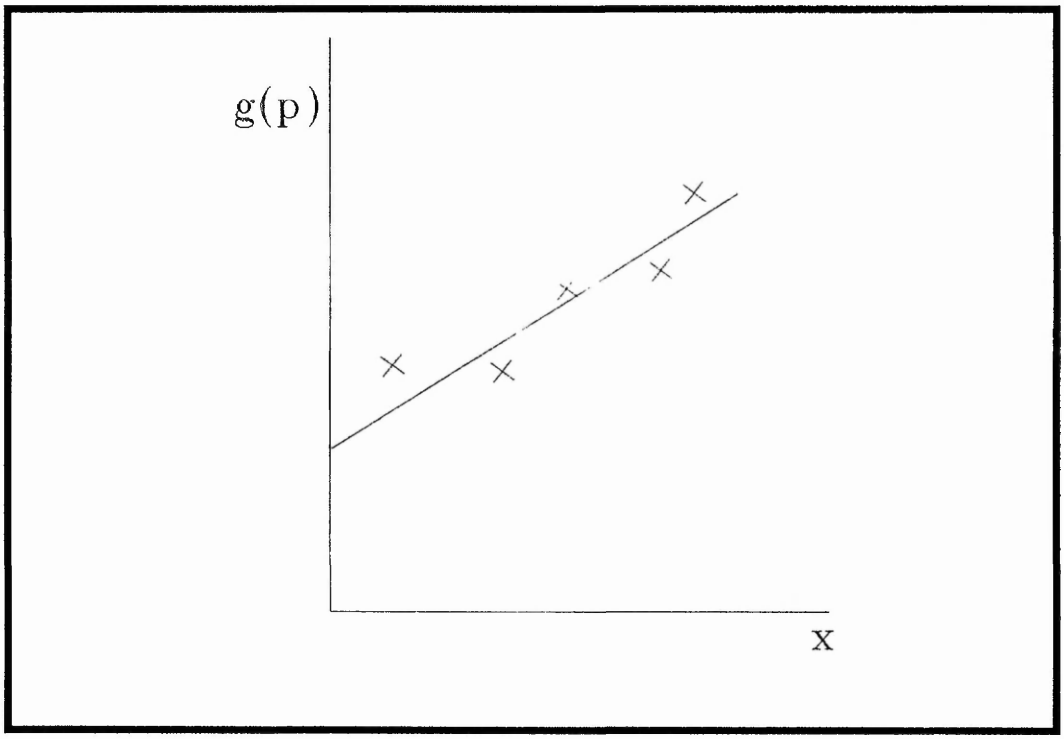


FIGURE 7.2. ILLUSTRATIVE PLOT OF g(p) AGAINST x WITH MODEL (2) FITTED

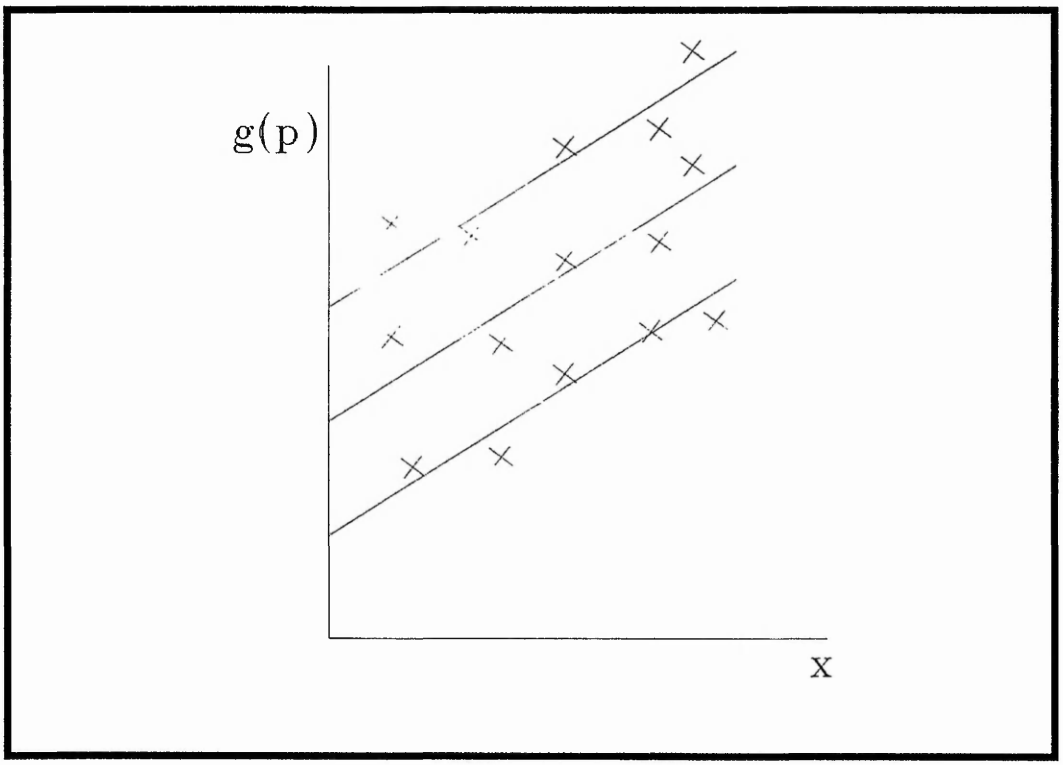
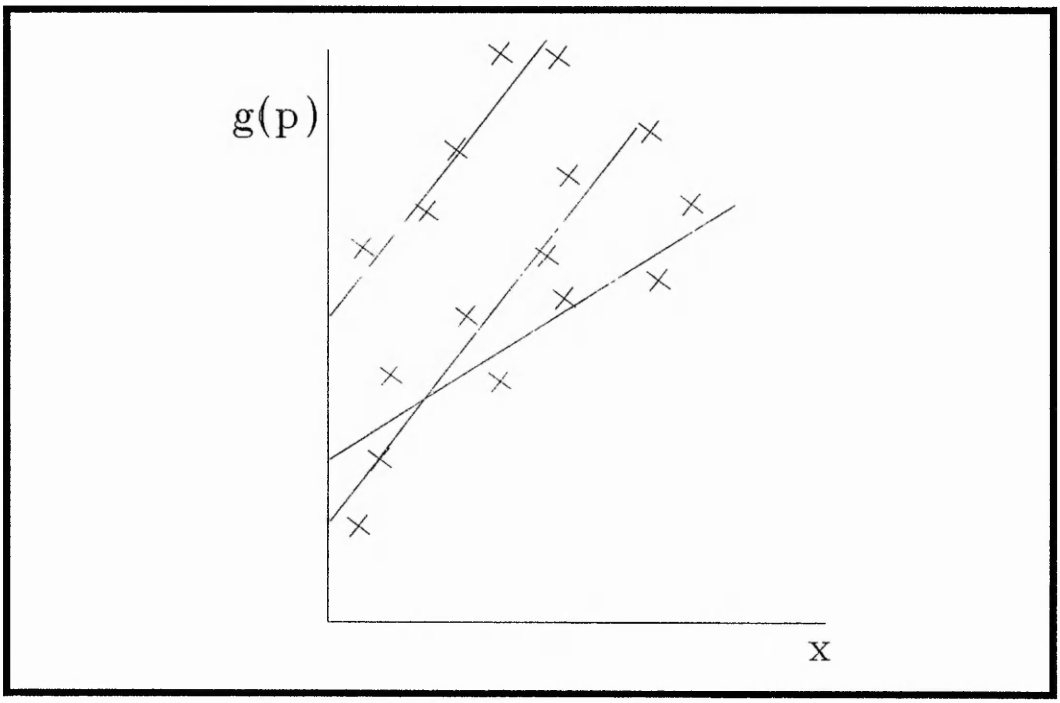


FIGURE 7.3. ILLUSTRATIVE PLOT OF g(p) AGAINST x WITH MODEL (3) FITTED



7.4.2 DATA PLOTS AND ANALYSIS

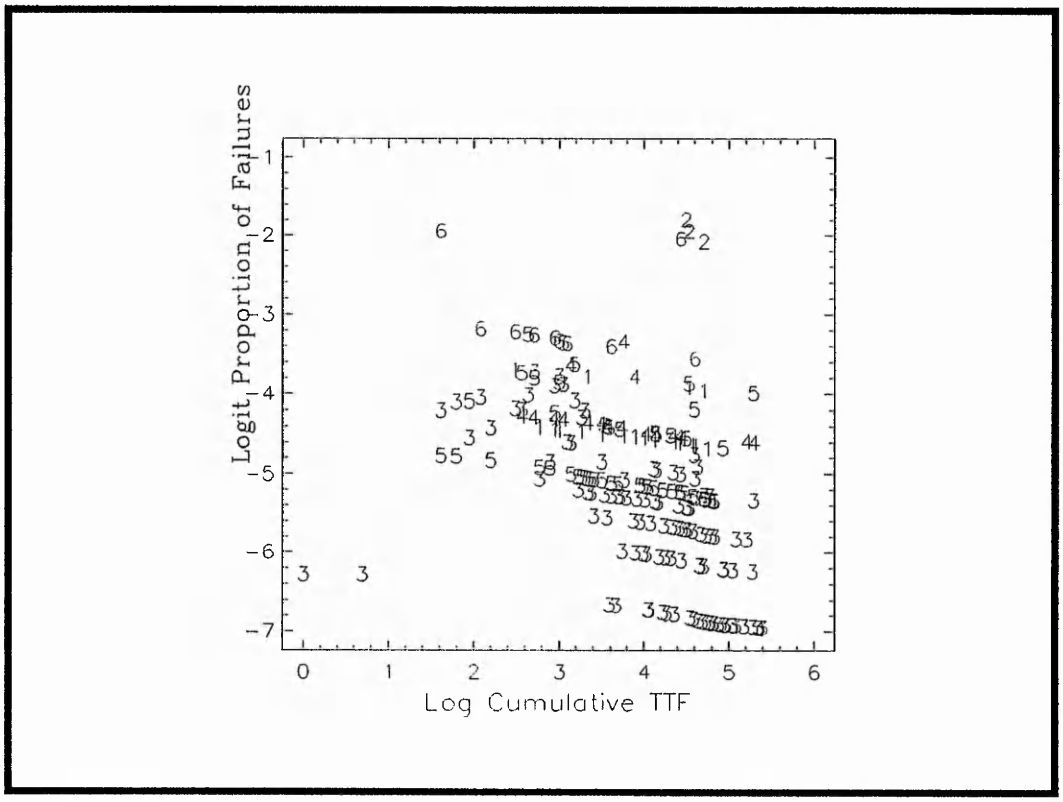
A number of plots of the data were drawn and subsequent analysis was carried out using the generalised linear modelling computer package GLIM, (Baker and Nelder (1978)). The models (1), (2), (3) were applied to proportion of failures per day, proportion of faults per day, proportion of faults and censorings per day and proportion of failures and censorings per day against cumulative time to failure and time to failure for three different types of function of the proportion. In each case there was still a lot of unexplained variation after the models were fitted. One way of reducing variation is to take a transformation of

the explanatory variables such as the logarithm or the square root. A recent reliability paper which incorporated plots of transformed data is by Follman (1990).
Further analysis was carried out, with the proportion of failures per day only, as this analysis showed this data to have the least variation after the initial model fits. Figure 7.4 is one such plot of the function of the proportion of failures per day

$$g(p) = \log(p/(1-p))$$

(known as the logit function) against the logarithm of the cumulative time to failure in days for the data in figure 3.5.

FIGURE 7.4. PLOT OF THE LOGIT PROPORTION OF FAILURES AGAINST LOG CUMULATIVE TIME TO FAILURE



From the figure, it can be seen that:

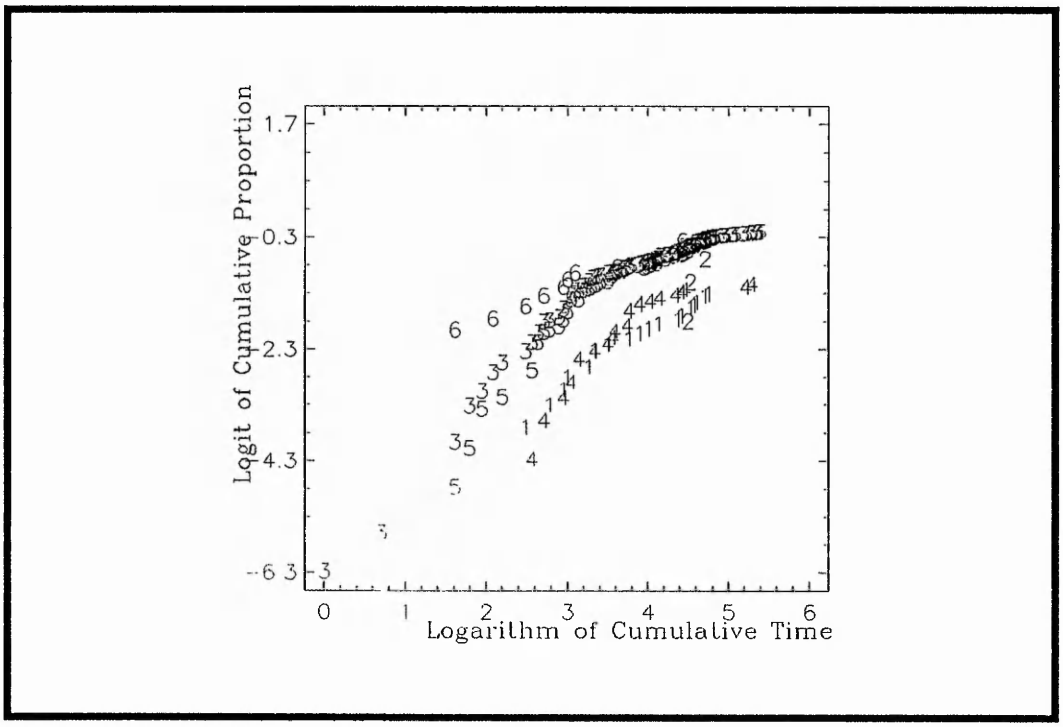
- there is a difference within types of source which is mainly due to the same proportion failing per day for the duration of the project. As there was very little change in the total number of sources running per day, this difference is mainly due to the number of failures of a given type failing per day.

- there is a difference between types of source, which can be accounted for by the number of failures for each type.

- the number of failures per day is reducing over time for each type. This suggests that there is reliability growth in the data which may be expected for a software development project.

This growth can be seen more clearly if instead of the proportion of failures per day, the cumulative number of failures up to a point in time is divided by the cumulative number of sources failed with the cumulative number of failures. The logit of the cumulative proportion of failures is shown plotted against the logarithm of cumulative time in figure 7.5. From figure 7.5, it can be seen that either model (2) or (3) is applicable to the data and this can be tested with GLIM.

FIGURE 7.5. PLOT OF LOGIT CUMULATIVE PROPORTION OF FAILURES AGAINST LOG CUMULATIVE TIME FOR EACH SOURCE TYPE



This model (2) is known as a proportional odds model (Pettit (1984), Aitken et al (1989), Crowder et al (1991)) and figure 7.5 shows a plot similar to a multiple Duane plot, (Duane (1964)). It is shown in chapter 7.5 that model (2) tends to the proportional intensity formulation with Weibull intensity (a multiple Duane plot). The plot in figure 7.5 is levelling off as failures are detected and removed. A more appropriate intensity to model this curvature is the IBM model (Rosner (1961)) and also discussed by Ascher (1968). In the Ascher formulation, the intensity is given by

$$\lambda(t) = c + ab^t$$

where 0 < b < 1 and c > 0. The cumulative intensity or expected number of failures in time t is given by

$$E(N(t)) = ct + a(b^t - 1)/\log(b).$$

A plot of the Ascher model for another subset of this data is shown in the later section on proportional intensity modelling.

Tables 7.3 and 7.4 present the results of the GLIM analysis. To test the goodness of fit of one model over another, the difference in deviances, (explained in the reference by Baker and Nelder (1978)), of each model is compared with the χ^2 distribution with degrees of freedom being equal to the lost degrees of freedom when fitting a more complex model. As an example of this test, for model (1) compared to model (2), the change in deviance is 480.6 on 2 degrees of freedom which is saying that the model in figure 7.2 is a much better fit as the tabulated value of χ^2 on 2 degrees of freedom at the 5% level is only 5.991. The comparisons of the models are given in table 7.3.

Assuming asymptotic normality of the parameter estimates, a simple test of whether the parameters should be included in the model is to see if the parameter estimates are within two standard errors of zero. If they are, then they do not contribute to the overall model. These nonsignificant parameters are then removed from the model and the new model parameters are estimated and tested until all nonsignificant parameters are removed. This method is known as backward stepwise regression.

For model (2), the type 3 sources were used as the baseline of comparison as the type 3 data sample size is greater than each of the other types and also the type 3 data is more central in figure 7.5 than the other types. The parameter estimates of type 2, type 5 and type 6 sources

were not significant within model (2). A reason for this may be the lack of observations for these types. A comparison of the parameter estimates of the types of source irrespective of whether they were significant or not was carried out and this determined that types 3, 5 and 6 parameter estimates were similar, types 4 and 1 were similar but different to the other types and type 2 was on it's own. Figure 7.5 shows these findings.

On fitting the interaction term (between type of source and log cumulative time to failure) with model (3), the deviance is 5.17 compared to the χ^2_2 tabulated value of 5.991 which indicates that the interaction term is not significant. This is confirmed by the parameter estimates for the interaction terms in model (3) being nonsignificant. The parameter estimates for each model are given in table 7.4. The results support the PHM analysis in chapter 6.5.

TABLE 7.3.

GOODNESS OF FIT ESTIMATES FOR THE GLIM ANALYSIS MODELS

Fitted Model	Deviance (degrees of freedom:df)	Difference in Deviance (df)	p -values
Model [1]	1086.3(220)		
Model [2]	605.73(218)		
Model [1],[2]		480.6(2)	0.00
Model [3]	600.56(216)		
Model [2],[3]		5.17(2)	0.08

TABLE 7.4.

ESTIMATES OF MODEL PARAMETERS FOR THE GLIM ANALYSIS

Fitted Model	Parameter	Estimate	Standard Error	p - values
Model [1]	β_0	-3.394	0.04046	0.00
,,	β_1	0.6363	0.00916	0.00
Model [2]	β_0	-3.31	0.04045	0.00
,,	β_1	0.6224	0.009147	0.00
,,	Type (1)	-1.126	0.0749	0.00
,,	Type (2)			n.s.
,,	Type (4)	-0.9258	0.07253	0.00
,,	Type (5)			n.s.
,,	Type (6)			n.s.
Model [3]	β_0	-3.299	0.04071	0.00
,,	β_1	0.62	0.009207	0.00
,,	Type (1)	-2.113	0.5369	0.00005
,,	Type (2)			n.s.
,,	Type (4)	-1.416	0.4063	0.00026
,,	Type (5)			n.s.
,,	Type (6)			n.s.
,,	$\beta_1.Type(1)$	0.2387	0.1276	0.0307
,,	$\beta_1.Type(2)$			n.s.
,,	$\beta_1.Type(4)$	0.1196	0.09715	0.1093
,,	$\beta_1.Type(5)$			n.s.
,,	$\beta_1.Type(6)$			n.s.

7.5 RELATIONSHIP BETWEEN RESPONSE MODELS AND PROPORTIONAL INTENSITY MODELS

As shown above, a model may be formulated as $g(p)=\beta_0+\beta_1\log(t)+\epsilon....(1)$ where the proportion p is the cumulative number of failures x up to time t divided by the total number of failures n, t is the cumulative time to failure and ϵ is a binomial error term. If g(p) is the logit function $g(p)=\log(p/(1-p))$ then (1) can be rewritten as

$$\log(x)-\log(n-x)=\beta_0+\beta_1\log(t)+\epsilon.$$

If n tends to a large constant such that n-x is very much greater than x, say k, and p is very small, the Poisson approximation may be used for the binomial response and instead of the logit function, the log of the number of failures is obtained. On taking exponentials of each side of the equation,

$$E(x)=e^{\beta_0+k}t^{\beta_1}$$

is obtained, where E(x) is the expected number of failures. This is the formulation of the nonhomogeneous Poisson process with Weibull intensity, (Crow (1975)).
Using the Poisson approximation for the logit function of the proportions resulting in the log of the number of failures, (known as the log link function; McCullagh and Nelder (1983)), then model (2) has been shown to be the proportional intensity formulation with Weibull intensity of Lawless (1987).

7.6 PROPORTIONAL INTENSITY MODELLING

In 1980, Lee considered comparing the intensities of independent Duane NHPP's as a hypothesis test however Lawless (1987) considers the situation where a number of individuals experience repeated events, with the time of each event recorded along with covariate information within a single model. Although the individual experiences a sequence of events, the covariate information for the individual is fixed. In a software reliability modelling context we can consider the individuals to be systems/packages/modules/languages etc; that is, a level of application in which a common baseline is reasonably thought to exist, or more commonly an application level at which data is available.

The methods discussed by Lawless are based on the proportional intensity Poisson process model. The model can be specified as

$$\lambda_x(t) = \lambda_0(t)\exp(\dot{x}\beta) \quad \ldots.(1)$$

where t is the time from the start of observation, $\lambda_0(t)$ the baseline intensity function, x a vector of covariate values and β a vector of parameters. The formulation in Lawless (1987) means that the covariates have a proportional effect on the baseline intensity function.

7.6.1 RELATIONSHIP TO PROPORTIONAL HAZARDS MODELLING

Lawless shows in section 4 of his 1987 paper the equivalence of proportional hazards modelling based on the partial likelihood construction of Cox, (1972), and the Poisson process with unspecified baseline. However, given the different approach to the construction of the partial likelihood and the likelihood for the semiparametric

Poisson process, it is not possible to use standard proportional hazards modelling software for the estimation of parameters in the Poisson case.
To carry out the analysis using proportional intensity models with covariates and unspecified intensity function, specific software has been written, details of which are given below.

7.6.2 MODEL FORMULATION

The model formulation and details of the likelihood equation are given in Lawless (1987). In particular, to obtain the β coefficients for the covariates the following set of equations have to be solved (the first partial differential of the log-likelihood),

$$\frac{\delta \log L(\beta)}{\delta \beta_r} = \sum_{i=1}^{m} n_i z_{ir} - \sum_{i=1}^{m} \{n(T_i) - n(T_{i-1})\} \frac{\sum_{l=i}^{m} z_{lr} e^{z_l \beta}}{\sum_{l=i}^{m} e^{z_l \beta}} \qquad , r = 1, 2, \ldots, k \qquad (2)$$

where z_{Sr} is the r'^{th} covariate value for the system s (s=1,2,3,....,m), n the number of failures for the system s, $n(T_i)-n(T_{i-1})$ the number of failures between the end of observation on system (i-1) and end of observation on system i (such that $t_1<t_2<....<t_m$).
To solve the equations in (2), a Taylor series expansion ($F(x)=\frac{\delta \log L}{\delta \beta}$) is used and then a Newton-Raphson iteration procedure; the method of scoring is applied. A source program has been written in Microsoft Fortran which runs on an IBM PC to estimate the parameters of the model.
In order to run the source program, a data file is created from the observed information:- the total number of failures, the total number of systems and the number of

covariates, whether covariates are included in the analysis, the final observation time for each system, the covariate values and the time to failure of each system. After re-ordering the data, the source proceeds with the estimation of the coefficients. Starting with initial values of zero for the coefficients, the Newton-Raphson equations are solved to provide a $\delta\beta$ value, where at the n^{th} iteration $\beta_n = \beta_{n-1} + \delta\beta$. The iteration procedure is continued until the incremental $\delta\beta$ for all the covariates is less that one thousandth of the existing β value or the number of iterations has reached 25 (indicating problems with convergence).

Upon convergence for β each coefficient is tested (using the asymptotic normality of the coefficient) to see if it is significantly different from zero. At this stage of the estimation procedure, the most non-significant covariate is dropped from the model (backwards stepwise regression) and the remaining β coefficients re-estimated. This procedure is continued until a set of significant (on a 5% two tailed test) β's are obtained. At this point desired information such as the β values, z-scores and p-values are reported in a computer output.

Having obtained a set of β coefficients, the source then calculates the base-line intensity function (using the formulation reported in 4.4 of Lawless (1987)) which is then available for comparison with well known intensity models. If only two systems exist and one binary covariate then it is possible to solve (2) directly, so that a partial check on the source may be performed. Carrying out this procedure, the same β value was obtained.

Musa et al (1987) and this thesis classify many of the well known software reliability models as non-homogeneous Poisson processes. It is therefore possible when applying

proportional intensity function models to compare the baseline intensity function against the intensity function for individual software reliability models.

7.6.3 COVARIATES

The covariates that can be included in the model, as with proportional hazards modelling, are obviously dependent on the context in which the data arises. However, from (our) observation on the model formulation it is noted that each system in proportional intensity modelling "plays" the same role as one failure in proportional hazards modelling. Thus there is a severe restriction on the number of covariates that may be included in any analysis if data is available only on a small number of systems. To carry out any meaningful analysis, data may have to be available on a large number of systems (Lawless in a medical example had 48 subjects). A recent paper which showed proportional intensities for software systems but did not carry out any intensity modelling is by Selby, (1990). The proportional intensity formulation software described above is applied to part of Alvey data set number 3.

7.6.4 ANALYSIS OF THE TWELVE LEAST RELIABLE SOURCES

Figure 7.5 indicates that the logit cumulative proportions against cumulative time for each source type are proportional to one another (i.e. vertical separation between types) and so an analysis of the reliability growth/decay for each of the twelve sources in figure 3.4 using proportional intensity modelling with cumulative time to

failure in days as the time metric appeared to be reasonable based on the plot and the approximate relationship between proportional odds and proportional intensity models.

The binary covariate, source designation, was used in this proportional intensity formulation. The baseline sources chosen were numbers 5 and 6 in figure 3.4. Source numbers 1, 3, 4, 7, 8 and 12 were shown to be not significantly different from the baseline intensity and were included into it. Sources 2, 9, 10 and 11 were significantly more reliable sources than the baseline sources although it must be stressed that censoring information was not included for sources which did not fail after the sources went into service use after 110 hours. This conclusion is the same as for the PHM analysis of chapter 6.3.

Figures 7.6 and 7.7 are an alternative representation of figure 3.4 and show that the more reliable sources are number 2 in plot 7.6 and numbers 3, 4 and 5 in figure 7.7 which represent the sources 9, 10 and 11. The vertical separation of the sources in the two plots are showing that the assumption of proportional intensities (and hence proportional hazards) is reasonable.

FIGURE 7.6 PLOT OF THE FIRST SIX SOURCES: FAILURE NUMBER AGAINST TIME IN DAYS

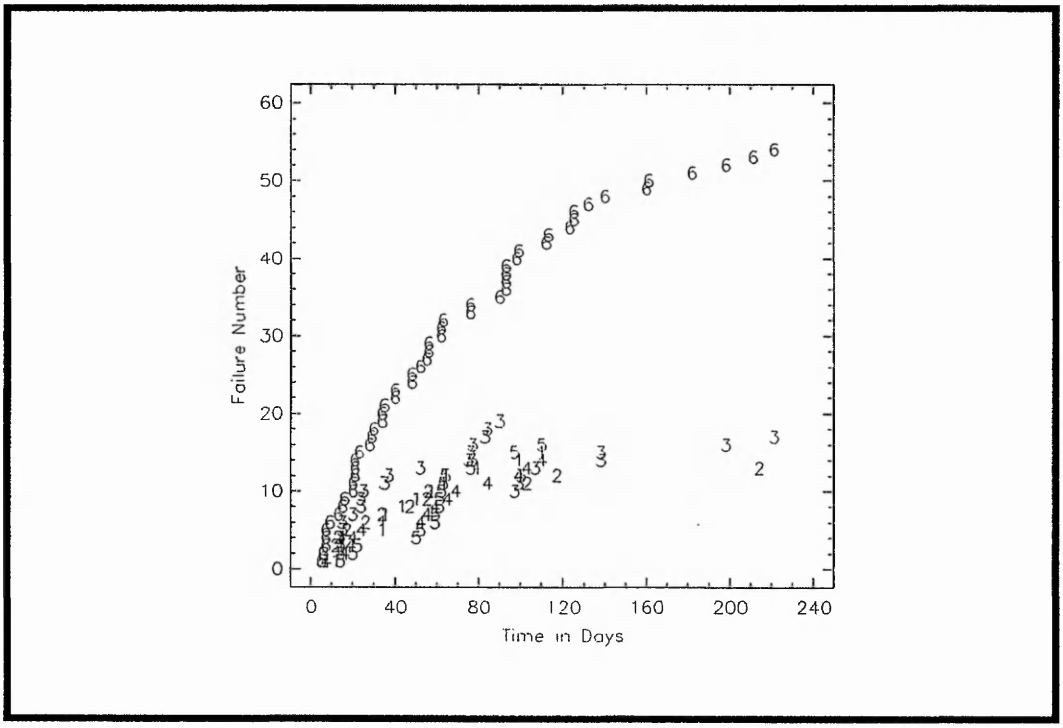
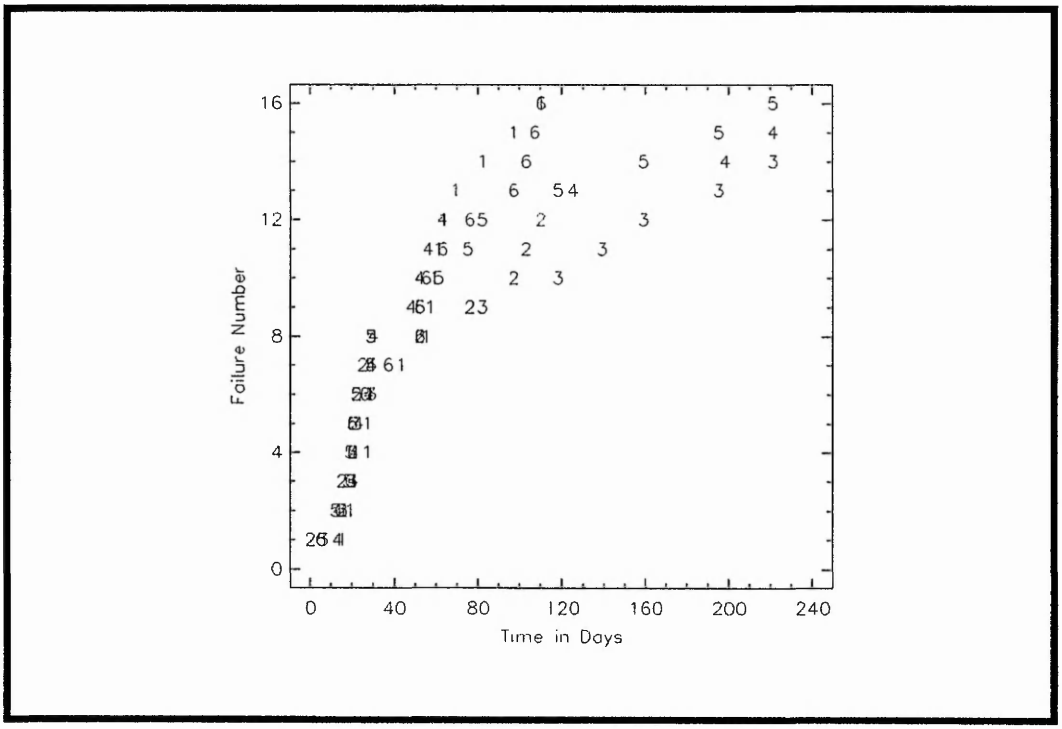


FIGURE 7.7 PLOT OF THE LAST SIX SOURCES: FAILURE NUMBER AGAINST TIME IN DAYS

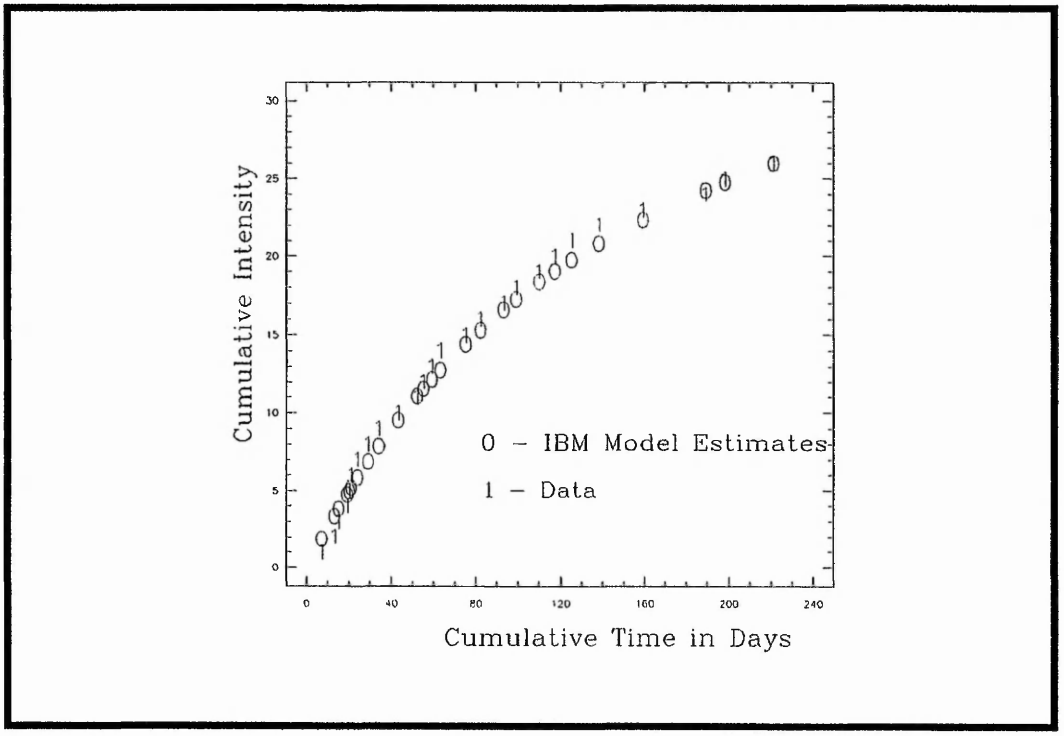


7.6.5 ANALYSIS OF THE BASELINE INTENSITY

The two aspects of a software reliability model which provide useful information; the reliability in a future period and the final system reliability are discussed here. The IBM model is such that the reliability tends towards zero as time increases indefinitely, the expected number of failures will tend to infinity and the intensity will tend to a constant, not necessarily zero. Therefore, the analysis of the intensity and the mean value function are analysed in a time frame close to the end of the project where the IBM model assumptions are probably still

valid. A plot of the cumulative intensity against time with a superimposed plot of the IBM model (Rosner (1961)) is shown in figure 7.8.

FIGURE 7.8. PLOT OF OBSERVED AND EXPECTED CUMULATIVE INTENSITY AGAINST TIME



In the Ascher formulation, the intensity is given by

$$\lambda(t) = 0.03428 + 0.2412 * 0.98786^{t}$$

It can be seen that the intensity will eventually tend towards 0.034 and at the end of the project of 220 days, it had reduced from 0.2755 to 0.0507. This constant value of the final intensity shows that not all failures have

or will be removed according to the model. The baseline cumulative intensity or baseline expected number of failures in time t is given by

$$E(N(t)) = 0.03428t - 19.7547*(0.98786^t - 1).$$

The expected number of failures for the four significantly more reliable sources is determined by multiplying the baseline cumulative intensity by each exponential covariate term $e^{(\beta x)}$.

The constant term and the decaying term for the IBM model are listed separately in the table of mean values below.

TABLE 7.5. TABLE OF MEAN VALUES $E(N(t))$

Time Period	Cum. Number of Failures (Constant term -Ct)	Diff. from Next Period Value (Ct)	Cumulative Number of Failures (Exponential term -Et)	Difference from Next Period Value (Et)
110	3.770	1.029	14.598	1.581
140	4.799	1.028	16.179	1.097
170	5.827	1.028	17.276	0.760
200	6.855	0.686	18.036	0.373
220	7.541		18.409	

Assuming the IBM baseline intensity is valid beyond 220 days and that the variation of the data about the model

remains constant, the number of failures per source will tend towards approximately one every thirty days as the exponential term of the IBM model will eventually die out.

Prediction intervals for the non-homogeneous Poisson process have been discussed in Engelhardt and Bain (1978) and Calabria, Guida and Pulcini (1990) but only for the Duane model. Assuming the IBM model is valid beyond the data with constant variation and expressing the baseline intensity as $\lambda(t) - 0.03428 = 0.2412 * 0.98786^{t}$, tolerances may be selected for the $\lambda(t) - 0.03428$ difference term and the time to reach a specific intensity may be estimated as $t = \frac{\ln\left(\frac{(\lambda(t) - 0.03428)}{0.2412}\right)}{\ln 0.98786}$. The results are presented in table 7.6 below.

TABLE 7.6. TABLE OF TIMES TO REACH A GIVEN INTENSITY

Tolerance : Difference from 0.03428	Time to reach intensity in days
10^{-1}	72.084
10^{-2}	260.599
10^{-3}	449.115

The results above show that the IBM model fits the data graphically quite well (see figure 7.5) and provides estimates for the final intensity (reliability) and the time to reach a specific intensity assuming that the IBM model is valid for prediction.

7.6.6 FURTHER PROPORTIONAL INTENSITY MODELLING OF THE TWELVE SOURCES

The data on the twelve sources above also included explanatory information regarding the type of source and type of use (test or test/live). Both of these covariates were found to be significant indicating that the intensity is different for different types of source and that those sources which failed in the test and live phase were significantly less reliable than those sources which only failed in the test phase. The results on the source types support the conclusions of the generalised linear modelling of chapter 7.4. The baseline intensity was plotted against time and was found to have a good IBM model fit.

In conclusion, the proportional intensity framework models the cumulative time to failure with covariates and can take complex structure (such as curvature) into account with an appropriate intensity specification.

A number of multivariate techniques have been applied to Alvey data set number 3 with varying degrees of success and these methods may be viewed as useful tools for highlighting and reducing structure within complex data sets.

8 OTHER WORK

During the period of registration, related work peripherally with software reliability data analysis was carried out. The approaches to the analysis of hardware reliability data in chapter 8.1 are similar to those described in chapter 1 of this thesis. Chapter 8.2 details my comments on a recent reliability paper by Ansell and Phillips.

8.1 COMMENTS ON THE EUREDATA BENCHMARK EXERCISE

Five groups of members of Euredata (a European association of industrial and academic data bank operators and analysts) undertook an analysis of one databank of valve data. The reasons for this were to compare the methods available and highlight areas of research into data bank analysis.

Each analysis undertaken by the participants created by various means a homogeneous data set of valves. This homogeneity classified components according to their physical characteristics. It is usually assumed that physical homogeneity is the same as statistical homogeneity however an analysis of the SRS databank (including pump and valve data) (Walls and Bendell (1985)) showed that 34 per cent of the data sets exhibited trend and/or serial correlation. By adopting the approach of Walls and Bendell supplemented by the approach of Ansell and Phillips (1989), each analysis and assumptions of each analysis may be listed for each participant, (See tables 8.1 and 8.2).

The analysis of Interatom showed that there was trend in the data (between the first and second failure times) however NUKEM determined that there was no trend based on

the results of a failure intensity analysis and an ad hoc rule for rejection of the trend. JRC-ISPRA showed there was no trend under the assumption of the binomial-beta model.

Confirmation of trend under other assumptions may be considered. The Laplace test (Cox and Lewis (1966)) may be used to see if other types of trend exist. The use of proportional hazards modelling with a covariate to describe the difference between the first and subsequent failure times would also allow reliability to be estimated, (Walker (1989)). The Bayesian time series approach of Davies, Naylor and McCollin (1989) may also be adopted.

JRC-ISPRA showed that certain components did not come from the same population whereas all the other participants assumed statistical identicality was the same as physical identicality (although VTT required this assumption for their availability analysis).

The application of time series to search for serial correlation and cyclic trend and the application of branching processes (where a series of primary events generates subsidiary series of events), (Cox and Lewis (1966), Ascher and Feingold (1984)), are frequently overlooked in reliability analysis. Further work and application is required in both these areas for the techniques to become widespread.

The problems of outliers, multiple events and censoring is discussed in Walls and Bendell (1985). A solution to the problems of multiple events and censoring is to perform the analysis under various assumptions. For multiple failures, we may assume extra failures are secondary or

a quirk of the repair procedures (e.g; block preventive maintenance after a single component failure) and ignore them from the analysis. Alternatively we may include them as separate events. The results of each analysis may then be compared to determine the effect of the assumptions on the analysis.

Determination of outliers for various statistical tests is also described in Walls and Bendell but this area still requires further work.

TABLE 8.1 COMPARISON OF ANALYSES FOR THE EUREDATA BENCHMARK EXERCISE
DEFINITIONS OF POPULATION AND FAILURE

COMPANY	NUKEM	VTT	JRC-ISPRA	INTER-ATOM	ENEA-VEL
Grouping of parts	Physical	Physical	Physical SF	Physical SA	Physical CA
Failures analysed (with classes)	AO AS AF	AO AS AI repairs	AO AS AD	AS	Wear Fatigue Erosion Design OE CO
Time metric	OT	OT	OT	OT	OT

TABLE 8.2 COMPARISON OF ANALYSES FOR THE EUREDATA BENCHMARK EXERCISE
QUANTITATIVE ANALYSIS

COMPANY	NUKEM	VTT	JRC-ISPRA	INTER-ATOM	ENEA-VEL
EDA	M graph		TT Kaplan Meier plot	TT	
Trend test	No trend found	NT	No trend found GF	Trend found - CF	TD
Serial Correlation	NT	NT	Log rank WR LR	NT	NT
Distribution	EX WE GF	EX WE GF	See EDA above OS GF	EX WE	EX WE
Assumptions :	OU CE	OU CE Trend SC	OU CE	OU CE	OU
Other Analysis		AV			

Symbology for tables above :-

SF - Similarity of function
SA - Similarity of application
CA - Correspondence analysis
AO - All operational
AS - All sudden
AF - All sudden with function loss
AI - All incipient
AD - All complete on demand, All on demand
OT - Operating time
CO - Corrosion
OE - Operator error
GF - Goodness of fit test
TD - Trend test discussed
EX - Exponential
TT - Total time on test plot
WR - Wilcoxon rank test
LR - Likelihood ratio test
NT - No test carried out
WE - Weibull
OS - Outliers
OU - Operational use
CE - Censorings
SC - Serial Correlation
AV - Availability
CF - Comparison of first and second failure times

In conclusion, the approach to data analysis by the five groups of Euredata members were similar and incorporated searching for trend, serial correlation and structure within the data as described within this thesis. Each approach to analysis are subsets of the generalised approach to data analysis outlined in chapter 1.

8.2 COMMENTS ON THE ANSELL AND PHILLIPS 1989 PAPER

Ansell and Phillips presented a paper on the 'Practical problems in the statistical analysis of reliability data (with discussion)' to the Royal Statistical Society in 1989 and I was asked to comment on the paper which subsequently appeared in the Journal of the RSS, series C.

My comments were as follows:

"I will comment on two points mentioned in page 2 of the paper. It was quoted that "There are many disincentives to presenting data sets in this field often because of commercial considerations."

Two examples of databased reliability information are mentioned below with reasons why data has been unavailable for statistical analysis.

NASA performed a prediction using data based information in the 1960's to estimate the probability of returning from the moon. The calculated probability was so small that three recommendations were suggested:

(a) to improve the database figures by increased component testing;
(b) not to go to the moon;
(c) to scrap predictions altogether.

The last alternative was taken.

In the gas and oil industries, it may cost up to one million pounds to install and maintain a database for

certification purposes and so companies are very wary of requests for data in case it is used by other companies either for their own certification purposes or to discredit the data collector.

The second point concerns the quote "As the objective of any reliability study is reliability assessment, estimation and prediction....". One main objective of a reliability study not mentioned above is how to improve reliability by reducing design faults. An example of how to reduce reliability problems by using statistics is given below.

Companies which manufacture large electronic/electrical systems, e.g., radar, use the U.S. military standard MIL-HDBK-217 to predict in service failure rates. Point estimates calculated by this method are then compared with specification requirements. Reliability in practice is achieved by test, fail and redesign. The prediction method does not really show future reliability, however the statistics (the Aarhenius model) and the quality factors incorporated in the failure rate equations in the military handbook show relationships between reliability and the component factors which affect reliability. The reliability engineer chooses components so that these factors are minimised and thus increases reliability at the design stage by good practices.

Proportional hazards modelling may be used as a diagnostic tool for determining the factors which affect component reliability and thus reliability engineers will create new design methodologies based on the results."

The first point describes why data is not readily obtainable from companies. The use of the techniques outlined in this thesis may supply an incentive to companies to supply more data as the analyses presented here highlight the problems of data analysis which are likely to occur within any data collection.

The second point outlines the use of statistical reliability models incorporating a structural term as an important tool to aid the improvement of reliability at the design stage of a product.

9 CONCLUSIONS

9.1 CONTRIBUTIONS TO KNOWLEDGE AND REVIEW OF THESIS

In chapter 1, an approach to the analysis of software and hardware failure data sets is described. This thesis covers the use of the approach to one specific data set.

The use of time series to model a data collection has been carried out by the Box-Jenkins models. This is the first reported approach of forecasting in a software context to determine when the software should be fault-free.

Comparative analysis of this technique has been undertaken in chapter 6 using proportional hazards modelling. PHM supplemented the time series approach by confirming the time series structure and allowed more detail to the structure to be modelled, i.e. day of failure.

It was shown in chapter 5 that most of the well-known software reliability models are special cases of proportional hazard models with either an extreme value (Gumbel) or an exponential hazard rate.

A coherent approach to software and hardware reliability growth modelling is described in chapter 6. This procedure using PHM highlights which are the appropriate NHPP's to model a specific data set. The advantage of this investigative PHM approach of software reliability modelling over other statistical approaches is that models cannot be mis-specified. The procedure has been applied to a software failure data set.

In chapter 7, various multivariate methods were applied for the first reported time to software failure data to determine possible structure and determining possible missing information.

The relationship between generalised linear modelling and proportional intensity modelling has been explored. The proportional intensity modelling of the software data is the first application of PIM to software failure data.

9.2 TYPES OF ANALYSES UNDERTAKEN

Table 9.1 below shows the analyses which have been undertaken in this thesis and the analyses these may be compared with. All of the analyses apart from proportional intensity modelling were carried out by using the commercial software packages of MINITAB or GLIM with some of the plots drawn with the Statgraphics statistical package. The proportional hazards modelling was carried out with an updated version of the software available in Kalbfleisch and Prentice (1980). The hazard plots were plotted using MINITAB.

MINITAB was the easiest package to use for plotting however the graphics were difficult to fit into the format of the wordprocessing package. The Statgraphics package required additional data editing via the Freelance package for a reasonable graphics output.

Each of the comparable analyses give similar conclusions, the easiest to use and understand being EDA. EDA should be carried out as a prerequisite for any further statistical modelling.

The analysis of the twelve least reliable sources in Alvey data set number 3 by EDA shows that an IBM model may be appropriate for the baseline intensity. This may be verified by proportional intensity modelling but not proportional hazards modelling. The covariate 'type of use' may have modelled the curvilinearity which the IBM model shows however it was found that this covariate was collinear with the time metric. The problems of multi-collinearity and monotonicity found in PHM analyses is useful in determining if relationships between covariate values exist and multivariate techniques such as PIM, GLIM, etc may then be used to model the highlighted structure.

Analysis of waiting times to failure of the twelve least reliable sources in Alvey data set number 3 with PHM showed which of the well known reliability models fitted the data. It was found that the less complex the model specification, the more likely it was to fit the data.

Modelling a time metric against the software attributes was carried out by EDA, PHM and multivariate techniques. These were attempted to show some relationship between time metric and the software attributes without much success. However the conclusions were to be expected given the knowledge of the software product. The techniques are still very powerful and may provide useful insight into the structure of future data analyses.

TABLE 9.1 TABLE OF COMPARISON OF METHODS

Analysis Method	Dependent Variable	Independent Variable	Comparison with :	Chapter
1. EDA	Number of failures per day	Days	6, 7, 8, 9	3
2. EDA	Cumulative Time to Failure	Source number for the twelve least reliable sources	11, 17	3
3. EDA	Proportion of Failures per source type	Time to Failure	15	3
4. EDA	Cumulative Time to Failure	Failure number, Log of CTTF, Log of Failure number for the twelve least reliable sources	10	3
5. EDA	Number of Faults	Source Size, Type, Language	12, 13, 14	3

TABLE 9.1 (CONTINUED) TABLE OF COMPARISON OF METHODS

6. EDA (Box and whisker plot)	Number of Failures	Ten day periods	7	4
7. Box - Jenkins Time Series	Failure Count	Trend, Moving Average, Seasonality	6	4
8. Box - Jenkins Time Series	Log Failure Count	Trend, Moving Average, Seasonality	9	4
9. Continuous PHM	Failure Count	Cumulative Failure Count, Day of Week, Previous Day number of Failures	8	6
10. Continuous PHM	Time Since Last Failure	Failure number, CTTF, Log failure number for the twelve least reliable sources	4	6

TABLE 9.1 (CONTINUED) TABLE OF COMPARISON OF METHODS

11. Continuous PHM	Time Since Last Failure	Source number for the twelve least reliable sources	2,17	6
12. Discriminant Analysis	Number of Faults	Source number, type, language, size, programmer, version, first and final appearance, time	5	7
13. Principal Components Analysis	as above	as above	5	7
14. Log linear modelling	Number of Faults	Source type, language, size	5	7
15. GLIM Logit, Probit and Complementary Log-Log Links	Proportion of Failures	Log Cumulative Time to Failure	3	7

TABLE 9.1 (CONTINUED) TABLE OF COMPARISON OF METHODS

16. GLIM Logit Link	Cumulative Proportion of Failures	Log CTTF, Source type	Tends to Log Link model as $n \to \infty$ p small	7
17. Proportional Intensity Modelling	Cumulative Time to Failure	Source Designation	Like GLIM Log Link, 2, 11	7

9.3 FURTHER WORK

The following are some analyses which may be applied to the data available from a software data collection scheme.

Modelling with PIM with an IBM intensity has been carried out in this thesis. Other suitable intensities for the baseline within PIM may be the Littlewood or the Weibull NHPP.

In the time series analysis of the number of failures per day, a reduction in the error variance may be possible by fitting covariates (such as staff available, type of use, etc) within the model. The software package BATS produced by and available from Warwick University allows this.

Analysis of the types of source which fail in the factory and those which fail with the customer may be modelled with a multivariate model.

An analysis of the repairs and times to repair would show the number of amendments outstanding, the workload per programmer and the range of times it takes to repair a source. Also, the repair time gives a measure of severity of the failure which may be used as explanatory information in multivariate analyses. The number of repairs per fault may also be regarded as a measure of the severity of a fault and hence may be useful as a covariate in proportional hazards modelling.

Stratification of data for time to first failure, time to second failure, etc as applied by Walker (1989) may be useful in determining the effect of the repair of the first failures of sources. Analysis of only the first instance of failures of sources may highlight the rate at which unfailed sources may fail.

In conclusion, the data obtainable from a software data collection scheme provides a wealth of information which may be analysed in a systematic way to provide insight into final system reliability and data structures.

REFERENCES

Aitken, M., Anderson, D., Francis, B. and Hinde, J. (1989). Statistical Modelling in GLIM. Oxford Science Publications. pp 240-241.

Anderson, B. (1986). The Role of Entropy and Information in Software Reliability. BAe ST30581.

Anderson, P.K. and Gill, R.D. (1982). Cox's Regression model for counting processes a large sample study. Annals of Statistics, 10. pp 1100-1120.

Anderson, T.W. and Darling, D.A. (1952). Asymptotic Theory of Certain "Goodness of Fit" Criteria based on Stochastic Processes. Annals of Mathematics Vol 23. pp 193-212.

Anderson, T. and Randell, B. Computing Systems Reliability An Advanced Course. Cambridge University Press 1979. Chapter 9.

Ansell, J.I. and Phillips, M.J. (1989). Practical problems in the statistical analysis of reliability data (with discussion). Journal of the RSS, series C Vol 38, No. 2. pp 205-247.

Ascher, H.E. (1968). Evaluation of Repairable System Reliability using the 'Bad-As-Old' Concept. IEEE Transactions on Reliability. R-17, pp 103-110.

Ascher, H.E. and Feingold, H. (1984). Repairable Systems Reliability Modelling, Inference, Misconceptions and their Causes. Marcel Dekker.

Baker, R.J. and Nelder, J.A. (1978). The GLIM System. Release 3.77 Generalized Linear Interactive Modelling Manual. Oxford, U.K.: National Algorithms Group.

Basili, V.R. and Selby Jr, R.W. (1984). Data Collection and Analysis in Software Research and Management. Proceedings of the American Statistical Association and Biometry Society Joint Statistical Meetings. Philadelphia.

Bastos Martini, M.R., Kanoun, K. and Moreira de Souza, J. (1990). Software-Reliability Evaluation of the TROPICO-R Switching System. IEEE Transactions on Reliability, Vol 39, No. 3.

Bastos Martini, M.R. and Moreira de Souza, J. (1991). Reliability Assessment of Computer Systems Design. Microelectron. Reliab., Vol 31, No 2/3. pp 237-244.

Bazovsky, I. (1961). Reliability Theory and Practice. Prentice Hall. Englewood Cliffs, N. J.

Bendell, A. and Walls, L.A. (1985). Exploring Reliability Data. Quality and Reliability Engineering International. Vol 1. pp 37-51.

Bendell, A. (1988). An Overview of Collection, Analysis, and Application of Reliability Data in the Process Industries. IEEE Transactions on Reliability, Vol 37, No. 2. pp 132-137.

Bendell, A., McCollin, C., Wightman, D.W., Linkman, S. and Carn, R. (1988). Software Reliability Data Collection, Problems and Possibilities. Proceedings of the 20th Euredata Conference, Siena.

Bishop, Y.M, Fienberg, S.E. and Holland. P.E., (1975) Discrete Multivariate Analysis. MIT Press.

Box, G.E.P and Jenkins, G.M. (1976). Time Series Analysis: Forecasting and Control. Holden-Day, London.

Box, G.E.P. and Pierce, D.A. (1970). Distribution of residual correlations in autoregressive-integrated moving average time series models. J. Am. Statist. Assoc. 65. pp 1509-26.

Breslow, N.E. (1974). Covariance analysis of censored survival data. Biometrics 30. pp 89-99.

Brocklehurst, S. (1987). On the Effectiveness of Adaptive Software Reliability Modelling. City University report. (for Alvey SRM project).

BS 5750. (1987). British Standard for Quality Systems.

Cain, K.C. and Lange, N.T. (1984). Approximate Case Influence for the Proportional Hazards Regression Model with Censored Data. Biometrics, 40. pp 493-499.

Calabria, R., Guida, M. and Pulcini, G. (1990). Point Estimation of Future Failure Times of a Repairable System. Reliability Engineering and System Safety, 28. pp 23-34.

Chatfield, C. and Collins, A.V. (1980), Introduction to Multivariate Analysis. Chapman Hall.

Conte, S.D., Dunsmore, H.E. and Shen, V.Y. (1986). Software Engineering Metrics and Models. Benjamin/Cummins. pp 3-7.

Cox, D.R. and Lewis, P.A.W. (1966). The Statistical Analysis of Series of Events. London: Methuen.

Cox, D.R. (1972a). The Statistical Analysis of Dependencies in Point Processes. Stochastic Point Processes. ed P.A.W. Lewis, Wiley, New York pp 55-66.

Cox, D.R. and Snell, E.J. (1968). A General Definition of Residuals. J. R. Stat. Soc. B 30. pp 248-275.

Cox, D.R. (1972). Regression Models and Life-tables (with discussion). J. R. Statistic. Soc., (B), 34. pp 187-220.

Cox, D.R. (1975). Partial Likelihood. Biometrika. 62. pp 269-276.

Cox, D.R. and Oakes, D. (1984). Analysis of Survival Data. Chapman and Hall, London.

Cozzolino, J.M. (1968). Probabilistic Models of Decreasing Failure Rate Processes. Naval Research Logistics Quarterly. Vol. 15. pp 361-374.

Crow, L. (1975). On Tracking Reliability Growth. Proc. Twentieth Conference on the Design of Experiments. ARO Report 75-2, pp 741-754.

Crow, L.H. and Singpurwalla, N.D. (1984). An Empirically Developed Fourier Series Model for describing Software Failures. IEEE Transactions on Reliability Vol 33. No. 2.

Crowder, M.J., Kimber, A.C., Smith, R.L. and Sweeting, T.J. (1991). Statistical Analysis of Reliability Data. Chapman Hall. pp 73-74.

Csenki, A. (1989). Bayesian Formulations of the Jelinski-Moranda Software Reliability Model. City University report. (For Alvey SRM project).

Dale, C. (1989). Software Safety Certification in Potentially Hazardous Industries. Submitted as a deliverable to the Alvey project.

Dale, C. (1991). The Assessment of Software Reliability. Reliability Engineering and System Safety 34. pp 91-103.

Davies, N., Marriott, J.M., Wightman, D.W. and Bendell, A. (1987). The Musa Data Revisited: Alternative Methods and Structure in Software Reliability Modelling and Analysis. Achieving Safety and Reliability with Computer Systems. Proceedings of the Safety and Reliability Symposium. Edited by B.K. Daniels. Elsevier Applied Science. pp 118-130.

Davies, N., Naylor, J.C. and McCollin, C. (1989). Bayesian and time series modelling techniques in transportation reliability. Proceedings of the Safety and Reliability Society Symposium. Reliability on the Move. Elsevier press. Bath. pp 260-268.

DEF-STAN-0055 The Procurement of Safety Critical Software in Defence Equipment. Ministry of Defence. 1991.

Disney, J.D., McCollin, C. and Bendell, A. (1990). Taguchi Methodology within Mechatronics. Proceedings of the International Conference on Mechatronics Designing Intelligent Machines. Cambridge.

Duane, J.T. (1964). Learning Curve Approach to Reliability Monitoring. IEEE Transactions. Aerospace. Vol 2.

Engelhardt, M. and Bain, L.J. (1978). Prediction Intervals for the Weibull Process. Technometrics. Vol.20. No. 2. pp 167-169.

Everitt, B.S., (1977), The Analysis of Contingency Tables. Chapman Hall.

Fagan, M.E. (1976). Design and Code Inspections to Reduce Errors in Program Development. IBM Systems Journal 15(3). pp 182-211.

Fleming, T.R. and Harrington, D.P. (1991). Counting Processes and Survival Analysis. Wiley.

Follman, D.A. (1990). Modelling Failures of Intermittently Used Machines. Applied Statistics. JRSS(C). 39. No. 1. pp. 115-123.

Font, V. (1985). Une Approche de la Fiabilite des Logiciels: Modeles Classiques et Modeles Lineaire Generalise. Thesis L'Universite Paul Sabatier de Toulouse, France.

Fry, T.C. (1965). Probability and its Engineering Uses, 2nd ed. Princeton N.J. Van Nostrand, pp 24-248.

Gamerman, D. and West, M. (1988). Non-proportionality of hazards. A time series application in employment studies. Internal report, University of Warwick.

Gaver, D.P. and Avar, M. (1979). Analytical hazard representations for use in reliability, mortality and simulation studies. Commun. Statist. 8. pp 91-111.

Goel, A.L. and Okumoto, K. (1979). Time-Dependent Error Detection Rate Model for Software Reliability and other Performance Measures. IEEE Transactions on Reliability. R-28(3). pp. 206-211.

Goldstein, M. and Dillon, W.R. (1978). Discrete Discriminant Analysis. Wiley.

Gray, C.T. (1986). A Framework for Software Reliability. Pergamon Infotech State of the Art Report. Pergamon Infotech, UK pp 81-94 and 249-250.

Gross, A.J. and Clark, V.A. (1975). Survival Distributions: Reliability Applications in the Biomedical Sciences. Wiley, New York.

Gumbel, E.J. (1958). Statistics of Extremes. Columbia Univ. Press, New York.

Harrison, P.J. and Stevens, C.F. (1976). Bayesian Forecasting (with discussion). J.R. Statistic Soc. B38. pp 205-247.

Hartler, G. (1989). The Nonhomogeneous Poisson Process - A Model for the Reliability of Complex Repairable Systems. Microelectron. Reliability. Vol. 29. No. 3. pp 381-386.

Harvey, A.C. (1989). Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge University Press.

Harvey, A.C. and Fernandes, C. (1988). Time Series Models for Count or Qualitative Observations. London School of Economics internal report.

Hill, M.O. (1974). Correspondence Analysis: A Neglected Multivariate Method. Applied Statistics Vol 23 No. 3. pp 340-354.

Holden, R.T. (1987). Time series analysis of a contagious process. J.A.S.A. Vol 82, No. 400. pp 1019-1026.

Horigome, M., Singpurwalla, N.D. and Soyer, R. (1984). A Bayes Empirical Bayes Approach for (Software) Reliability Growth. Computer Science and Statistics: 16th Symposium on the Interface Proc. Atlanta, GA; North-Holland, Amsterdam.

Hufton, D.R. and Exley, J. (1989). Some Applications of Generalised Linear Models to Software Reliability. Report written for Alvey task area 8.

Hufton, D.R., Quinn, P. and McInnes, A. (1989). Analysis of Data set B1 using Generalised Linear Models. Report written for the REQUEST Project.

Hufton, D.R. (1989). The Application of Generalised Linear Models to Software Reliability. Report submitted to Alvey task 9 participants.

Hufton, D.R. (1989). Final Management Report on the Alvey SRM Project (Ref SE/072). Produced for the IED Directorate in confidence.

Jaaskelainen, P. (1982) Reliability Growth and Duane learning curves. IEEE Trans. Reliability 31. 151-154.

Jelinski, Z. and Moranda, P.M. (1972). Software Reliability Research (W. Freiberger, Editor) Statistical Computer Performance Evaluation. Academic, New York, pp. 465-484.

Jordan, C.W. (1975). Life Contingencies. The Society of Actuaries. Chicago.

Kalbfleisch, J.D. and Prentice, R.L. (1973). Marginal Likelihoods based on Cox's Regression and Life Model. Biometrika, 60. pp 267-278.

Kalbfleisch, J.D. and Prentice, R.L. (1980). The Statistical Analysis of Failure Time Data. John Wiley, New York.

Kapur, P.K. and Garg, R.B. (1991). Optimum Release Policy for an Inflection S-Shaped Software Reliability Growth Model. Microelectron. Reliab. Vol 31. No. 1 pp 39-41.

Keiller, P.A. and Miller, D.R. (1991). On the Use and the Performance of Software Reliability Growth Models. Reliability Engineering and System Safety. Vol. 32. pp 95-117.

Kitchenham, B.A. (1984). Program history records: a system of software data collection and analysis. ICL Technical Journal. pp 103-113.

Kleinbaum, D.G. Kupper, L.L. and Chambless, L.E. (1982). Logistic Regression Analysis of Epidemiologic Data: Theory and Practice. Communications in Statistics.- Theory and Methods., 11(5). pp 485-547

Kremer, W. (1983). Birth-Death and Bug Counting. IEEE Transactions on Reliability. Vol R-32, No. 1.

Landers, T.L. and Kolarik, W.J. (1986). Proportional Hazards Models and MIL-HDBK-217. Microelectronics and Reliability Vol. 26 No. 4.

Langberg, N. and Singpurwalla, N.D. (1985). A Unification of Some Software Reliability Models. SIAM Journal on Scientific and Statistical Computing. Vol. 6.

Lawless, J.F. (1982). Statistical Models and Methods for Lifetime Data. John Wiley, New York.

Lawless, J.F. (1987) Regression Methods for Poisson Process Data. Journal of the American Statistical Association. Vol. 82, No. 399. pp 808-815

Lee, L. (1980). Comparing Rates of Several Independent Weibull Processes. Technometrics. Vol. 22. No. 3. pp 427-430.

Lewis, P.A.W. and Schedler, G.S. (1976). Simulation of Nonhomogeneous Poisson Processes with Log Linear Rate Function. Biometrika Vol 63 No. 3. pp 501-5.

Littlewood, B. and Verrall, J.L. (1973). A Bayesian Reliability Growth Model for Computer Software. J.R.S.S. Series C (Applied Statistics) Vol. 22 No. 3. pp 332-346.

Littlewood, B. (1981). Stochastic Reliability Growth: A Model for Fault-removal in Computer Programs and Hardware Designs. IEEE Transactions. Reliability R-30. pp 313-20.

Littlewood, B. (1984). Rationale for a Modified Duane Model. IEEE Transactions on Reliability. Vol. R-33 No. 2.

Littlewood, B. and Sofer, A. (1987). A Bayesian Modification to the Jelinski-Moranda Software Reliability Growth Model. Software Engineering Journal. pp 30-39.

Lloyd, D.K. and Lipow, M. (1977). Reliability: Management, Methods, and Mathematics. 2nd Edition. Published by the authors.

Lloyd, D.A., Staley, J.E. and Sutcliffe, P.S. (1977). A Theoretical Study of Methods for analysing Reliability Growth. MOD (PE) Report No. GR/77/06.

Longbottom, R. (1980). Computer System Reliability. Wiley pp 83-85.

Madiedo, E. (1986). The Use of Time Series in the Reliability Field. Reliability Technology-Theory and Applications. Elsevier Science Publishers B.V. North-Holland.

McCollin, C. (1980). Statistical Test Plans for Reliability Growth. M.Sc thesis. University of Birmingham.

McCollin, C., Bendell, A. and Wightman D.W. (1989). Effects of Explanatory Factors on Software Reliability. Proceedings of Reliability 1989 Vol 2. pp 5Ba/1/1-11.

McCollin, C. (1990) Review of Task 9 Activities. Alvey Report number SRM/T9/RPT2/V0.1.

McCollin, C., Wightman, D.W., Dixon, P. and Davies, N. (1990). Some Results of the Alvey Software Reliability Modelling Project. Proceedings of the SARSS. Altrincham, 1990.

McCullagh, P. and Nelder, J.A. (1983). Generalized Linear Models. Chapman and Hall.

McKenzie, E. (1988). Discrete-variate time series models. Strathclyde University Report.

Meinhold, R.J. and Singpurwalla, N.D. (1983). Bayesian Analysis of a Commonly Used Model for Describing Software Failures. The Statistician. Vol. 32. pp 168-173.

Mellor, P. and Bendell, A (Editors). (1986). Software Reliability State of the Art Report. Pergamon Infotech Limited.

Mellor, P. (1986). Database Definition for Collection of Software Reliability Growth Data. ALV073/TSK2/TCU/DBA-SEDEF.

Mellor, P. (1987). Software Reliability Modelling: The State of the Art. Journal of Information and Software Technology. Vol. 29. No. 2. pp 81-98.

Mellor, P. (1987). Comments on Draft Database Structure Definition Submitted for Agreement by Task 9.

Miller, D.R. (1986). Exponential Order Statistic Models of Software Reliability Growth. IEEE Transactions on Software Engineering. Vol. SE-12, No. 1.

MINITAB Statistical Software. Statistics Department, The Pennsylvania State University USA.

Moranda, P.B. (1975). Predictions of Software Reliability during Debugging. Proceedings Annual Reliability and Maintainability Symposium. Washington DC. pp. 327-332.

Musa, J.D. (1975) A Theory of Software Reliability and its Application. IEEE Transactions on Software Engineering. SE-1(3). pp. 312-327.

Musa, J.D. (1980). Software Reliability Data Submitted to the DACS. Bell Telephone Laboratories.

Musa, J.D. and Okumoto, K. (1984). A Logarithmic Poisson Execution Time Model for Software Reliability Measurement. Proceedings of the 7th International Conference on Software Engineering. IEEE Computer Society Press, Washington DC. pp 230-8.

Musa, J.D., Iannino A. and Okumoto K. (1987). Software Reliability Measurement, Prediction, Application. McGraw-Hill.

Myers, G.J. (1976). Software Reliability Principles and Practices. John Wiley pp 34-37.

Nagel, P.M. and Skrivan, J.A. (1981). Software Reliability Repetitive Run Experimentation and Modelling. Rep BCS-40366. Boeing Computer Company NASA Report no. CR-165836.

Nelson, W. (1982). Applied Life Data Analysis. Wiley, New York.

O' Connor, P.D.T. (1982). Practical Reliability Engineering. John Wiley and Sons. 1982.

O' Connor, P.D.T. (1991). Statistics in Quality and Reliability. Lessons from the Past, and Future Opportunities. Reliability Engineering and System Safety Vol. 34. pp 23-33.

Ohba, M. (1984). Software Reliability Analysis Models. IBM. J. Res. Dev. 28 (4). pp 428-443.

Pankratz, A. (1983). Forecasting with Univariate Box-Jenkins Models: Concepts and Cases. Wiley.

Parzen, E. (1962). Stochastic Processes. Holden-Day. San Francisco. 1962.

Pettit, A.N. (1984). Proportional Odds Models for Survival Data and Estimates using Ranks. Appl. Statist. Vol 33. No. 2. pp 169-175.

Phelps, M. and MacCallum, F. (1989). Analysis of Data set Freda. Report for the SRM REQUEST project.

Potter, M. (1988). Software Reliability Modelling Database: Technical Manual.
Report for the Alvey Project.

Prentice, R.L, Williams, B.J and Peterson, A.V. (1981). On Regression Analysis of Multivariate failure time data. Biometrika Vol 68 No. 2. pp 373-379.

Raftery, A.E. (1988). Analysis of a Simple Debugging Model. Applied Statist. Vol. 37, No. 1. pp 12-22.

Reid, N. and Crepeau, H. (1985). Influence Functions for Proportional Hazards Regression. Biometrika, 72. pp 1-9.

Rook, P. (Editor) (1990). Software Reliability Handbook. Elsevier Science Publishers.

Rosner, N. (1961). System Analysis-Non-Linear Estimation Techniques. Proc. Seventh National Symp. on Reliability and Quality Control, Institute of Radio Engineers. (now IEEE).

Samanta, P.K., Levine, M.M., Teichmann, T. and Kato, W.Y. (1985). A Multivariate Statistical Study for Detection of Failure Trends and Patterns in the Analysis of Events at

Nuclear Power Plants. Proceedings: International Topical Meeting on Probabilistic Safety Methods and Applications. San Francisco.

Schneidewind, N.F. (1975). Analysis of Error Processes in Computer Software. Proceedings 1975 International Conference on Reliable Software. Los Angeles. pp. 337-346.

Schoenfeld, D. (1982). Partial Residuals for the Proportional Hazards Regression Model. Biometrika, 69. pp 239-241.

Shooman, M.L. (1972). Probabilistic Models for Software Reliability Prediction. (W. Freiburger, Editor) Statistical Computer Performance Evaluation. Academic, New York, pp. 485-502.

Selby, R.W. (1990). Empirically Based Analysis of Failures in Software Systems. IEEE Transactions on Reliability, Vol 39, No. 4. pp 444-454.

Simmonds, I. and Potter, M. (1989). The SRM Database User Manual. Alvey Deliverable Task 11 output.

Singpurwalla, N.D. and Soyer, R. (1985). Assessing Reliability Growth Using a Random Coefficient Autoregressive Process and its Ramifications. IEEE Transactions on Software Engineering. Vol. SE-11, No. 12.

Smith, S.A. and Oren, S.S. (1980). Reliability Growth of Repairable Systems. Naval Research Logistics Quarterly. Vol. 27. pp 539-547.

Statgraphics, Statistical Graphics System, Statistical Graphics Corporation.

Sukert, A. (1980). A Guidebook for Software Reliability Assessment. Proceedings of the Annual Reliability and Maintainability Symposium. pp 186-190.

Teichmann, T., Levine, M.M., Samanta, P.K. and Kato, W.Y. (1985). Statistical Cluster Analysis and Diagnosis of Nuclear System Level Performance. Proceedings: International Topical Meeting on Probabilistic Safety Methods and Applications. San Francisco.

Thompson Jr. W.R. (1981). On the Foundations of Reliability. Technometrics Vol 23, No 1. pp 1-13.

Thompson, W.A. (1988). Point Process Models with Applications to Safety and Reliability. Chapman and Hall.

TickIT: Guide to Software Quality Management System Construction and Certification using EN29001. Department of Trade and Industry.

Tsiatis, A.A. (1981). A Large Sample Study of Cox's Regression Model. Annals of Statistics. 9. pp. 93-108.

U.S. MIL-HDBK-189. (1981). Military Standard for Reliability Growth Plans. RADC.

U.S. MIL-HDBK-217. Reliability Prediction of Electronic Equipment. RADC.

Walker, E.V. (1989). Proportional Hazards Modelling for the Analysis of Reliability Field Data. M.Phil thesis. Nottingham Polytechnic.

Walls, L.A. and Bendell, A. (1985). The structure and exploration of reliability field data: what to look for and how to analyse it. Proceedings of the Reliability Symposium. pp 5B/5/1-18.

Walls, L.A. and Bendell, A. (1987). Time Series in Reliability. Reliability Engineering. Vol 18. pp 239-265.

Walston, C.E. and Felix, C.P. (1977). A Method of Programming Measurement and Estimation. IBM Systems Journal No. 1.

West, M. Harrison, J. and Pole, A. (1988). BATS: A User Guide. University of Warwick.

Whitehead, J. (1980). Fitting Cox's Regression Model to Survival Data using GLIM. Applied Statistics. 29. No. 3. pp 268-275.

Wightman, D. and Bendell, A. (1986). The Practical Application of Proportional Hazards Modelling. Reliability Engineering 15. pp 29-53.

Wightman, D.W. (1987). The Application of Proportional Hazards Modelling to Reliability Problems. Ph.D thesis Trent Polytechnic.

Wightman, D.W. (1987). Review of models/techniques which incorporate explanatory variables and may be applied to software failure data. Nottingham Polytechnic report. (for Alvey SRM project).

Wightman, D.W., McCollin, C. and Dixon, P. (1991). Recent Applications of Some Statistical Techniques to Software Reliability Data. Proceedings of the 1991 Reliability Conference. Reliability 91. Elsevier Science Publishers.

Wightman, D.W., McCollin. C. and Bendell. A., Proportional Hazards Modelling of an Alvey Software Reliability Data Set. Awaiting publication.

Wright, D.R. A Modified U-Plot applied to Failure Count Prediction. City University report. (Report for Alvey SRM project).

Yamada, S. Ohba, M. and Osaki, S. (1983). Reliability Growth Models for Hardware and Software Systems based on Nonhomogeneous Poisson Processes. Microelectron and Reliability. Vol. 23. pp 91-112.

Zeger, S.C. (1988). A regression model for time series of counts. Biometrika 75 4. pp 621-9.

APPENDIX 1

Bendell, A., McCollin, C., Wightman, D.W.,
Linkman, S. and Carn, R. (1988).
Software Reliability Data Collection,
Problems and Possibilities.
Proceedings of the 20th Euredata Conference, Siena.

SOFTWARE RELIABILITY DATA COLLECTION - PROBLEMS AND POSSIBILITIES

A BENDELL C McCOLLIN D W WIGHTMAN
Trent Polytechnic

S LINKMAN STC and
R CARN Reliability Consultants

SUMMARY

The paper identifies lessons learned from software reliability data collection in the Alvey Software Reliability Modelling Project. The lessons are both technical and organisational in nature.

1 BACKGROUND

The paper is concerned with documenting the lessons learned for software reliability data collection from the collection exercise undertaken as part of the Alvey Software Reliability Modelling (SRM) Project. This is a multi-task project consisting of a collaborative team of initially some 10 UK organisations. The membership of the team has fluctuated with time, but currently consists of the National Centre for Systems Reliability (UKAEA), British Aerospace, STC, Logica, Trent Polytechnic and City and Newcastle Universities. The project's aims are somewhat diverse covering, for example, improvement to current models, investigation of models with different underlying assumptions, incorporation of auxiliary information concerning the product, development and use into models, modelling of special systems and the conflict between testing for debugging and for reliability prediction.

All these areas share a requirement for data for parametrising and testing the various model formulations against the real world. Accordingly, from early in its conception, the project incorporated a data collection exercise as an integral part. It was anticipated that data would be made available primarily internally from members of the consortium, but also from external sources including other Alvey projects and possibly Esprit projects. The most relevant Alvey project was perhaps the Software Data Library and fortuitously its membership overlapped with that of the Alvey Software Reliability Modelling project.

2 EVOLUTION OF DEMAND FOR DATA

The first set of problems that the project experienced in relation to the requirement for data collection was associated with its delayed and staggered start. Procedural, participant and documentation problems implied that when the project did eventually receive funding to commence, participating organisations were only able to assign or recruit staff on very differing timescales. This consequential scheduling problem for the very integrated work of tasks and subtasks has remained a problem of the project. In particular, organisations that did not have staff in place early were unable or unwilling to assist in the early discussions to define the detail of the data requirement.

The task area with a work package including data collection was Task 9 led by Trent Polytechnic and including STC and Logica. Early meetings took place between the then Project Manager, the leader of Task 9 and the then data-provider members of the consortium. These were intended to establish likely data availability against the somewhat vague data requirement specifications of the data-users. In parallel, requests for more detailed data requirement specifications were made to data users in the various tasks and meetings took place with the potential users. On the basis of this a requirements/availability matrix was drawn-up to identify

likely areas of deficiency and this was used as the basis of litigation by the data-providers in looking for further (and typically more expensive to process) sources. It is, perhaps, of interest to note that even at this stage a potential mismatch between data providers and users became apparent. The more theoretical data-users requiring 'high quality' data based on execution or a 'proxy' that was typically not available but at the same time not requiring some of the richness of auxiliary information that was on offer from the data sources.

The task 9 group met regularly throughout this formation stage and has continued to do so, typically monthly. At an early stage it decided that it would be beneficial to have representatives of the data-user organisations present. These were accordingly invited in; some have attended regularly, some occasionally, some never.

The group established a procedure to agree the benefits and cost effectiveness of data-sets both internal and external to the project members.

3 DEVELOPMENT OF SRM DATA STRUCTURE

The purpose of the data-base was to hold the data collected by Task 9. The data-base would facilitate initial analysis within Task 9 and subsequent access and statistical analysis by the other tasks.

From the beginning it was recognised that all data would be in a computable form. Textual documents would not be included. Of the numeric data, failure and repair data would be common to all data-sets but the availability of product, management and other project data would vary from data-set to data-set.

A second major problem lay in the nature of the demand for data and the way it was to be used. Informed discussions with task leaders uncovered two very broad areas of interest: relating reliability data to the structures of the software and relating reliability data to the project and process of development. Later an interest in testing regions also emerged. Project and process information was perceived as being mainly in documentary form and not suitable for inclusion directly in the data-base structures.

The most obvious outcome of the discussions was that the data-base logical design needed to be flexible in order to take account of different data-sets. It also needed to take account of the changing demands from researchers as their ideas matured. To deal with this a number of relations in the data-base were intended to act as "nodes" or growth points to which further relations could be linked as required. For example "staff" could be developed to include information on project organisation.

The third major problem overlay in the different methods and units used to capture the data in different data-sets. This was inevitable since the data-sets were collected by different organisations for different purposes. Specifically in some data-sets execution time had been recorded whereas in others calendar time had been used. The unit relation was included in my attempt to control this variability and make the data-sets comparable.

Design of the data-base was developed in Mellor (1). However, the design of the SRM data-base was intended not to be a standard structure for all projects but to provide a framework within which existing data-sets could be stored and accessed. It was also intended that the framework could be extended to future projects in response to research needs. The data-base structure actually achieved can be considered as a meta-model of software project reliability data.

The gap between the data-base structure and each specific data-set was to be bridged by tailoring the meta-model to model the actual structure of information in the data-set. The model was then instantiated by creating a new version of the data-base management system. A spin off benefit of this approach allowed the greater than hoped variability between data-sets at the field level to be taken into account.

The main entities included in the data-base structure can be grouped into the following 4

Project related

Life cycle, staff, manager, user.
User document, project documents
Configuration management (versions, repair groups).

Product related

Product, program type.
Modules, source fits control path, data path, interface.
Installation.

Software failures

Failures, unit, fault.

Repair

Repair
Fault
Investigation

The entities, and their relationships, and the invariants used to validate the data are described in detail in the Technical Manual (2).

It is interesting to note that the data-base structure includes a class of specified one to many relations, the one to two and only two relations, symbolised in the structure diagram as a double bodied arrow. This occurs in

the relations describing interfaces, (path, control path, datapath, and interface). For example as defined in the data-base a control path always links precisely two modules, although in the case of recursion the source and target of control are the same module.

4 IDENTIFICATION OF SUITABLE DATA-SETS

The process of finding and clearing for release reliability data-sets, for use in collaborative projects is one which demands the skills of both detective and diplomat.

The detective skills come into play as data-sets which are suitable for reliability research are few and far between, as has been shown constantly by the investigations of this project as well as the Alvey funded Software Data Library project and the ESPRIT funded REQUEST project.

The problems are twofold, first the data-sets are at a minimum required to contain an ordered sequence of events with associated time metric. It is usually the latter that is unavailable or in unacceptable form. The second problem is that for use in comparison to other data-sets, to establish more broadly applicable results, one must also have a large amount of other explanatory and comparability data.

A company will only have a consistent and well supported policy on the collection of such data, if the organisation itself has perceived use for the data. The data collector is then dependent on finding individual groups within the organisation having an interest in or use for that data.

Typically one knows of such data from either personal contact, due to shared interest, or because such groups as do collect the data required access to some support group, or centre of excellence.

Assuming that one can track down the data-sets having the desired basic measures of failures and associated times, then the real detective work begins in the acquisition of the other data which is required. Such data must be assembled from a large number of individual sources, often stored on a variety of electronic and other media.

A typical list of problems to be overcome for a data-set to be useful are given in the list below;

In collecting data, especially reliability data, one often has to extract and join data from a large number of sources if it is to be more than a simple set of failures against time. Also to obtain valid data it is necessary to deal with a number of problems.

TYPOGRAPHICAL ERRORS

The data will invariably have some typographical errors, some proportion of which will have an impact on the analysis to be done. A knowledge of the area for which the data is being collected is usually necessary to correct these.

INCOMPLETENESS

In many cases the data collected may be incomplete. The impact of this will be dependent on the activities to be undertaken with the data. The solutions used will also be dependent on this. In some cases the analysis might continue if the amount of data lost is limited to some missing metric values.

NUMBERING PROBLEMS

In many cases, unless the system for collection is automated, one can finish up with various problems associated with the allocation of numbers to incidents, for example links to other problems which lead nowhere. * In order to recover this the organisation must be searched to find the people who can provide the missing links. *See section 7.

DUPLICATION OF REPORTS

It is quite often possible to get multiple reports of the same problem occurring either displaced in site or displaced in time. This might be due to multi-site working in one case or simply more than one person or group of people doing the testing. The key is to establish precedence, but also possibly to combine reports to a better description.

MISMATCHED REPORTS, MULTIPLE PROBLEMS ON ONE REPORT

Often, especially in the testing phase more than one problem will be raised on the same problem report. This may be because it appears as a single problem, but in fact has multiple causes. Alternatively it may summarise faults found in a given testing session. It will be necessary to manipulate this data.

EXECUTION TIME CAPTURE

If the data is for reliability purposes then it is necessary to establish execution time for each of the failures. However there are a number of ways in which this may be done. For example, one might know the running time on a daily basis and use this to convert the calendar date; or this might be only available by summing operation time over a number of operating machines in the field population. Depending on the options chosen this will require an effort in modifying the data. At most a major additional data collection exercise may be necessary and not be possible retrospectively.

QUALITY OF RECORDING

In some cases the data will be badly recorded. This may be due to time pressure, or laziness on the part of the recorder, for example by choosing the same value on a scale 1-5 scale of complexity.

This is often compounded by the fact that the people collecting data may not be under the control of the data collector.

OTHER PROBLEMS

Problems also exist with regard to the definition of measurement units and the difficulty of extracting certain metrics.

5 CLEARANCE

The problems of obtaining clearance for data to be released to researchers who are not members of the same organisation can be immense.

The experience of the authors has been that it only takes one incident of data getting into the wrong hands, or used for the wrong purposes for the chances of releasing data, even in a sanitised form, to drop drastically. The following sub-sections discuss some of the issues which relate to the different areas.

SECURITY

Should data be available related to certain systems, it might be used in planning a strategy to attack the system, for example via overload. To date, the authors are not aware of any published cases of computer systems attacked in this way, but examples are known of the equivalent attacks on radar systems in earlier days and also unpublished examples of frauds involving people taking advantage of the unreliability of the system.

ABUSE OF COMMERCIAL CONFIDENCE

Another area where suspicion might exist relates to the availability of the data to a competitor of the data supplier or the client from whoms' system the data was acquired.

An example of the problems in this area relates to a company, some of whose development failure information was published in a study. A competitor of that company then used these figures in a sales presentation. However they did not compare like with like, they quoted field failure information

against the design information that was published. The original company was the favourite for a contract prior to this presentation; they did not get the contract!

TRUST OF ACADEMICS, OPENNESS OF ACADEMIC INSTITUTIONS

Having dealt with the problems of the commercial worries of companies who might give data, we must now consider the worries of the possible data providers related to the academics who will have access to the data. A bad experience regarding a particular researcher, may destroy future trust in unrelated projects. A second area of worry relates to the security of the data once on an academic site. Universities' are renowned and rightly so for the openness and freedom of access to the facilities. However a commercial company worries who might get access to the data and what might they do with it.

CLIENT TRUST

One of the other problems is that the people who may have data do not know, and hence do not trust, the researchers. In the case of a software house asking for a client data this can be particularly so. It then takes a period of time to gain trust of the data provider. Invariably this must be on a personal basis and effectively the provider is basing his trust of the overall group on the contact with the individual.

BUREAUCRATIC PROBLEMS

In large companies multiple levels of clearance are often required. This is especially so when the company is sensitive, due to past experience of being misrepresented. It does not usually matter if the problems are in the same area as the data whose clearance is being requested. However, quite often these do relate to failure information and hence to the reliability data area since system failures can be sensational.

A CLASSIC EXAMPLE

A classic example is the case of a data-set which relates to failed repairs which had been previously cleared for release to the researchers. The owner of this data-set was reprimanded by a senior manager who did not appreciate the reasons for the release of the data. Subsequently the owner would not let other people within the same organisation have access.

6 DELIVERY

After all the possible problems and pitfalls discussed in the previous sections have been overcome, one might be forgiven for thinking that the process of actually giving the data to the researchers and the process of their accessing it would be the easy part of the operation. However the problems still abound in this area.

The problems fall into two basic categories:-

1) The physical transfer of the data.

2) The layout and structure of the files, both in supplied and final form.

The problems of physical transfer are limited to the problems of finding a suitable route from the point at which the data is held, to the point of creating the physical media and file type.

A typical chain associated with this problem might start with the main body of the data stored on a Mainframe system. The data might then have to be transferred to another machine which has links to a workstation. In turn this might have a link to a machine which is capable of writing the floppy disc that is the required input medium.

The above chain would be impossible without the support of a suitable data network. Of course, if one had captured the data on a suitable machine in the first place then the problem would not occur. However, since the data is not usually collected for the purposes of the research this is not under the control of the people involved in the delivery of the data-set.

The second category of problems is associated with the mismatch between the form of the data and the predefined structure of the data-base. The onus on mapping such data across must lie with the data-base holder, consultation with the provider.

Three problems must be overcome:-

1) How is the structure of the information to be conveyed?

2) What does one do if the data does not have fields that are in the database?

3) What does one do if the missing attribute is a key to the researchers database?

Consider a file of records, each of which has some structure, which may be given in a record header; this file can be dumped in a binary form so as to preserve the data. A small program must be written for transferring the data using the provided structure information into the data-base format. However a problem will occur because the binary data will cause the utilities in the receiving environment to fall over.

The second problem must be overcome by a person with a good knowledge of the structure of the receiving data-base. They must decide which links or entities to remove in order to load the data. This takes time and effort to overcome.

The third type of problem implies immense difficulties if comparisons and analyses are to be made across the data-sets as effectively it implies a restructuring of the data-base in order for the new data to be accommodated with retrospective changes to the other data-sets.

7 PROBLEMS OF ANALYSIS

Due to the difficulties and time delays in establishing the central computer facility and to loading data centrally, a number of data-sets were delivered directly for initial Task 9 analysis, and subjects to other researchers. The format of the data-sets have ranged from summaries of failure counts on networked systems, completed failure and repair reports on field data, software test and inspection information and cpu times to failure for individual computer installations. The problems and methods of analysing one of the delivered data-sets are discussed in the rest of this section.

In the absence of a relational data-base at Trent and as the data-set was very large; more than one hundred thousand data items; a suite of programs had to be written so that the data could be easily manipulated. Examples of the type of program written for data analysis were one to sort the records by any column numerically, one to sort by chronological order, one to merge files by one or two relations and one to count the number of records in a file.

Two files 'Fault' and 'Failure' were looked at. The first file to be looked at was the file 'Failure' because this would supply the information gleaned from the failure reports, notably, failure number and the failure date which would be immediately useful for statistical analysis. By sorting this file by date order, the majority of the failure numbers occurred in the correct numerical order apart from the first twenty two which were all dated 01/01/1985 and the product version was 0. The failure numbers for these

were mainly in the 600's. These failures could be assumed to have occurred prior to the start of the project which was given as 16/07/1986 and were collected under a different numbering scheme. However by re-ordering the data failure number, the 22 failure numbers did not tally with any other numbers. The tentative conclusion was that these 22 failures were not date coded on the original failure reports and so the date 01/01/1985 and product version 0 were default values.

The number of failures in the file 'Failure' were counted and compared with the highest failure number. This showed that there were four failure numbers missing. This same technique was carried out for fault number and this showed that there were 500 numbered faults and 170 '0' numbered faults out of a total numbering sequence of 607. Some of the reasons given for '0' numbered faults were as follows: no fault found; the failure record was superseded and cleared by another record; the fault was unconfirmed; a change in the functional specification caused the fault to be non-relevant; minimal effort was required to effect the fault repair; the fault was not important enough at the time to be repaired and was left until a later date.

The two files 'Failure' and 'Fault were merged to find other missing fault information. Merging was carried out using the common 'fault' field. Three fault numbers were missing out of a total of 607 fault numbers in the file 'Fault' and 119 faults were not recorded on failure reports. A similar exercise was carried out on the other files and a number of omissions and anomalies were noted.

A number of files were merged to reduce the information which had only been collected to establish relationships in the data-base structure and the resulting data was surveyed for usable statistical data. Information pertaining to programmer, repair date and repair programmer

could not be utilized as too much data was missing. As an example, there were 276 failure investigations with the default date 01/01/1985 out of a total of 670.

Information pertaining to 'source language' could not be incorporated in a proportional hazards analysis as the twelve sources analysed were all in Cobol.

For the purposes of analysis, certain features of the data are extremely inconvenient. For example, an analysis of the 96 failure free days was carried out and only on two occasions were failures recorded on a weekend. Out of the 16 weekends of the test phase during which data was collected, there were no failures recorded on 12 Saturdays and 10 Sundays. On two occasions there were long sequences of failure free days during the field phase. One of these periods was identified as Christmas and New Year.

An analysis of the number of failures per day was carried out using time series analysis and proportional hazards modelling, the results of which will be presented at the 1989 Reliability Symposium.

8 CONCLUSIONS

There are organisational and technical difficulties in collecting software reliability data.

These are potential features of software data which make the reliability analysis extremely inconvenient.

It is essential that from the inception of a software reliability project, the collection and analysis of reliability data is under strong management control. For example, there must be good feedback to the data providers.

9 REFERENCES

(1) Mellor. P. 'Database Definition for Collection of Software Reliability Growth Data.' ALV073/TSK2/TCU/DBA-SEDEF Issue 1.0, 18.12.86.

(2) Potter. M. 'Software Reliability Modelling Database: Technical Manual.' August 1988.

Note: References are documents internal to the Alvey Software Reliability Modelling Project ALV/PRJ/SE/702. The reference 'ALV073' in (1) was assigned in error when this document was first produced.

APPENDIX 2

McCollin, C., Bendell, A. and Wightman D.W. (1989).
Effects of Explanatory Factors on Software Reliability.
Proceedings of Reliability 1989 Vol 2, pp 5Ba/1/1-11.

EFFECTS OF EXPLANATORY FACTORS ON SOFTWARE RELIABILITY

C. McCollin, A. Bendell and D.W. Wightman
Trent Polytechnic Nottingham

As well as considering the general discussion of explanatory factors within software reliability, this paper is primarily concerned with the statistical analysis of a software reliability failure data set containing explanatory factors. The data set analysis is that designated number three under the Alvey Software Reliability Modelling (SRM) Project. This is a multi-task project consisting of a collaborative team of initially 10 UK organisations. The membership currently consists of the National Centre for Systems Reliability (UKAEA), British Aerospace, STC, Logica, Trent Polytechnic and City and Newcastle Universities. The project is split into a number of tasks of which this paper describes work in task area 9 (data collection and analysis) and task area 3 (statistical models with explanatory variables).

1 BACKGROUND

In the hardware reliability field, MIL-HDBK-217 has been used in military applications for nearly twenty years, and contains failure rate data for electrical and electronic components. The statistical model employed in this standard is the Aarhenius equation, a model which relates part base failure rate to temperature stress. A failure rate for most environments, quality factors and component stresses may be calculated by multiplying part base failure rate by each of these factors. The document is mainly used for compiling lists of failure rates of parts within a system to be compared with some contractual reliability statement. Its main advantage is that it relates component failure rate to explanatory factors and hence by choosing components which have a low valued explanatory factor e.g. Voltage stress, current rating; the failure rate (by analysis) is reduced.

For software, the explanatory factors are more diverse and problems can arise in analysis of estimating these factors due to external influences such as data collection methodology, quality of data and inappropriate statistical models.

In Table 1, we show the software life cycle and the potential explanatory factors that may be appropriate to describe the subsequent failure pattern. A number of previous papers on statistical models incorporating explanatory factors for software are referenced as appropriate in Table 1.

As can be seen in the table 1, the main statistical models which have been used in the literature on the subject of explanatory factors in software reliability have been time series and various forms of the Cox proportional hazards model.

The work in this paper uses standard Box and Jenkins time series and the distribution free base-line hazard Cox model with the metrics, time between failures and number of failures per day. In section 4, the explanatory factors used in these analyses allow the data collection strategy to be scrutinised. In section 5, the proportional hazards model is used to compare within source variation and between sources variation by using suitable explanatory factors.

2 DATA SET DESCRIPTION

The data set was delivered to Trent Polytechnic in twenty files containing more than 100,000 records. This is now installed in a relational database on the Alvey computer Mainframe at City University. The following system description uses the database definitions as described in the SRM database document.

The system under analysis is one software product running on a single installation. The software comprises of 1198 source version codes of which 1096 are written in Cobol, 99 in VOS and 3 in PL1. 1117 of these are greater than zero 4K blocks long of program code and text. There are 87 command macros, 6 command macro data files all in VOS, 608 module main source codes, 78 bind control files, 126 Cobol include files and 21 screen form definition files.

Explanatory data analysis on the Alvey number three data set was carried out including use of time series analysis and proportional hazards modelling.

3 PRELIMINARY ANALYSIS

Sorting, counting and merging files was carried out initially to find any missing or corrupted data. The following observations were made of the data set:

1 There were 125 days with failure and 96 days without failure.

2 After the software was delivered to the customer (during the "live" phase) there was a large increase in the number of days per product version.

3 In total, 570 "test" failures and 100 "live" failures were recorded.

4 There were four missing failures:numbers 455, 457, 591 and 660.

5 There were 170 zero numbered faults, some of the reasons being as follows: no fault found; the failure record was superceded and cleared by another record; the fault was unconfirmed; a change in the functional specification caused the fault to be nonrelevant; minimal effort was required to effect the fault repair; the fault was not important enough at the time to be repaired and was left until a later date.

6 The same fault occurred on separate failure reports 28 times.

7 Two failures, numbers 118 and 279, did not correspond to any fault or repair information.

8 No fault and repair information was found for fault numbers 35, 547 and 553. 9 Out of 670 failure investigations, there were 276 with the default date 01/01/1985.

10 There were 677 unfailed source versions and 514 source versions which failed at least once.

Figure 1 summarises the failure and repair information of the data set. As can be seen, the data format could come from any data collection scheme.

A count of the number of source failures per day was carried out and a time series approach was adopted to determine if trend, lags and/or seasonality had any effect on this count. The fitted time series structure was then used as covariate information in a proportional hazards model. The results are summarised in section 4.

Table 2 and table 3 show the number of times sources and source versions were repaired. The three sources, numbers 489, 274 and 655, which required most repairs were all Cobol include files. Of the remaining 9 sources which were repaired more than 10 times, eight were module main source codes, the other being a Cobol include file. These twelve sources were repaired 217 times out of a total 926 sources with 1356 repairs. The twelve sources were all Cobol files of size greater than 9 4K blocks of code and text of which for 10 of these, only one particular source version was repaired. The twelve sources have been analysed using proportional hazards modelling, the results being discussed in chapter 5.

4 ANALYSIS OF FAILURE COUNTS

A large portion of the literature on failure analysis in the past has dealt with times between failures, Thompson(1981) and Ascher and Feingold (1984).

This section describes possible methods of analysing numbers of failures per day of the software product irrespective of the source codes which have failed.

A paper by Smith and Oren (1980) describes a Nonhomogeneous Poisson Process derived from number of failures in a time interval which may be more applicable in this instance.

Proportional hazards modelling has been carried out in the past with a number of failures as a metric and also as a covariate: Kalbfleisch and Prentice (1980), and Lawless (1987) and hence may also be relevant.

Observing the plot of number of failures per day of the data set illustrated in figure 4, a time series model approach appeared appropriate. The Box Jenkins (1976) approach was used and the following parsimonious model was derived. Walls and Bendell (1986) applied the same method but tried to explain all the variation in their data with complex time series models.

The number of failures on a certain day =

a constant C1 x The number of failures on the previous day

+ a constant C2 x A moving average number of failures per day

+ a constant C3 x A weekly seasonal component.

The values of C1, C2, C3 were 0.92, 0.65 and 0.87 respectively which shows that the number of failures per day is decreasing and is tending towards zero. Hence the debugging strategy appears to be effective.

However, the test phase accounted for most of the structure in the data as found when the data was split into the test and live phases. If there is any structure during the live phase, it cannot be resolved into a simple model.

An analysis of the 96 failure free days was carried out and the following observations were made:
1. There were 26 failure free days during the test phase and 70 during the live phase.
2. Only on two occasions were failures recorded on a weekend. These two occasions occurred near the start of the data collection.
3. Out of the 16 weekends of the test phase during which data was collected, there were no failures recorded on 12 Saturdays and 10 Sundays. The system was running continuously for

the duration of the project however the data collection does not seem to be effective at weekends.
4. There were only 4 other failure free days during the test phase.These probably occurred on personnel holidays.
5. On two occasions there were long sequences of failure free days during the live phase. One period of 13 days was identified as Christmas and New Year and the other of 11 days occurred after a very large number of failures (15) in a day.

Possible reasons for these are:
System utilization was high.
It was decided to raise reports against all minor failures which have been previously reported and ignored.
A new and enthusiastic repair programmer!

5 ANALYSIS OF TIMES BETWEEN FAILURES

Proportional hazards modelling (PHM) was applied to the data using time between failures in days as the metric. Analysis was carried out using the covariates; age, previous number of faults, source version change and type of use for within-sources variation and age, previous number of faults, type of use, source size and source type for between sources variation. Information pertaining to programmer, repair date and repair programmer could not be utilized as too much data was missing. The covariate, "source language", could not be incorporated as the twelve sources analysed were all in Cobol. The results of the analysis of the twelve sources which were repaired the greatest number of times is given in table 4.

The computer routines for fitting PHM written by Dr Wightman successively removes each insignificant covariate one at a time in the model until all the remaining covariates are significant. A number of diagnostic plots are available to determine goodness of fit, outliers and distribution fit and some examples are supplied with a summary of the analysis below.

The covariates, age and previous number of faults were fitted into the model for each of the twelve sources analysed and in every case, even though some of the covariates were nonsignificant at the 0.05% level, the hazard decreased as the sources aged and the hazard increased with the increasing previous number of faults.

The influence of an event at time t, upon the estimate of the covariate is calculated by taking the first order approximation based on a Taylor series expansion of the difference between the estimate of the covariate value with all the observations included and the estimate of the covariate value with the observation at t, omitted. This is then transformed into a normal deviate and compared with $\pm$ 1.96 to determine if the event alters the significance of the covariate if it is omitted.

Figure 3 shows the influence plot of the covariate no. Of faults for source number 102. Removing any of these influential points from the analysis makes the covariate nonsignificant.

Cox and Snell (1968) obtained residual quantities which should be roughly exponentially distributed if the proportional hazards model is a good fit. Plotting a product limit survivor function estimated from the set of residuals against the residual estimates produces a graphical goodness of fit test for the model since the plot should result in a straight line with gradient 1. Source numbers 175, 606 and 737 do not produce good fits possibly due to lack of data. Source number 489 Cox and Snell stabilized fit (figure 4) is showing a marked deviance from the 45° line and this may be due to missing covariate information. For the sources 274, 422 and 546 there were two versions of the software being tested at once, one on the test facility and one on the customer site. There was no significant difference between the hazard rates of versions one and two of each of the sources.

Between sources variation was analysed by combining all the data of the individual sources and using covariates such as source size or type and source designation (a binary covariate).
A summary of the analyses follows:
The hazard rate of all the sources decreased as age increased.
The hazard rate of all the sources increased as number of failures increased. Figure 5 shows the baseline hazard rate for all the sources times between failures to be Weibull.
The hazard rate of all the sources decreased as the software went from a test to the live phase.

The hazard rate was not significantly different for different types of source, different source sizes and different designations. A reason for this may be the sample size of each of the source times between failures being too small.

6 CONCLUSIONS AND FURTHER WORK ENVISAGED

Time series analysis and Proportional hazards modelling have been applied to a large Alvey data set. By using explanatory factors the number of faults raised per day was found to be dependent on the day of the week. The test phase accounted for most of the structure within the data and the number of faults per day during the live phase appear randomly.

The analysis showed that poor quality or insufficient data gave rise to explanatory factors either being nonsignificant in the model such as source size or unable to be combined in the model at all, e.g. Programmer.

Previous data analyses in the literature had to use artificial explanatory factors such as fault count or days to failure which can only partially explain the true nature of the fault process. It was shown in the Table 1 that types of test, failure criticality and source attributes have not yet been analysed anywhere in the literature. This will be possible with some of the Alvey SRM project future data sets analyses.

Statistical analysis has not, so far as the author knows, been carried out in the requirements phase of a software life cycle. The use of sequential probability ratio tests within a proportional hazards framework (Sellke and Siegmund 1983) may be useful in closing the gap.

The object of this paper was to identify the explanatory factors which may affect software reliability and show that the proportional hazards model is useful for analysing data collection schemes, within and between sources variation and different environments.

REFERENCES

1 Ascher, H.E.and Feingold, H. 1984,"Repairable Systems Reliability Modelling, Inference, Misconceptions and their Causes." Marcel Dekker.

2 Box, G.E.P. And Jenkins, G.M. 1976, "Time Series Analysis Forecasting and Control", Holden-Day.

3 Cain, K.C., And Lange, N.T., 1984, "Approximate Case Influence for the Proportional Hazards Regression model with Censored data.", Biometrics,40, 493-499.

4 Cox, D.R., And Snell, E.J., 1968,"A General definition of residuals.", J.R.S.S.,Ser B, Vol 30, 248-275.

5 Davies N. Et Al "The Musa data revisited: Alternative Methods and Structure in Software Reliability Modelling and Analysis" Safety and Reliability Symposium, Altrincham, 1987.

6 Font. V. 1985, "une approche de la fiabilite des logiciels: modeles classiques et modeles lineaire generalise", Thesis L'Universite Paul Sabatier de Toulouse, France.

7 Kalbfleisch, J.D., And Prentice, R.L., 1973, "The Statistical Analysis of Failure Time Data". Wiley, New York.

8 Lawless, J.F., 1987,"Regression Methods for Poisson Process Data" J.A.S.A. Vol 82 No.399, 808-815.

9 Mellor. P. 1988, "Design of a Relational Database to hold Software Reliability Data", ALV072/TSK9/TCU/DBASEDES.

10 MIL-HDBK-217 "Reliability Prediction of Electronic Equipment" RADC.

11 Nagel. P.M. and Skrivan. J.A., 1981 "Software Reliability Repetetive run experimentation and modelling", Rep BCS-40366, Boeing Computer Company NASA Rep no CR-165836.

12 Reid, N., And Crepeau, H., 1985, "Influence functions for proportional hazards regression.", Biometrika, 72, 1, 1-9.

13 Sellke. T. And Siegmund. D. 1983, "Sequential analysis of the proportional hazards model", Biometrika, 70, 2, 315-326.

14 Smith.S.A. And Oren, 1980 "reliability Growth of Repairable Systems" Nav Res Logistics Quarterly, 27, 539-547.

15 Thompson Jr. W.R. 1981 "on the foundations of Reliability" Technometrics Vol 23, No 1 February, 1-13.

16 Walls. L.A. and Bendell. A. 1986, "Time Series Methods in Reliability", 9th Advances in Reliability Technology Symposium.

17 Wightman, D.W., 1987, "The Application of Proportional Hazards Modelling to Reliability Problems." Unpublished Ph.D thesis, Trent Polytechnic.

18 Wightman. D.W. And Bendell. A. 1986 "Proportional Hazards modelling of software failure data" Software Reliability State of the Art Report, Pergamon Infotech.

Table 1

List of potential explanatory factors for the software lifecycle

Phase	Explanatory Factor	Reference Code
Requirements, Specification	count of faults per page	
	hours to check	
	length of document	
	type of fault found	
	experience of checker	
	method of checking	
Code/Test	number of faults per source	WBC,MC,FC
	time between failures:days	WBC,MC
	,, ,, ,, :operating hours	
	,, ,, ,, :execution	WBC,FC
	total time since start of test	WBC,NC,MC
	time to first failure	
	length of source:lines of code/text	MC(NS),NC
	,, ,, ,, :object code	
	,, ,, ,, :executable code	
	language	
	type of source	MC(NS),NC
	computational complexity	
	type of test/test comparison	NC
	test regime	
	intensity of test	
	type of fault/fault comparison	NC
	criticality of fault	
	skill/experience of programmers	
	comparison of programmers	NC
	type of input	FC
	type of output	FC
	user interface	FC
	usage:day	FC
	,, :night	FC
	,, :continuous over a period	FC
	loading	
	number of calls to external modules	
	nesting complexity	
	status of the compiler	
	effect of design change	DC
	functions of explanatory factors	DC
	effect of maintenance	
	mathematical complexity	
	fault reports open/closed	
Integration	hardware/software fault	
	type of source	
	level of integration	
	type of test	
	intensity of test	
	skill/experience of tester	
	comparison of testers	
Installation/Use	type of installation	
	comparison of installations	
	usage/loading	
	effect of maintenance	
	effect of design change	

Table 1 (continued)

List of potential explanatory factors for the software lifecycle

Phase	Explanatory Factor	Reference Code
Others	comparison of phases	MC
	comparison of software products	
	comparison of source versions	MC(NS)
	data collection methodologies	
	:number of faults per day	MTS,MC
	:day of the week	MC
	:previous number of faults/day	MC,MTS
	:cumulative number of faults	MC
	time series components:autocorrelation	WTS,MTS
	,, ,, ,, :moving average	WTS,MTS
	,, ,, ,, :seasonality	WTS,MTS
	,, ,, ,, :nonnormality	DTS
	,, ,, ,, :nonlinearity	DTS
	,, ,, ,, :outliers	DTS
	change in management structure	
	costs	
	staffing levels	

Reference Codes

Reference	Data Set	Statistical Model	Code
Davies et al	Musa	Cox/distribution free hazard	DC
,, ,, ,,	,,	Bayesian time series	DTS
Font	Font	Cox/Littlewood,Musa hazard	FC
McCollin et al	Alvey	Cox/distribution free hazard	MC
,, ,, ,,	,,	Box and Jenkins time series	MTS
Nagel and Skrivan	Nagel	Cox/exponential hazard	NC
Walls and Bendell	Musa	Box and Jenkins time series	WTS
Wightman and Bendell	Musa	Cox/distribution free hazard	WBC

(NS) after code MC means that the explanatory factor was nonsignificant in the cox proportional hazard model.

TABLE 2-Number of Times Sources Repaired

Repairs	0	1	2	3	4	5	6	7	8	9
Sources	437	232	100	44	30	28	20	4	10	3

Repairs	10	11	12	13	14	15	20	32	51
Sources	6	3	1	2	2	1	1	1	1

TABLE 3-Number of Times Source Versions Repaired

Repairs	0	1	2	3	4	5	6	7	8
Source Versions	639	289	111	50	32	27	16	8	8

Repairs	9	10	11	12	13	14	15	19	21
Source Versions	5	5	0	1	2	2	1	1	1

Figure 1-Failure and Amendment Record Information

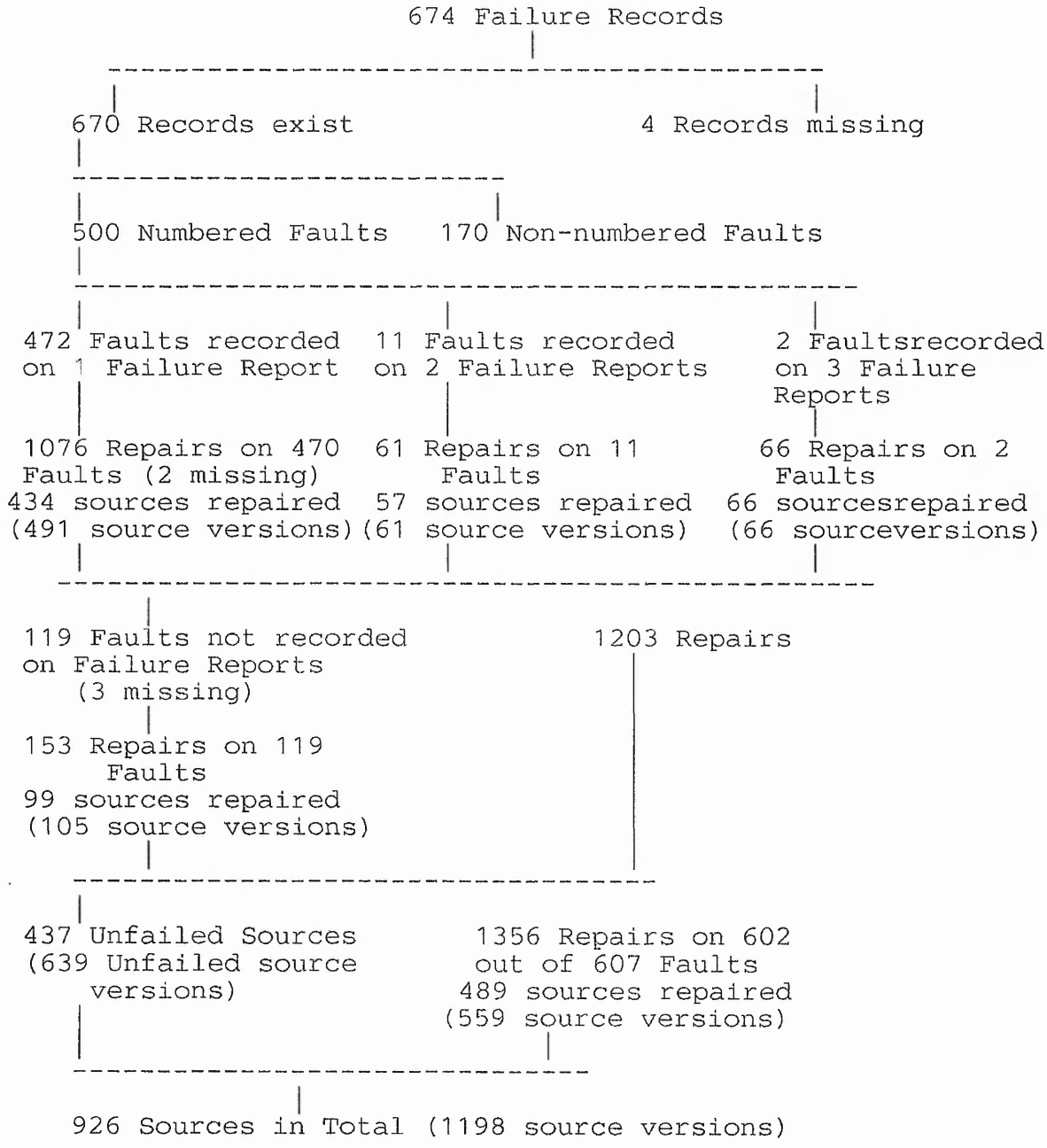


TABLE 4-Covariate information of within sources variation

SOURCE NUMBER	NO.OF FAILURES /FAULTS/ CENSORS	NO. OF SOURCE VERSIONS	COVARIATE	VALUE	SIGNIF-ICANCE	LIKELI-HOOD RATIO	COMMENTS
102	11/0/2	2	AGE	-0.164	0.0123	9.332	
			NO.OF FAULTS	0.967	0.0187		4 infl.pts Not a good Cox fit
175	10/2/2	1	AGE	-0.0361	0.0153	11.066	Not a good Cox fit
			NO.OF FAULTS		N.S.		
175 TEST PHASE	9/1/1	1	AGE		N.S.		
			NO.OF FAULTS		N.S.		
274 VERSION ONE	9/4/1	1	AGE	-0.4721	0.0158	13.815	5 infl.pts Very little data
			NO.OF FAULTS	1.5212	0.0138		5 infl.pts Very little data
274 VERSION TWO	14/5/1	1	AGE		N.S.		
			NO.OF FAULTS		N.S.		
			TYPE OF USE		N.S.		
274 TWO INDEPENDENT VERSIONS	23/9/2	2	VERSION CHANGE		N.S.		
307	12/1/1	1	AGE		N.S.		
			NO.OF FAULTS		N.S.		
422 VERSION ONE	4/0/0	1	AGE		N.S.		
			NO.OF FAULTS		N.S.		
422 VERSION TWO	10/0/1	1	AGE		N.S.		
			NO.OF FAULTS		N.S.		

TABLE 4-Covariate information of within sources variation

SOURCE NUMBER	NO.OF FAILURES /FAULTS/ CENSORS	NO. OF SOURCE VERSIONS	COVARIATE	VALUE	SIGNIF-ICANCE	LIKELI-HOOD RATIO	COMMENTS
422 TWO INDEPENDENT VERSIONS	14/0/2	2	VERSION CHANGE		N.S.		
489	29/20/23	23	AGE	-0.0129	0.0019	10.853	Missing Covariate?
			NO.OF FAULTS		N.S.		
			VERSION CHANGE		N.S.		
546 VERSION ONE	9/0/1	1	AGE		N.S.		
			NO.OF FAULTS		N.S.		
546 VERSION TWO	4/0/1	1	AGE		N.S.		
			NO.OF FAULTS		N.S.		
546 TWO INDEPENDENT VERSIONS	13/0/2	2	VERSION CHANGE		N.S.		
560	10/5/1	1	AGE		N.S.		
			NO.OF FAULTS		N.S.		
606	11/0/3	3	AGE		N.S		
			NO.OF FAULTS	-0.2915	0.0068	7.276	Not a good Cox fit
655	9/10/5	5	AGE		N.S.		
			NO.OF FAULTS		N.S.		
737	11/0/3	4	AGE	-0.0271	0.0065	12.114	Not a good Cox fit
			NO.OF FAULTS	NO CONVERGENCE			
807	13/1/1	2	AGE		N.S.		

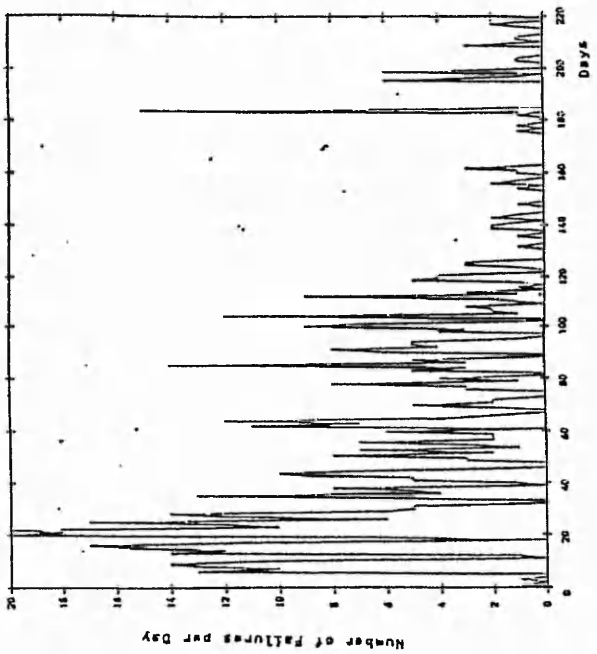


Figure 2:Plot of Number of Failures per Day against Days

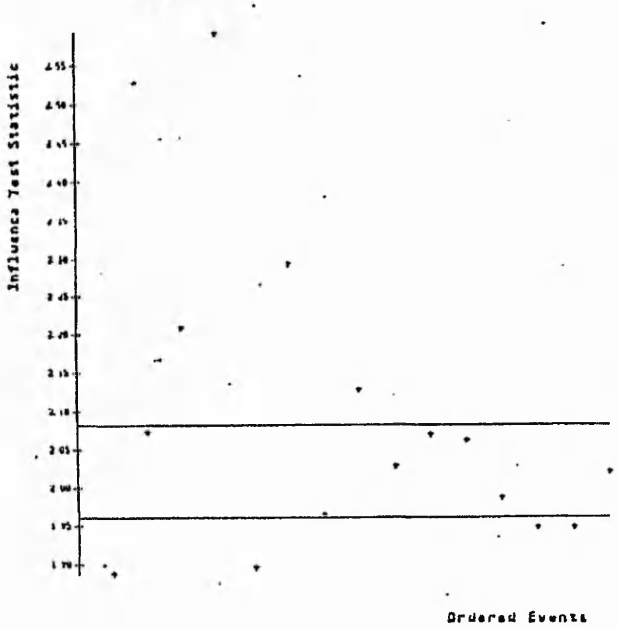


Figure 3:Influence Plot for Covariate "Number of Faults", Source 102

CAPTIO

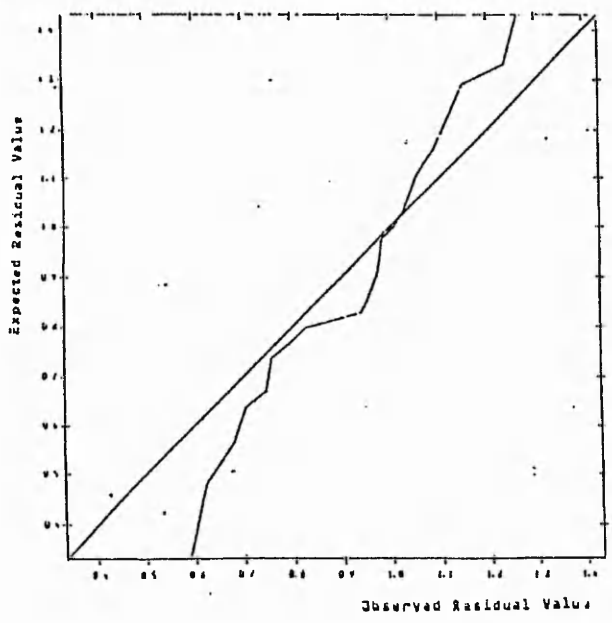


Figure 4:Stabilised Cox and Snell Residual Plot for Source 489

PTION

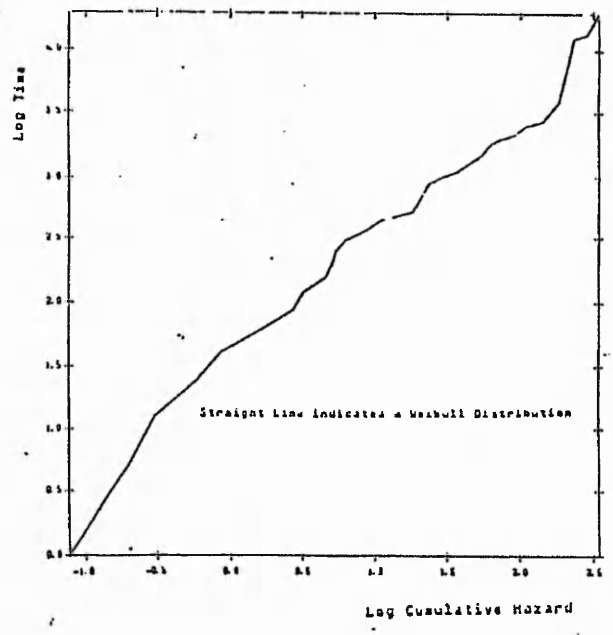


Figure 5:Weibull Baseline Hazard Plot For all 12 Sources

5Ba/1/11

Paper and Page No.....

Master Sheet for illustrations -- use as ¼ ½ or full page

For reduction to A4

APPENDIX 3

McCollin, C., Wightman, D.W., Dixon, P. and Davies, N. (1990).
Some Results of the Alvey Software
Reliability Modelling Project.
Proceedings of the SARSS. Altrincham, 1990.

SOME RESULTS OF THE ALVEY SOFTWARE RELIABILITY MODELLING PROJECT

C. McCOLLIN, D.W. WIGHTMAN, P. DIXON and N. DAVIES
Department of Mathematics, Statistics &
Operational Research,
Nottingham Polytechnic
Burton Street, Nottingham NG1 4BU

ABSTRACT

The Alvey software reliability modelling project is a multi-tasked project consisting of a collaborative team from UK industry and academia. Over the duration of the project, the membership has consisted of the National Centre of Systems Reliability (AEA Technology), British Aerospace, STC, Logica, Nottingham Polytechnic and City and Newcastle Universities. The objectives of the software reliability modelling project were to investigate a wide variety of methods, to judge the relative merits of each method, to effectively communicate the results of the research and to indicate the direction of future research. The project consisted of a number of tasks of which this paper describes areas in which Nottingham Polytechnic were task leaders; these are task 3 (statistical models with explanatory variables), task 4 (statistical models with different underlying assumptions) and task 9 (data collection and initial analysis).

INTRODUCTION

History

There has been growing concern in the software industry about unreliable software for many years and as a result there have been some initiatives aimed at reducing the impact of the problem. Customers have imposed codes of practice on suppliers, lists of "approved" software have been specified and work has been done on better testing strategies. Up until now little co-ordinated research has been done nationally on modelling software reliability. The appearance in 1984 of the "Software reliability and metrics programme" document from the Alvey Directorate formed a natural focus for this work. A consortium was formed containing members from both academic and commercial backgrounds with the intention of conducting a research programme to improve the state of the art.

In July 1985, the Alvey Directorate placed a contract for the detailed study of software reliability modelling (SRM) with the aim of producing a plan for a National SRM Programme. The suggested course for the research was instilled into a set of project tasks. These tasks were as follows:-

Task 1: Improving Current Statistical Models
Task 2: Methods of Evaluating Statistical Models
Task 3: Statistical Models with Explanatory Variables
Task 4: Alternative Statistical Models
Task 5: Functional Modelling
Task 6: Models for Special Systems
6.1: Models for VLSI Systems
6.2: Models for Distributed Systems
6.3: Concurrent/Real Time Systems
6.4: Models for Fault Tolerant Systems
6.5: Reusable Software Components
Task 7: Cost Based Models
Task 8: Testing and Reliability
Task 9: Data Collection and Analysis

The project finished in June 1990 and more than sixty documents have been written during the project and a number of these are available to the public. The address for further details is given in the summary.

This paper describes some work carried out in each of the tasks 3, 4 and 9. The process of analysis of one of the data sets collected for the Alvey project is described with the problems of assessing software reliability for this data set using the classical software reliability models e.g [1], [2], [3].

A description of other work carried out in task area 3 includes models which incorporate explanatory variables and extensions to published models. In task 4, the application of time series and multivariate techniques to software reliability is described.

WORK CARRIED OUT IN TASK AREA 9

In task area 9, considerable effort was invested in the creation of a software reliability database which is installed on a dedicated Vax computer at City University. Private computer links to the database are available to the other academic partners. It is envisaged that the usefulness of the database will continue beyond the duration of the project. The format of the data sets collected ranged from summaries of failure counts on networked systems, completed failure and repair reports on field data, software test and inspection information and cpu times to failure for individual computer installations.

The problems of collecting software failure data are highlighted in reference [4] and a description of a statistical analysis of one of the data sets appears in [5]. The main purposes of the data analysis exercise were to determine suitable models for software reliability estimation and to establish models which would incorporate explanatory factors found in analysis. The data set was collected during the development phase of the project and the software was continually being operated and repaired after failure.

The analysis of Alvey data set 3 and conclusions which led to further analysis were carried out in the following order:

Sorting, counting and merging file data to find any corrupt or missing data.

Plotting the number of failures per day against day.

Applying a parsimonious Box-Jenkins time series model to the data. This revealed a decreasing trend and a seasonal component.

An analysis using Proportional Hazards Modelling (PHM) was applied to the data set to determine the effect of the seasonality. The analysis showed that the hazard rate was increasing on specific days of the week. Two observations were made during the analysis. The first was that a number of explanatory factors could not be fitted together in PHM. This was due to the factors being collinear.

The problem of multicollinearity of the covariates was investigated by applying multivariate techniques to the data set and the results of this are described under the task 4 work heading. The second observation was that for this data set and a number of others execution time to failure was not available. The main software reliability models [1], [2], [3], use execution time as their time metric. It was found during the data collection exercise that companies do not usually collect execution time to failure of programs because:

It is a costly exercise to collect execution time to failure.

The customer only usually requires execution time if he wishes to estimate software reliability by using one of the available models.

The collection of execution time is not usually a requirement of general software guidelines or standards.

Based on these observations, further work on explanatory factors was carried out and this is described in the Task 3 section.

WORK CARRIED OUT IN TASK AREA 3

Review of models
A report [6] was written which reviewed the models and techniques which incorporate explanatory variables and can be adapted to software reliability modelling. Techniques and methodologies reviewed were Software Science, Information Theoretic approach, simple regression, multivariate analysis, proportional hazards modelling and generalized linear modelling.

Explanatory Variables highlighted in task 9: Development of a modelling framework which incorporates these.
The following is a list of explanatory variables and their associated time metric. Each variable should be taken into account when deciding on a suitable software reliability model and its modelling assumptions.

Time metrics associated with the explanatory factors

CPU time
Execution time
Operating time
Calender time

Explanatory factors associated with CPU time

A1-Language
A2-Type of program
A3-Computational volume
  -Length of machine code/text
A4-Computational complexity
  -Nesting complexity
  -Number of calls to external modules
  -Number of conditional statements
  -Type of input/output
A5-Mathematical complexity
A6-Loading
A7-Programmer skill/experience

Explanatory factors associated with execution time

A1-A7
B1-CPU time
B2-Compiler status
B3-Parallel/serial processing
B4-Queueing
B5-Priority
B6-Available storage
B7-Systems availability
  -Peripherals

Explanatory factors associated with operating time

A1-A7
B1-B7
C1-Execution time
C2-Usage
  -Idle time
C3-Type of installation

Explanatory factors associated with calendar time

A1-A7
B1-B7
C1-C3
D1-Operating time
D2-Stoppages
  -Holidays
  -Strikes
  -Shutdown
D3-Seasonal variation
D4-Number of staff
D5-Project deadlines
D6-Data collection method
D7-Job priority

The analysis of the times between failures of the Alvey data set 3 presented in [5] described a PHM formulation with days to failure of programs as the time metric and program type and program size as two of the explanatory variables.

Different program types and program sizes affect the cpu time directly and the calendar time only indirectly by the cpu time. For the results of [5] concerning the two explanatory variables and time metric, days between failure, it was known that the software was continually operating all the time for the duration of the project so that the calendar time between failures was the same as the cpu time between failures accumulated for the complete software package. However, it is not possible to relate the hazard function based on cpu times to program failure to the two explanatory variables unless certain assumptions are made about other explanatory variables, e.g. the usage of the individual programs.

As PHM uses the ranking of the failure times and not the failure times themselves, it can be shown that as long as the ranking of the days between failures remains the same for cpu time, execution time or operating time between failures, then the conclusions concerning the hazard for days between failures are valid for the other time metrics. For example, if execution time can be controlled so that it is always the same function of calendar time, e.g calendar time = a constant multiplied by execution time, then conclusions about the calendar time hazard function will apply to the execution time hazard function.

Extensions to models

The City University has contributed three reports on extensions to existing software reliability models. These cover task areas 3, 4 and 1 (improvement of current models). A Bayesian formulation of the Jelinski-Moranda software reliability model [7] reports that the model performance seems to be at least as good as some other models. In reference [8], a simulation study is reported which investigates if a general but simple adaptive procedure which improves the accuracy of predictions also increases the variability of the predictions. Reference [9] describes an extension to the "u-plot" (used for assessing predictive performance or for obtaining improved "adapted" or "re-calibrated" predictors) to allow for discrete or mixed predictive distributions. Two further modifications of the u-plot are documented which improve the performance of re-calibrated predictors.

A paper relating experience of applying a proportional hazards modelling formulation to data set 5 (data set number as defined in Task 9) has been written. The paper has still to be presented to and cleared by the data providers, [10]. Previous application of proportional hazards modelling has been based upon modulated renewal processes; where the explanatory variables modulate the underlying renewal process, [11], [12], [13] [14]. Recently, Lawless [15] has introduced model formulations which allow explanatory variables to be considered within a Poisson process. These proportional intensity Poisson process models allow the traditional non homogeneous Poisson process software models to be combined with explanatory variables. A particular model formulation investigated at Nottingham Polytechnic is one which depending upon parameter values leads to either a modulated renewal process or a proportional intensity model.

Software has been developed at Nottingham Polytechnic for Poisson Proportional Intensity models with covariates and an unspecified baseline intensity. No data has been applied to this software as yet, however details of the approach are available from the authors.

Work has been carried out at Nottingham in expressing binomial type models and Poisson type models of exponential class (as classified by Musa et al [1], within a PHM framework. Details are available from the authors.

Font [16] derived a proportional hazards model with the Musa model as the hazard function. The following formulation of the Musa model within a PHM framework is useful as a goodness of fit test for the Musa model in that if the number of software failures is not a significant explanatory variable in the PHM formulation then the Musa model is not appropriate for the data analysis.

The Musa basic execution time model [1] takes the form

$$\lambda(t) = \lambda \exp(-\phi t)$$

where

t = total execution time

λ = initial failure intensity

$\lambda(t)$ = failure intensity function

ϕ = "Constant hazard which characterises any individual failure"

The expected number of failures in time t is given by

$$\mu(t) = \int_0^t \lambda(w)dw$$

$$\mu(t) = \lambda(1 - \exp(-\phi t))/\phi$$

From [1], the cumulative hazard function is

$$H(t'/t) = \mu(t+t') - \mu(t)$$

Letting t = last failure time and hence t' = time since last failure.

Thus, $H(t'/t) = -\lambda(\exp(-\phi(t+t')) - \exp(-\phi t))/\phi$

If we differentiate $H(t'/t)$ with respect to t',

$$(dH(t'/t)/dt' = \lambda \exp(-\phi t')\exp(-\phi t) \quad , \quad (1))$$

$$dH(t'/t)/dt' = h(t')$$

Our PHM formulation is $h(t', z) = h_0(t')\exp(\beta z)$, (2)

Where t' :- time since last failure, z is an explanatory variable, β is a parameter of the model and $h_0(t')$ is the baseline hazard.

Now comparing (1) and (2),

i) If $h_0(t')$ from PHM = $\lambda \exp(-\phi t')$ and

ii) $\exp(\beta z) = \exp(-\phi t)$ *with* $\beta = -\phi$ and z=t

then the basic execution model is a sub model of PHM.

WORK CARRIED OUT IN TASK AREA 4

Time series

Time series methods in reliability have been implemented by many authors [17], [18], [19]. The techniques have included the application of traditional linear ARMA models of Box and Jenkins [20]. Although 'discoveries' of trend and cyclical features have been made using these techniques, the whole area is rather unsatisfactory. Nottingham Polytechnic research concludes that invariably almost all the assumptions that are made in applying linear modelling techniques are violated by reliability, and in particular software reliability data. Violated assumptions are linearity, normality, constant parameters, change points and outliers.

Alternative, and more flexible model formulations are provided by the Dynamic Linear models and implementable using the BATS package, developed at Warwick University [21], [22]. Typically, time between failures or time to failures (TTF's) are described by an observation equation

$$TTF(i) = m(i) + r(i) + v(i)$$

where i is the failure number, m(i) a level parameter that evolves with the failures, r(i) is a set of possible covariates (failure dependent) and v(i) is white noise. The extra flexibility is provided by allowing the evolving nature of m(i) and r(i) to be stochastic. The Bayesian (Kalman filter) recursion allows outliers to be handled/detected automatically, missing values, and user intervention with the model. Nottingham Polytechnic has used these techniques to model the MUSA data sets and Alvey data set 8. The approach also allows flexibility in traditional Weibull and hazard modelling. Some results of the above work have been presented to the Highlands local group of the RSS in March 1989.

Multivariate Techniques

The purpose of this section is to describe briefly attempts made to analyse software data by multivariate methods.

The data was designated Alvey data set number 3 and contained 1198 observations on the following variables:

X_1 = program X_6 = type of program
X_2 = program version X_7 = first appearance
X_3 = programmer X_8 = final appearance
X_4 = language X_9 = number of faults
X_5 = size of program X_{10} = time

Data screening and editing were necessary to overcome idiosyncrasies and to render the data meaningful and suitable for analysis. The screening and editing were undertaken using MINITAB. Subsequent analysis was undertaken using MINITAB and GLIM.

Principal Components Analysis. (PCA)
A commonly used multivariate technique is that of Principal Components Analysis, where p correlated variables are combined to obtain a new set of uncorrelated variables, called Principal Components.

The new variables are linear combinations of the original variables and are derived in decreasing order of importance so that PC(1) accounts for as much as possible of the variation in the original data. If the first few components account for most of the variation in the original data, the effective dimensionality of the problem is less than p.

Let $\underline{X}^T = \{X_1, X_2, \ldots, X_p\}$ be a p-dimensional random variable with variance-covariance matrix Σ and let

$$Y_j = a_{1j}X_1 + a_{2j}X_2 + \ldots + a_{pj}X_p = \underline{a}^T\underline{X} \quad , \qquad (j = 1, 2, \ldots, p)$$

where $\underline{a}^T = \{a_{1j}, a_{2j}, \ldots, a_{pj}\}$ such that $\underline{a}_j^T\underline{a}_j = \sum_{k=1}^{p} a_{kj}^2 = 1$ and

$\underline{a}_i^T\underline{a}_j = 0, \quad (i <> j)$

Y_1 is found by choosing a_1 so that Y_1 has the largest possible variance, Y_2 is found by choosing a_2 so that Y_2 has the next largest variance and is uncorrelated with Y_1; Y_3 is found by choosing a_3 so that Y_3 has the next largest variance and is uncorrelated with Y_1 and Y_2 , ... and so on.

Thus obtained, $Y_1, Y_2, \ldots, Y_p$ are the Principal Components (PC) of x having variance equal to the eigenvalues of the sample variance - covariance matrix S ($= \hat{\Sigma}$); [23].
In the case of the software data it is wise to base PCA on the sample correlation matrix P rather than S, thus rendering the variables, which are heteroscedastic, equally important.

The Minitab results were:

(i) Examination of the correlation matrix P showed a sufficiency of non-zero elements to warrant the PCA worthwhile.

(ii) Eigenanalysis of P.

i	1	2	3	4	5	6
Eigenvalue λ_i	2.25	1.52	1.12	0.98	0.64	0.42*
Proportion $\lambda_i/\Sigma\lambda_i$	0.32	0.22	0.16	0.14	0.09	0.06
Cumulative Proportions	0.32	0.54	0.70	0.84	0.93	0.99

(*denotes that subsequent eigenvalues exist but account for only 1% of the variation).

Note that as many as five PC's are required before more than 90% of the variation in the data is explained. Ideally, it is desirable that the majority of the variation in the data should be explained by two or three components at the most. Unfortunately no such reduction of the effective dimensionality was obtained. Reduction to two or three components is useful in that 2D or 3D plots of component score might be examined for patterns or clusters and that attempts at reification might be made. However, it is of some interest that the effective dimension of the data reduces to about five, with this technique.

MINITAB also supplies the coefficients $\underline{a}_j^T$ from the eigenvectors corresponding to each eigenvalue.

Discriminant Analysis.
Multivariate discriminant analysis is a technique which allows the multivariate response for $\underline{X}_*^T = \{X_1, X_2, \ldots, X_j, \ X_{j+2}, \ldots, X_p\}$ to be attributed to known groups according to X_{j+1} provided X_{j+1} is a group indicator, via discriminating functions. The discriminating functions then may be used to assign further observations on $\underline{X}_*^T$, not so far identified on a X_{j+1}, to a X_{j+1}.

A feature of the software data is that, in a number of cases, the multivariate response has not been identified by programmer (X_3). It is of interest to use the data on cases where the programmer is known as a "learning set" for discriminating between programmers, thereby making it possible for cases in the "prediction set", with programmer unknown, to be identified with a programmer.

The procedure is to calculate

$$w_i = \underline{L}_i^T \underline{X}_* - 0.5 \underline{L}_i^T \overline{\underline{X}}_{*_j} + \ln(\pi_i) \quad ; \qquad i = 1, 2, \ldots\ldots, j, j+2, \ldots\ldots m$$

where m is the number of distinct groups (programmers) indicated by X_{j+1}, $\overline{\underline{X}}_{*_i}$ is the mean vector for group i, S_* is the pooled within groups estimate of Σ_*, the variance-covariance matrix of $\underline{X}_*$ and $\underline{L}_i = \underline{S}_*^{-1} \overline{\underline{X}}_{*i}$, π_i is the prior probability that a case belongs to group i, and to allocate the individual to that group for which the w_i is the greatest, [23].

Minitab Results.
Unfortunately, the success rate for correctly identifying the

multivariate response on $\underline{X}_*$ by known programmer in the training set was found to be low, with only 25.5% of cases correctly identified.

However, the success rate varied from programmer to programmer, ranging from no cases correctly identified to 88.9% of cases correctly identified. With a low overall success rate it is inappropriate to attempt to identify programmers for cases in the prediction set.

It is possible that the failure of the technique to achieve a reasonable success rate may be attributed to a violation of the theoretical assumptions of discriminant analysis, that the discriminating variables have a multivariate normal distribution and have equal variance - covariance matrices within groups, (programmers); [23]. The data under study, consisting mainly of variables having a discrete or categorical nature, do not conform to these requirements. Reference [24] gives a discussion on techniques of discrete discriminant analysis applied to data not conforming to the multivariate normal, homoscedastic groups pattern. Reference [25] gives a similar treatment.

Log - linear models.
It is possible to obtain from the software data multi-way tables containing number of faults as response corresponding to variables such as X_4 = program language, X_5 = program size and X_6 = program type. With such categorical data it is appropriate to fit log-linear models, beginning with the no-association model.

$$E(F_{ijk}) = N \pi_{i..} \pi_{.j.} \pi_{..k} \qquad (1)$$

where F_{ijk} = number of faults in the cell of the multi-way table corresponding to the i'th language, j'th program size, k'th program type;

N = grand total of faults in the multi-way table;

$\pi_{i..}$ = marginal probability in the i'th category of X_4 (language) irrespective of X_5 and X_6 (size and type of program), $\pi_{.j.}$ = marginal probability in the j'th category of X_5 (size) irrespective of X_4 and X_6 (language and type); $\pi_{..k}$ = marginal probability in the k'th category of X_6 (type) irrespective of X_4 and X_5 (language and size).

Taking logarithms in (1)

$$\ln E(F_{ijk}) = \ln N + \ln \pi_{i..} + \ln \pi_{.j.} + \ln \pi_{..k} \qquad (2)$$

With a little manipulation it is possible to write (2) in the form

$$\ln E(F_{ijk}) = u + u_{1(i)} + u_{2(j)} + u_{3(k)} \quad , \qquad (3)$$

where the u's are functions of the theoretical marginal fault counts.

Now, (3) is reminiscent of a two-way ANOVA model, with no interaction. It is possible to fit (3) using GLIM, employing the deviance statistic equal to $-2\log(l_c/l_f)$ as the goodness-of-fit criterion, where l_c = likelihood of the data under the current model and l_f = likelihood of the data under the fullest possible model, following the notation of [26].

Failure of the no-association model to fit the data encourages the inclusion of further model terms, firstly the two-way associations

$$u_{12(ij)}, \quad u_{13(ik)}, \quad u_{23(jk)}$$

corresponding to first-order interaction in ANOVA, and then, if necessary, the three-way association $u_{123(ijk)}$, corresponding to second-order interaction in ANOVA, (see [27]).

GLIM Results:

Model	Scaled deviance	change	residual df	change in df
A (3)	89.75			
B A+SIZE.TYPE	77.78	11.97	11	5
C B+SIZE.LANG	26.94	50.84	10	1
D C+TYPE.LANG	0.44	26.50	5	5

Now, the scaled deviance (or change in scaled deviance) is approximately X^2 - distributed with the residual degrees of freedom (or change in degrees of freedom).

It can be concluded that

(a) there is a significant association between size and type of program,

(b) there is a significant association between size of program and language,

(c) there is a significant association between type of program and language,

(d) there is no significant three-way association, suggesting that

(i) the association between size of program and type of program is the same for all languages,

(ii) the association between size of program and language is the same for all program types,

(iii) the association between type of program and language is the same for all program sizes.

Resulting from (b), close examination of the model parameters suggests that a negative association between SIZE (2) and TYPE (5) is indicative of a tendency for a lower fault count with medium to large programs than with small programs of the type "INCLUDE FILE".

Also, resulting from (c), a negative association between TYPE (4) and LANG (2) suggests a tendency for a lower fault count with system operating language programs than with COBOL programs of the type "FIND CONTROL FILE".

Comments: The multivariate procedures described earlier in this section revealed relatively little. However this should not malign the power and usefulness of techniques such as PCA and discriminant analysis, and they should be used if appropriate on other examples of software data in attempts to reveal data structure.

Log-linear modelling, a useful example of which is discussed above, has a very positive usefulness in investigating data of the type considered and is recommended as an important tool in future work.

SUMMARY

Some results of the Alvey Software Reliability Modelling (SRM) project have been presented. A number of models have been generalised or extended to incorporate explanatory variables. Problems of software reliability have been highlighted and further work is required in this area. Multivariate techniques and time series analysis are shown to be applicable to software reliability data and a number of results have been presented. A number of documents have been written for the whole project including tasks 3, 4, and 9 and some of these are available to the public. The authors may be contacted initially for further details of their work in task areas 3, 4 and 9. For further details of other Alvey SRM work, the contact address is DTI/IED, Alvey SRM project, Kings Gate House, 66-74 Victoria Street, London SW1E 65W.

REFERENCES

1. Musa, J.D. and Okumoto, K., Application of Basic and Logarithmic Poisson Execution Time Models in Software Reliability Measurement. Software System Design Methods, Springer-Verlag Berlin Heidelberg 1986, Nato ASI series, Volume F22 pp275-298.

2. Littlewood, B. and Verrall, J.L., A Bayesian reliability growth model for computer software., Journal of the Royal Statistical Society., C (Applied Statistics), 22, 1973 pp 332-346.

3. Jelinski, Z. and Moranda, P.B., Software Reliability Research, in Statistical Computer Performance Evaluation, W. Freiberger New York Academic Press, 1972, pp 465-484.

4. Bendell, A., McCollin. C., Wightman. D.W., Linkman, S. and Carn, R. Software Reliability Data Collection - Problems and Possibilities , Proceedings of the Sixth EureData Conference, Siena, 1989.

5. McCollin, C., Bendell, A. and Wightman. D.W., Effects of Explanatory Factors on Software Reliability, Reliability 1989, Vol 2.

6. Wightman, D.W., Review of models/techniques which incorporate explanatory variables and may be applied to software failure data. Nottingham Polytechnic report (for Alvey SRM project).

7. Csenki, A., Bayesian Formulations of the Jelinski-Moranda Software Reliability Model. Unpublished City University report (For Alvey SRM project).

8. Brocklehurst, S., On the Effectiveness of Adaptive Software Reliability Modelling. Unpublished City University report (for Alvey SRM project).

9. Wright, D.R., A Modified U-Plot applied to Failure Count Prediction., Unpublished City University report (for Alvey SRM project).

10. Wightman, D.W., McCollin. C. and Bendell. A., Proportional Hazards Modelling of an Alvey Software Reliability Data Set. Awaiting publication.

11. Cox, D.R., The Statistical Analysis of Dependencies in Point Processes. Stochastic Point Processes ed P.A.W. Lewis, Wiley, New York pp55-66.

12. Wightman, D.W., 1987. The Application of Proportional Hazards Modelling to Reliability Problems. Unpublished Ph.D thesis Trent Polytechnic.

13. Prentice, R.L, Williams, B.J and Peterson, A.V., On Regression Analysis of Multivariate failure time data. Biometrika Vol 68 No. 2, 1981 pp 373-379.

14. Anderson, P.K. and Gill, R.D., (1982) Cox's Regression model for counting processes a large sample study. Annals of Statistics, 10 pp 1100-1120.

15. Lawless, J.F. (1987) Regression Methods for Poisson Process Data. Journal of the American Statistical Association. Vol. 82, No. 399.

16. Font, V., Une approache de la fiabilite des logiciels: modeles classiques et modele lineaire generalise. Thesis L'Universite Paul Sabatier de Toulouse, France 1985.

17. Singpurwalla, N.D., (1978). Time series analysis of failure data. Proceedings Annual Reliability and Maintainability Symposium., pp 107-112.

18. Singpurwalla, N.D. and Soyer, R. Assessing (software) reliability growth using a random coefficient autoregressive process and its ramifications., IEEE Transactions on Software Engineering 1985.

19. Walls, L.A. and Bendell, A., The structure and exploration of reliability field data; What to look for and how to analyse it. Proceedings of 5th National Reliability Conference 1985 pp5B/5/1-17.

20. Box, G.E.P and Jenkins, G.M., Time Series Analysis: Forecasting and Control Holden-Day, London, 1976.

21. Harrison, P.J. and Stevens, C.F., Bayesian Forecasting (with discussion). J.R. Statistic Soc., 1976, B38, pp205-247.

22. West, M. Harrison, J. and Pole, A., BATS: A User Guide. University of Warwick, 1988.

23. Chatfield, C. and Collins, A.V., (1980), Introduction to Multivariate Analysis, Chapman Hall.

24. Goldstein, M. and Dillon, W.R., Discrete Discriminant Analysis. Wiley, 1978.

25. Bishop, Y.M, Fienberg, S.E. and Holland. P.E., (1975) Discrete Multivariate Analysis, MIT Press.

26. Baker, R.J. and Nelder, J.A., (1978), GLIM System, Released by the Royal Statistical Society.

27. Everitt, B.S., (1977), The Analysis of Contingency Tables, Chapman Hall.

APPENDIX 4

Wightman, D.W., McCollin, C. and Dixon, P. (1991).
Recent Applications of Some Statistical Techniques to Software Reliability Data.
Proceedings of the 1991 Reliability Conference. Reliability 91.
Elsevier Science Publishers.

RECENT APPLICATIONS OF SOME STATISTICAL TECHNIQUES TO SOFTWARE RELIABILITY DATA

D.W. Wightman, C. McCollin and P. Dixon
Department of Mathematics, Statistics and Operational Research
Nottingham Polytechnic

ABSTRACT

A number of statistical models are discussed. Three logistic regression models and a proportional intensity model are applied to software reliability data.

A large software reliability dataset was collected which included failure and fault information as well as attributes of the programs such as size, type of program, etc. Execution time had not been collected and so the well known software reliability models were not applicable in this instance. However, the statistical models presented here equate some function of failure count (or time to failure) to the explanatory information available and the results are very encouraging for further research.

The final section shows how the exponential type software reliability models may be incorporated into a proportional hazards framework.

DESCRIPTION OF THE DATASET

The system under analysis is one software product running on a single installation. The software comprises of 1198 program versions of which 1096 are written in Cobol, 99 in an operating system language and 3 in a third language. There are 6 types of program: 87 command macros, 6 command macro data files in the operating system language, 608 module main source codes, 78 control binding files, 126 binding Cobol files and 21 screen definition files. These were numbered types 1 to 6 respectively for convenience.

The suite of programs were run as a package and failures were recorded on a daily basis with the programs at fault. Repair information was minimal so that this information could not be included in the analysis. The total number of programs found at fault was 1356 for which 674 resulted in the package failing. The package was expanded throughout the development phase and on certain days the number of programs running increased without failures or faults occurring. This censoring information and the failure data with the type of program failed was collected over an eight month period and, for the six types of program, 269 failure and censoring points and 222 failure points were analysed. The total number of programs running per day was recorded and the number of failures/faults per day to total number of programs running per day with number of failures/faults per day was calculated as a proportion. This adding of the top term (number of failures/faults per day) to the bottom of the expression made sure that the proportion always lay between zero and one. Figure 1 is a plot of the proportion of failures against time to failure for each program type. From the figure, there are no immediate observations of any note.

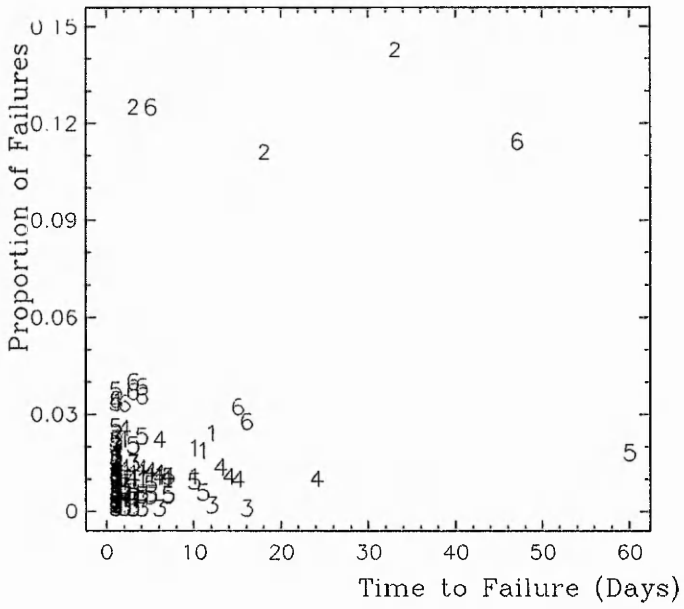


Figure 1. Plot of the Proportion of Failures against the Time to Failure for each Program Type.

GENERALISED LINEAR MODELLING

Generalised linear modelling is a commonly applied approach within the area of medical statistics when determining the proportion of subjects affected by different amounts of a drug. A standard textbook for generalised linear modelling is by McCullagh and Nelder [1]. A paper in the area of exposure to a disease using the models described here is [2]. However, generalised linear modelling has not been utilised to any great extent previously in the analysis of software reliability. The methods are used here to determine the proportion of a package of programs of a certain type which have failed by a given time.

In this paper, a generalised logistic regression model is formulated as follows. The proportion of programs of a certain type failing at a particular instant in time is calculated and a transformation is taken so that when plotted against a linear

combination of continuous variables (e.g; time since last failure, cumulative time to failure), a straight line results. The analyses carried out with the data under the model assumption are then tested for goodness of fit.

Description of Models

Three model formulations which have been applied to the described data are illustrated in figures 2, 3 and 4. Each figure shows a function of the proportion of failures/faults per day, g(p), (where p is the proportion), plotted on the vertical axis with the explanatory variable of interest, x, on the horizontal axis. The function g(p) is chosen so that the plot should be linear. In figure 2, model (1) is

$$g(p) = \beta_0 + \beta_1 x + \epsilon$$

irrespective of the type of program, which can be tested for goodness of fit. The term ϵ is an error term which explains any variation not already described by the fit of the model. Model (2) is

$$g(p) = \beta_0 + \beta_1 x + \alpha_i + \epsilon \qquad i = 1, 2, \ldots, k.$$

If the model is not significant on the factor, α_i = type, for the 6 types of program, then the plot is similar to figure 2. However if the proportion of failures/faults is affected by the type of program, then figure 3 is more appropriate. Model (3) is given by

$$g(p) = (\beta_0)_i + (\beta_1)_i x + \alpha_i + \epsilon$$

and, in this case, if the interaction between the type of program and the explanatory variable is not significant then figure 3 applies and if there is an interaction between the type of program and the explanatory variable, the plot will be similar to figure 4.

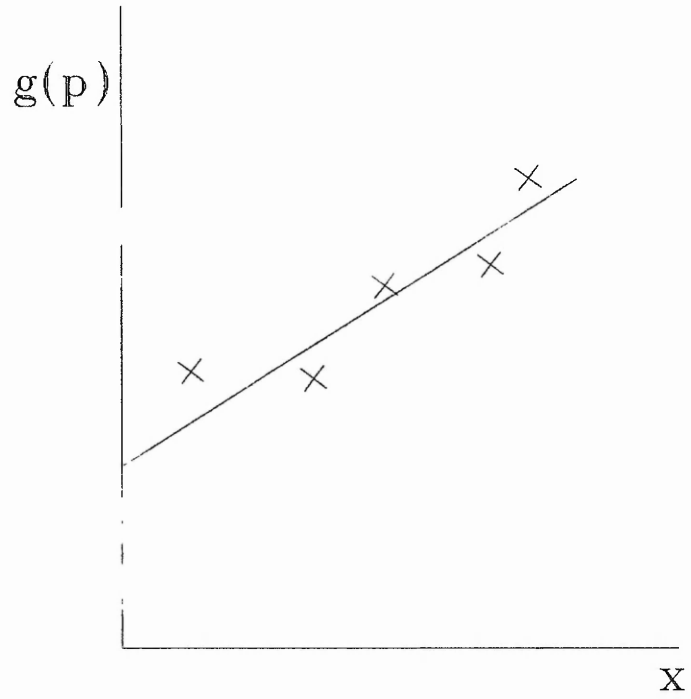


Figure 2. Plot of g(p) against x with Model (1) fitted.

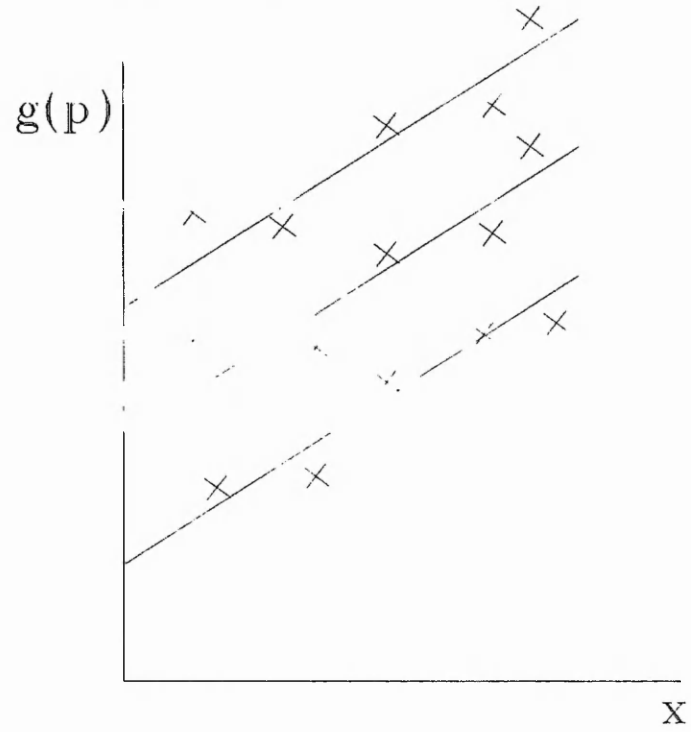


Figure 3. Plot of g(p) against x with Model (2) fitted.

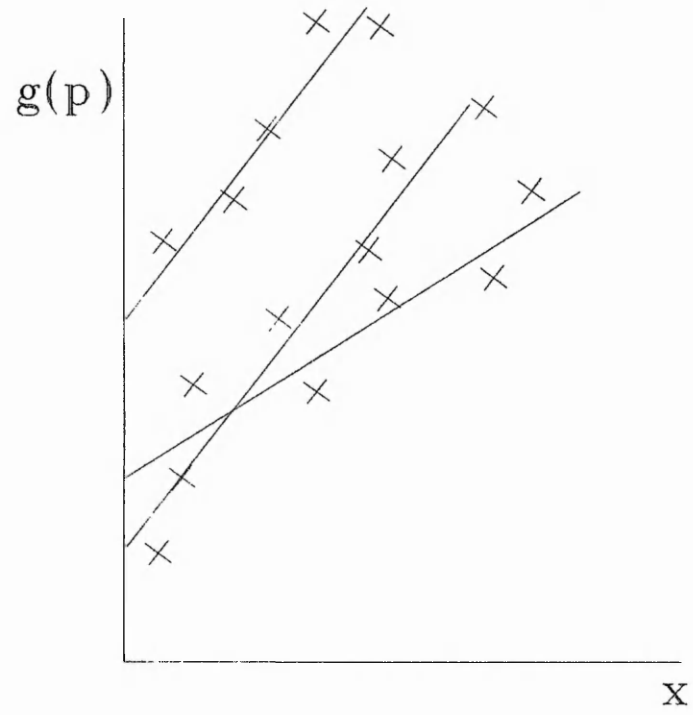


Figure 4. Plot of g(p) against x with Model (3) fitted.

Data Plots and Analysis

A number of plots of the data were drawn and subsequent analysis was carried out using the generalised linear modelling computer package GLIM, [3]. The models (1), (2), (3) were applied to proportion of failures per day, proportion of faults per day, proportion of faults and censorings per day and proportion of failures and censorings per day against cumulative time to failure and time to failure for three different types of function of the proportion. In each case there was still a lot of unexplained variation after the models were fitted. One way of reducing variation is to take a transformation of the explanatory variables such as the logarithm or the square root. A recent reliability paper which incorporated transformed data is by Follman [4]. Further analysis was carried out, with the failures only, as this analysis showed this data to have the least variation after the initial model fits. Figure 5 is one such plot of the function of the proportion of failures per day

$$g(p) = \log(p/(1-p))$$

(known as the logit function) against the logarithm of the cumulative time to failure in days.

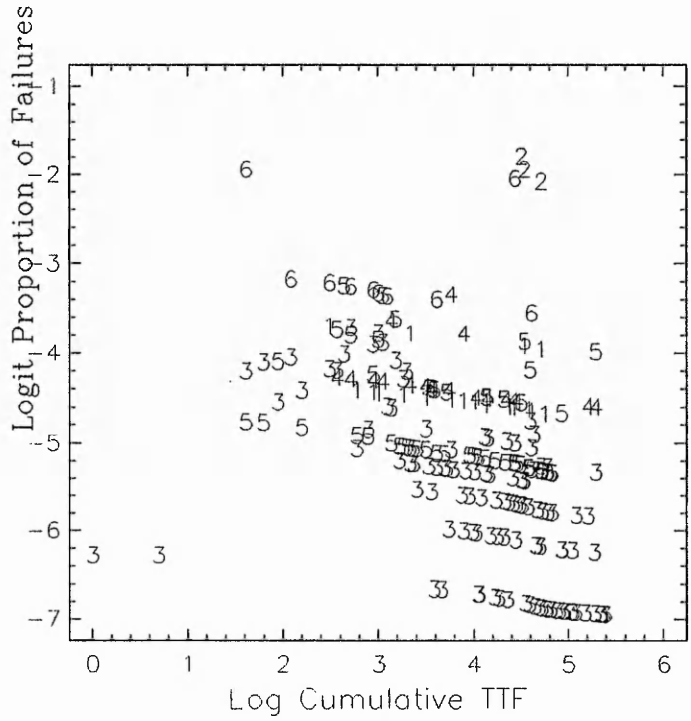


Figure 5. Plot of the Logit Proportion of Failures against Log Cumulative Time to Failure.

From the figure, it can be seen that:

- there is a difference within types of program which is mainly due to the same proportion failing per day for the duration of the project. As there was very little change in the total number of programs running per day, this difference is mainly due to the number of failures of a given type failing per day.

- there is a difference between types of program, which can be accounted for by the number of failures for each type.

- the number of failures per day is reducing over time for each type. This suggests that there is reliability growth in the data which may be expected for a software development project.

This growth can be seen more clearly if instead of the proportion of failures per day, the cumulative number of failures up to a point in time is divided by the cumulative number of programs failed with the cumulative number of failures. This cumulative proportion of failures is shown plotted against the logarithm of cumulative time in figure 6. From figure 6, it can be seen that either model (2) or (3) is applicable to the data and this can be tested with GLIM.

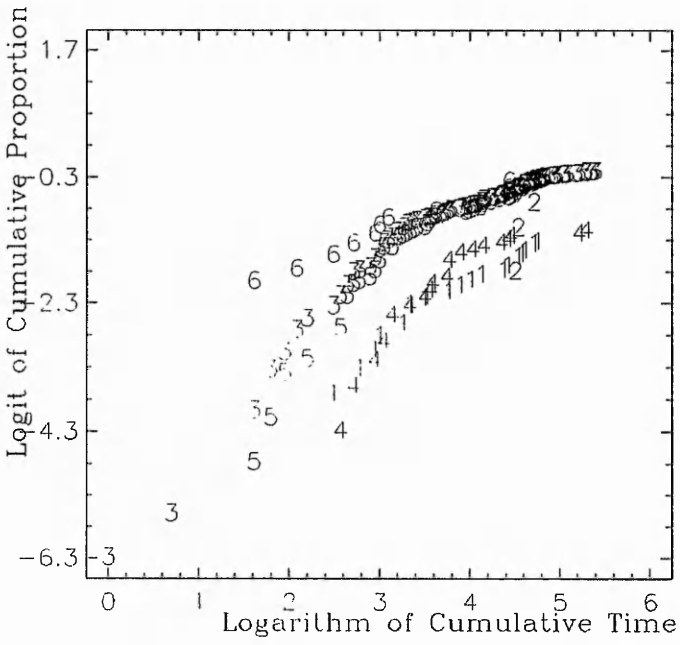


Figure 6. Plot of Logit Cumulative Proportion of Failures against Log
Cumulative Time for each Program Type.

Figure 6 is showing a plot similar to a multiple Duane plot, [5]. It has been shown by Lawless [6] that model (2) is the same as the proportional intensity formulation with Weibull intensity. The plot in figure 6 is levelling off as failures are detected and removed. A more appropriate intensity for this software dataset is the IBM model [7], also discussed by Ascher [8]. In the Ascher formulation, the intensity is given by

$$v(t) = c + a b^t$$

where $0 < b < 1$ and $c > 0$. The cumulative intensity or expected number of failures in time t is given by

$$E(N(t)) = ct + a(b^t - 1)/\log(b).$$

A plot of the Ascher model against a set of software data is shown in the later section on proportional intensity modelling. Tables 1 and 2 present the results of the GLIM analysis. To test the goodness of fit of one model over another, the difference in deviances, (explained in reference [3]), of each model is compared with the χ^2 distribution with degrees of freedom being equal to the lost degrees of freedom when fitting a more complex model. As an example of this test, for model (1) compared to model (2), the change in deviance is 480.6 on 2 degrees of freedom which is saying that the model in figure 2 is a much better fit as the tabulated value of χ^2 on 2 degrees of freedom at the 5% level is only 5.991. The comparisons of the models are given in table 1.

Assuming asymptotic normality of the parameter estimates, a simple test of whether the parameters should be included in the model is to see if the parameter estimates are within two standard errors of zero. If they are, then they do not contribute to the overall model. These nonsignificant parameters are then removed from the model and the new model parameters are estimated and tested until all nonsignificant parameters are removed. This method is known as backward stepwise regression.

For model (2), the type 3 programs were used as the baseline of comparison as the type 3 data sample size is greater than each of the other types and also the type 3 data is more central in figure 6 than the other types. The parameter estimates of type 2, type 5 and type 6 programs were not significant within model (2). A reason for this may be the lack of observations for these types. A comparison of the parameter estimates of the types of program irrespective of whether they were significant or not was carried out and this determined that types 3, 5 and 6 parameter estimates were similar, types 4 and 1 were similar but different to the other types and type 2 was on it's own. Figure 6 shows these findings.

On fitting the interaction term (between type of program and log cumulative time to failure) with model (3), the deviance is 5.17 compared to the χ^2_2 tabulated value of 5.991 which indicates that the interaction term is not significant. This is confirmed by the parameter estimates for the interaction terms in model (3) being nonsignificant. The parameter estimates for each model are given in table 2.

TABLE 1.
Goodness of Fit Estimates for Models

Fitted Model	Deviance (degrees of freedom:df)	Difference in Deviance (df)	Signif. level
Model [1]	1086.3(220)		
Model [2]	605.73(218)		
Model [1],[2]		480.6(2)	0.00
Model [3]	600.56(216)		
Model [2],[3]		5.17(2)	0.08

TABLE 2.

Estimates of Model Parameters

Fitted Model	Parameter	Estimate	Standard Error	Signif. level
Model [1]	β_0	-3.394	0.04046	0.00
,,	β_1	0.6363	0.00916	0.00
Model [2]	β_0	-3.31	0.04045	0.00
,,	β_1	0.6224	0.009147	0.00
,,	Type (1)	-1.126	0.0749	0.00
,,	Type (2)			n.s.
,,	Type (4)	-0.9258	0.07253	0.00
,,	Type (5)			n.s.
,,	Type (6)			n.s.
Model [3]	β_0	-3.299	0.04071	0.00
,,	β_1	0.62	0.009207	0.00
,,	Type (1)	-2.113	0.5369	0.00005
,,	Type (2)			n.s.
,,	Type (4)	-1.416	0.4063	0.00026
,,	Type (5)			n.s.
,,	Type (6)			n.s.
,,	$\beta_1.Type(1)$	0.2387	0.1276	0.0307
,,	$\beta_1.Type(2)$			n.s.
,,	$\beta_1.Type(4)$	0.1196	0.09715	0.1093
,,	$\beta_1.Type(5)$			n.s.
,,	$\beta_1.Type(6)$			n.s.

Relationship between Response Models and Proportional Intensity Models

As shown above, a model may be formulated as $g(p)=\beta_0+\beta_1\log(t)+\epsilon....(1)$ where the proportion p is the cumulative number of failures x up to time t divided by the total number of failures n, t is the cumulative time to failure and ϵ is a binomial error term. If g(p) is the logit function $g(p)=\log(p/(1-p))$ then (1) can be rewritten as

$$\log(x)-\log(n-x)=\beta_0+\beta_1\log(t)+\epsilon.$$

If n tends to a large constant such that n-x is very much greater than x, say k, and p is very small, the Poisson approximation may be used for the binomial response. and instead of the logit function, the log of the number of failures is obtained. On taking exponentials of each side of the equation,

$$E(x)=e^{(\beta_0+k)}t^{(\beta_1)}$$

is obtained, where E(x) is the expected number of failures. This is the formulation of the nonhomogeneous Poisson process with Weibull intensity, [9].
Using the Poisson approximation for the logit function of the proportions resulting in the log of the number of failures, (known as the log link function, [1]), then model (2) has been shown to be the proportional intensity formulation with Weibull intensity of Lawless [6].

PROPORTIONAL INTENSITY MODELLING

Lawless [6] considers the situation where a number of individuals experience repeated events, with the time of each event recorded along with covariate information. Although the individual experiences a sequence of events, the covariate information for the individual is fixed. In a software reliability modelling

context we can consider the individuals to be systems/packages/modules/languages etc; that is, a level of application in which a common baseline is reasonably thought to exist, or more commonly an application level at which data is available.
The methods discussed by Lawless are based on the proportional intensity Poisson process model. The model can be specified as

$$\lambda_x(t) = \lambda_0(t)\exp(\dot{x}\beta) \quad ....(1)$$

where t is the time from the start of observation, $\lambda_0(t)$ the baseline intensity function, x a vector of covariate values and β a vector of parameters. The formulation in [6] means that the covariates have a proportional effect on the baseline intensity function.

Relationship to Proportional Hazards Modelling

Lawless shows in section 4 of his paper the equivalence of proportional hazards modelling based on the partial likelihood of Cox, [10], and the Poisson process with unspecified baseline. However, given the different approach to the construction of the partial likelihood and the likelihood for the semiparametric Poisson process it is not possible to use standard proportional hazards modelling software for the analysis in the Poisson case. To carry out the analysis using proportional intensity models with covariates and unspecified intensity function, specific software has been written, details of which are given below.

Model Formulation

The model formulation and details of the likelihood equation are given in Lawless [6]. In particular, to obtain the β coefficients for the covariates the following set of equations have to be solved (the first partial differential of the log-likelihood),

$$\frac{\delta \log L(\beta)}{\delta \beta_r} = \sum_{i=1}^{m} n_i z_{ir} - \sum_{i=1}^{m} \{n(T_i) - n(T_{i-1})\} \frac{\sum_{l=i}^{m} z_{lr} e^{z_l \beta}}{\sum_{l=i}^{m} e^{z_l \beta}} \quad , r = 1, 2, \ldots, k \qquad (2)$$

where z_{Sr} is the r^{th} covariate value for the system s (s=1,2,3,....,m), n the number of failures for the system s, $n(T_i)-n(T_{i-1})$ the number of failures between the end of observation on system (i-1) and end of observation on system i (such that $t_1<t_2<....<t_m$).

To solve the equations in (2), a Taylor series expansion ($F(x)=\frac{\delta \log L}{\delta \beta}$) is used and then a Newton-Raphson iteration procedure; the method of scoring is applied. A program has been written in Microsoft Fortran which runs on an IBM PC to estimate the parameters of the model.

In order to run the program, a data file is created from the observed information:- the total number of failures, the total number of systems and the number of covariates, whether covariates are included in the analysis, the final observation time for each system, the covariate values and the time to failure of each system.

After reordering the data, the program proceeds with the estimation of the coefficients. Starting with initial values of zero for the coefficients, the Newton-Raphson equations are solved to provide a $\delta\beta$ value, where at the n^{th} iteration $\beta_n=\beta_{n-1}+\delta\beta$. The iteration procedure is continued until the incremental $\delta\beta$ for all the covariates is less that one thousandth of the existing β value or the number of iterations has reached 25 (indicating problems with convergence).

Upon convergence for β each coefficient is tested (using the asymptotic normality of the coefficient) to see if it is significantly different from zero. At this stage of the estimation procedure, the most non-significant covariate is dropped from the model (backwards stepwise regression) and the remaining β coefficients re-estimated. This procedure is continued until a set of significant (on a 5% two tailed test) β's are obtained. At this point desired information such as the β values, z-scores and p-values are reported in a computer output.

Having obtained a set of β coefficients, the program then calculates the base-line intensity function (using the formulation reported in 4.4 of Lawless [6]) which is then available for comparison with well known intensity models. If only two systems

exist and one binary covariate then it is possible to solve (2) directly, so that a partial check on the program may be performed. Carrying out this procedure, the same β value was obtained.
Musa et al [11] classifies many of the well known software reliability models within the framework of non-homogeneous Poisson processes. It is therefore possible when applying proportional intensity function models to compare the baseline intensity function against the intensity function for individual software reliability models.

Covariates

The covariates that can be included in the model, as with proportional hazards modelling, are obviously dependent on the context in which the data arises. However, from (our) observation on the model formulation it is noted that each system in proportional intensity modelling "plays" the same role as one failure in proportional hazards modelling. Thus there is a severe restriction on the number of covariates that may be included in any analysis if data is available only on a small number of systems. To carry out any meaningful analysis, data may have to be available on a large number of systems (Lawless in a medical example had 48 subjects). A recent paper which showed proportional intensities for software systems but did not carry out any intensity modelling is by Selby, [12]. The proportional intensity formulation software described above is applied to part of the dataset described at the beginning of the paper.

Analysis of a Software Reliability Dataset

The number of times programs and program versions were repaired was tabulated for the dataset described above. The three programs, which required most repairs were all binding Cobol files. Of the remaining 9 programs which were repaired more than 10 times, eight were module main source codes, the other being a binding Cobol file. These twelve programs out of a total of 926 were repaired 217 times out of a total of 1356 repairs. The twelve programs were all Cobol files of size greater than 9 4K blocks

of code and text of which for 10 of these, only one particular program version was repaired. The twelve programs have been analysed using Proportional Hazards Modelling (PHM), the results of which are discussed in references [13] and [14]. The plot of the cumulative time to failure for each program is shown in figure 7.

In the aforementioned analysis, the covariates; program version change, age, previous number of faults, type of use, program size and program type were tested for significance with the time metric taken as days between failures. The covariates age and previous number of faults were the most significant covariates and for each of the twelve programs analysed, the hazard decreased as the programs aged and the hazard increased with the increasing previous number of faults.

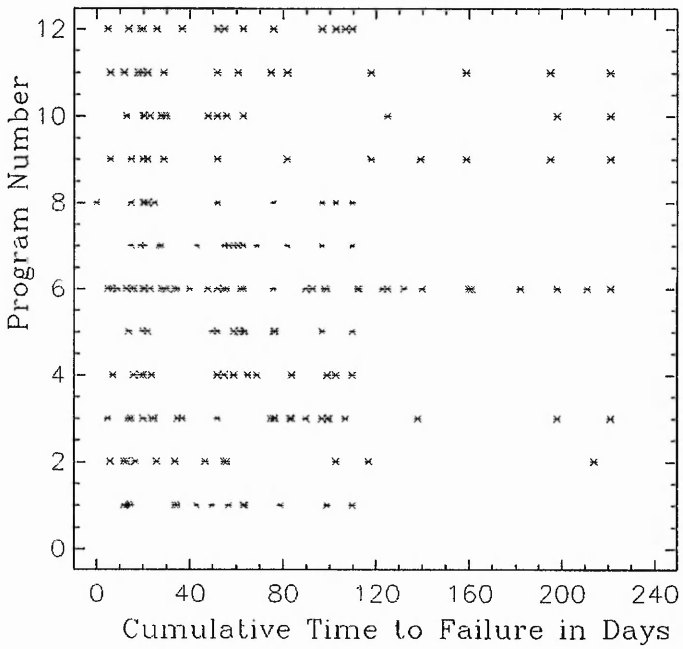


Figure 7. Plot of Cumulative Time to Failure against Program Number

An analysis of this reliability growth/decay was carried out using proportional intensity modelling with cumulative time to failure in days as the time metric for each of the twelve programs. A number of analyses were carried out and these are discussed in more detail in reference [14]. The results of one of these analyses follow.

The binary covariate, program, was used in one proportional intensity formulation. The baseline programs chosen were numbers 5 and 6 in figure 7. Program numbers 1, 3, 4, 7, 8 and 12 were shown to be not significantly different from the baseline intensity and were included into it.Programs 2, 9, 10 and 11 were significantly more reliable programs than the baseline programs although it must be stressed that censoring information was not included for programs which did not fail after the programs went into service use after 110 hours. A plot of the cumulative intensity against time with a superimposed plot of the IBM model [7] is shown in figure 8.

A problem of parameter estimation for this model is that if incorrect initial estimates of parameters are chosen then, because of the flatness of the likelihood function, a program to find the estimates will give the inflexion estimates as the solution and not the maximum likelihood estimates. The solution should be checked by making sure all the second differentials of the likelihood for the three parameters are negative.

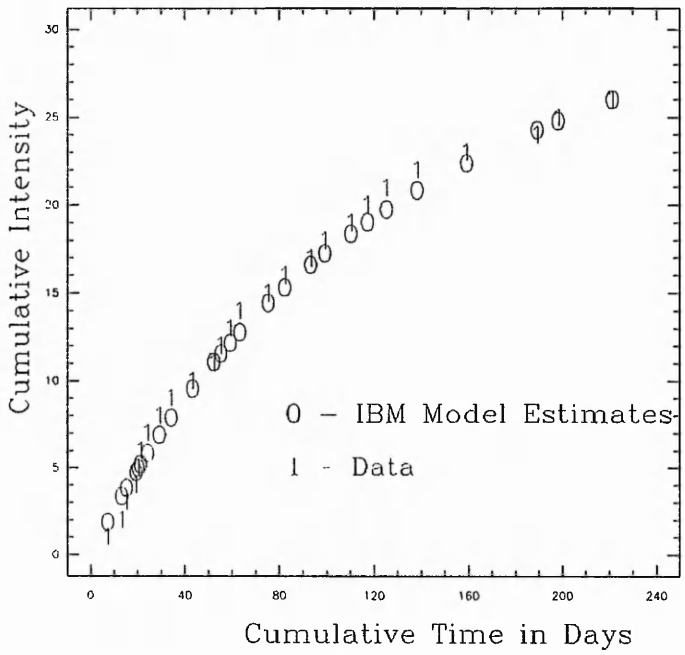


Figure 8. Plot of Observed and Expected Cumulative Intensity against Time

PROPORTIONAL HAZARDS FORMULATION OF SOFTWARE RELIABILITY MODELS

Two proportional hazards models (PHM) are formulated which incorporate a number of the well known software reliability models. References which describe the PHM approach are Kalbfleisch and Prentice [15], Lawless [16] and Cox and Oakes [17].
The application of PHM within a software context has been undertaken by Nagel and Skrivan [18], Font [19], Wightman and Bendell [20] and McCollin [13]. Also, under the Alvey Software Reliability Project, Nottingham Polytechnic undertook the analysis of a number of software reliability datasets using PHM.

The analyses of these datasets together with the PHM formulations presented in this paper, plus the other techniques discussed here will be more fully reported in McCollin [14].
The first formulation considered allows binomial type models of the exponential class (as classified by Musa et al [11]) to be incorporated within a proportional hazards framework. The exponential part of the classification refers to the failure distribution of each fault (assumed to be common) with the binomial part referring to the distribution of the number of faults experienced by time t. Examples of this type of software reliability model are Jelinski-Moranda [21] and Shooman [22].
The second formulation incorporates Poisson type models of the exponential class. In this formulation, the exponential distribution is again the assumed per fault distribution with the Poisson distribution referring to the number of faults experienced by time t. Examples of this type of model are Musa [23], Scneidewind [24], Moranda [25] and Goel-Okumoto [26]).

Binomial Type Models of the Exponential Class

The formulation of the proportional hazards model for the binomial type models of the exponential class is as follows. Following Musa et al [11], page 276, the program hazard rate for this class of model is

$$h(t_i'/t_{i-1}) = (N - i + 1)\theta$$

where N is the total number of faults present at time zero,
θ is the constant value of the hazard for each fault,
t_i' is the time from the (i-1) th failure, with t_0 =0,
t_{i-1} is the time of the (i-1) th failure.

This may be rewritten as

$$h(t_i'/t_{(i-1)}) = N\theta(1 - (i-1)/N) \ldots\ldots\ldots (1)$$

The proportional hazards formulation with the metric (t in equation (2)) taken as time since last failure is

$$h(t;z_1,\ldots,z_n)=h_0(t)\exp(\beta_1 z_1+\ldots+\beta_n z_n).....(2)$$

where β_i , i=1,...,n are the parameters of the model, z_i, i=1,...,n the values of the explanatory variables and h(t) is the baseline hazard.

Now if $h_0(t)=N\theta$, a constant, i.e., the well known exponential distribution and $z_1=loge(1-(i-1)/N)$ (with an appropriately chosen value for N); then a value of β_1 approximately equal to one obtained when PHM is applied indicates that a binomial type exponential model is appropriate for the data under investigation. The hypothesis that β_1 = 1 may be tested, as β_1 is asymptotically normal, see Tsiatis [27], Anderson and Gill [28]. Note that the explanatory variables $z_2,\ldots,z_n$ can be used to model (in the same model) the effects of other factors thought to influence the performance of the software. A classification of explanatory variables that may be applicable in software reliability modelling is given in McCollin [13].

Poisson Type Models of the Exponential Class

The formulation of the proportional hazards model for the Poisson type models of exponential type is as follows. From Musa et al page 276, the program hazard rate

$$h(t_i'/t_{(i-1)})=N\theta\exp(-\theta t_{(i-1)})\exp(-\theta t_i')$$

Let $t_{(i-1)}$ equal the time of the (i-1)th failure with N-i+1 faults left. From [11], for this model the number of faults left at $t_{(i-1)}$ is $N\exp(-\theta t_{(i-1)})$ so that

$$h(t_i'/t_{(i-1)})=(N-i+1)\theta\exp(-\theta t_i')$$

which may be written as

$$h(t_i'/t_{(i-1)})=N\theta(1-(i-1)/N)\exp(-\theta t_i')$$

In the PHM formulation (2), let

$h_0(t') = N\theta\exp(-\theta t')$ and $z_i = loge(1-(i-1)/N)$.

When applying PHM, if an estimate of β_1 approximately equal to one with a form of the baseline hazard shown above, then Poisson type exponential models are appropriate for the data under investigation.
The baseline hazard function in this formulation was first considered by Gompertz [29] in the context of actuarial studies. The hypothesis that β_1 is approximately equal to one may be tested as for the binomial class of models as the β's from PHM are asymptotically normal. The form of the baseline hazard for this software reliability model type may be investigated by plotting the logarithm of the cumulative baseline hazard against time t.

CONCLUSIONS

Two formulations of proportional hazards models have been proposed which incorporate some of the well known software reliability models. A semi-parametric proportional intensity model has been formulated and applied to a software reliability dataset. A number of generalised linear models have been formulated and applied to the software reliability dataset and it has been shown that the models can be used to show different rates of failure of programs and whether interaction exists. An appropriate intensity model for the analyses was the IBM model, [7].

REFERENCES

1. McCullagh, P. and Nelder, J.A. (1983). Generalized Linear Models. Chapman and Hall.

2. Kleinbaum, D.G. Kupper, L.L. and Chambless, L.E. (1982). Logistic Regression Analysis of Epidemiologic Data: Theory and Practice. Communications in Statistics.- Theory and Methods., 11(5), pp 485-547

3. Baker, R.J. and Nelder, J.A. (1978). The GLIM System. Release 3.77 Generalized Linear Interactive Modelling Manual. Oxford, U.K.: National Algorithms Group.

4. Follman, D.A. (1990). Modelling Failures of Intermittently Used Machines. Applied Statistics. JRSS(C). 39, No. 1. pp. 115-123.

5. Duane, J.T. (1964). Learning Curve Approach to Reliability Monitoring. IEEE Transactions. Aerospace. Vol 2.

6. Lawless, J.F. (1987). Regression Methods for Poisson Process Data. Journal of American Statistical Association. Vol 82, No 399.

7. Rosner, N. (1961). System Analysis-Non-Linear Estimation Techniques. Proc. Seventh National Symp. on Reliability and Quality Control, Institute of Radio Engineers. (now IEEE)

8. Ascher, H.E. (1968). Evaluation of Repairable System Reliability using the 'Bad-As-Old' Concept. IEEE Transactions on Reliability., R-17, pp 103-110.

9. Crow, L. (1975). On Tracking Reliability Growth. Proc. Twentieth Conference on the Design of Experiments. ARO Report 75-2, pp 741-754

10. Cox, D.R. (1972). Regression Models and Life-tables (with discussion). J. R. Statistic. Soc., (B), 30, 248-275.

11. Musa, J.D., Iannino A. and Okumoto K. (1987). Software Reliability Measurement, Prediction, Prediction. McGraw-Hill.

12. Selby, R.W. (1990). Empirically Based Analysis of Failures in Software Systems. IEEE Transactions on Reliability. Vol 39, No. 4. pp 444-454.

13. McCollin, C., Bendell, A. and Wightman D.W. (1989). Effects of Explanatory Factors on Software Reliability. Proceedings of Reliability 1989 Vol 2, pp 5Ba/1/1-11.

14. McCollin, C. (1991). Analysis Methods for Software Reliability Data. Unpublished PHD Thesis. Nottingham Polytechnic.

15. Kalbfleisch, J.D. and Prentice, R.L. (1980). The Statistical Analysis of Failure Time Data. John Wiley, New York.

16. Lawless, J.F. (1982). Statistical Models and Methods for Lifetime Data. John Wiley, New York.

17. Cox, D.R. and Oakes, D. (1984). Analysis of Survival Data. Chapman and Hall, London.

18. Nagel, P.M. and Skrivan, J.A. (1981). Software Reliability Repetitive Run Experimentation and Modelling. Rep BCS-40366, Boeing Computer Company NASA Report no. CR-165836.

19. Font, V. (1985). Une Approche de la Fiabilite des Logiciels: Modeles Classiques et Modeles Lineaire Generalise. Thesis L'Universite Paul Sabatier de Toulouse, France.

20. Wightman, D. and Bendell, A. (1986). The Practical Application of Proportional Hazards Modelling. Reliability Engineering 15 pp 29-53.

21. Jelinski, Z. and Moranda, P.M. (1972). Software Reliability Research (W. Freiberger, Editor) Statistical Computer Performance Evaluation, Academic, New York, pp. 465-484.

22. Shooman, M.L. (1972). Probabilistic Models for Software Reliability Prediction. (W. Freiburger, Editor) Statistical Computer Performance Evaluation, Academic, New York, pp. 485-502.

23. Musa, J.D. (1975) A Theory of Software Reliability and its Application. IEEE Transactions on Software Engineering. SE-1(3). pp. 312-327.

24. Schneidewind, N.F. (1975). Analysis of Error Processes in Computer Software. Proceedings 1975 International Conference on Reliable Software. Los Angeles. pp. 337-346.

25. Moranda, P.B. (1975). Predictions of Software Reliability during Debugging. Proceedings Annual Reliability and Maintainability Symposium. Washington DC. pp. 327-332.

26. Goel, A.L. and Okumoto, K. (1979). Time-Dependent Error Detection Rate Model for Software Reliability and other Performance Measures. IEEE Transactions on Reliability. R-28(3). pp. 206-211.

27. Tsiatis, A.A. (1981). A Large Sample Study of Cox's Regression Model. Annals of Statistics. 9. pp. 93-108.

28. Anderson, P.K. and Gill, R.D. (1982). Cox's Regression Model for Counting Processes: A Large Sample Study. Annals of Statistics. 10(4) pp. 1100-1120.

29. Gompertz, B. (1825). On the Nature of the Function Expressive of the Law of Human Mortality. Phil. Trans. R. Soc. (London). 115. pp.513-583.