SOFTWARE RELIABILITY PREDICTION

THESIS SUBMITTED FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

IN COMPUTER SCIENCE

Centre for Software Reliability

City University

London

EC1V 0HB

By

David R. Wright

May 2001

I grant powers of discretion to the university librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Abstract

This thesis presents some extensions to existing methods of software reliability estimation and prediction.

Firstly, we examine a technique called 'recalibration' by means of which many existing software reliability prediction algorithms assess past predictive performance in order to improve the accuracy of current reliability predictions. This existing technique for forecasting future failure times of software is already quite general. Indeed, whenever your predictions are produced in the form of time-to-failure distributions, successively as more actual failure times are observed, you can apply recalibration irrespective both of which probabilistic software reliability model and of which statistical inference technique you are using. In the current work we further generalise the recalibration method to those situations where empirical failure data take the form of failure-*counts* rather than precise inter-failure *times*. We then briefly explore how the reasoning we have used, in this extension of recalibration to the prediction of failure-count sequences, might further extend to recalibration of other representations of predicted reliability.

Secondly, the thesis contains a theoretical discussion of some modelling possibilities for improving software reliability predictions by the incorporation of disparate sources of data. There are well established techniques for forecasting the reliability of a particular software product using as data only the past failure behaviour of that software under statistically representative operational testing. However, there may sometimes be reasons for seeking improved predictive accuracy by using data of other kinds too, rather than relying on this single source of empirical evidence. Notable among these is the economic impracticability, in many cases, of obtaining sufficient, representative software failure vs. time data (from execution of the particular product in question) to determine, by inference applied to software reliability growth models, whether or not a high reliability requirement has been achieved in a particular case, prior to extensive operational use of the software in question. For example, this problem arises in particular for safety-critical systems, whose required reliability is often extremely high. An accurate reliability assessment is often required in advance of a decision

whether to release the software for actual use in the field. Another argument for attempting to determine other usable data sources for software reliability prediction is the value that would attach to rigorous empirical confirmation or refutation of any of the many existing theories and claims about what are the factors of software reliability, and how these factors may interact, in some given context. In those cases, such as some safety-critical systems, in which assessment of a high reliability level is required at an early stage, the necessary assessment is in practice often currently carried out rather informally, and often *does* claim to take account of many different types of evidence—experience of previous, similar systems; evidence of the efficacy of the development process; expert judgement, etc—to supplement the limited available data on past failure vs. time behaviour which emanates from testing of the software within a realistic usage environment. Ideally, we would like this assessment to allow all such evidence to be combined into a final numerical measure of reliability in a scientifically more rigorous way.

To address these problems, we first examine some candidate general statistical regression models used in other fields such as medicine and insurance and discuss how these might be applied to prediction of software reliability. We have here termed these models *explanatory variables regression models*. The goal here would be to investigate statistically how to *explain* differences in software failure behaviour *in terms of differences in other measured characteristics* of a number of different statistical 'individuals', or 'experimental units': We discuss the interpretation, within the software reliability context, of this statistical concept of an 'individual', with our favoured interpretation being such that a *single* statistical reliability regression model would be used to model *simultaneously* a *family of parallel series of inter-failure times* emanating from measurably different software products or from measurably different installations of a single software product. In statistical regression terms here, each one of these distinct *failure vs. time histories* would be the 'response variable' corresponding to one of these 'individuals'. The other measurable differences between these individuals would be captured in the model as *explanatory variable* values which would differ from one individual to another.

Following this discussion, we then leave general regression models to examine a slightly different theoretical approach—to essentially the same question of how to incorporate diverse data within our predictions—through an examination of models for '*un*explained' differences between individuals' failure behaviours. Here, rather than assuming the availability of putative 'explanatory variables' to distinguish our statistical individuals and 'explain' the way that their reliabilities differ, we instead use *randomness* alone to model their differences in reliability. We have termed the class of models

produced by this approach *similar products models*, meaning models in which we regard the individuals' different likely failure vs. time behaviours as initially (i.e. a priori) indistinguishable to us: Here, either we cannot (or we choose not to attempt with a formal model to) *explain* the differences between individuals' reliabilities *in terms of other metrics* applied to our individuals, but we do still expect that the 'similar products" (i.e. the individuals') reliabilities will be different from each other: We postulate the existence of a *single probability distribution* from which we may assume our individuals' true, unknown reliabilities to have all been *drawn independently* in a random fashion. We present some mathematical consequences, showing how, within such a modelling framework, *prior belief* about the distribution of reliabilities assumes great importance for model consequences. We also present some illustrative numerical results that seem to suggest that experience from previous products or environments, so represented within the model—even where very high operational dependability has been achieved in such previous cases—can only modestly improve our confidence in the reliability of a new product, or of an existing product when transferred to a new environment.

Acknowledgements

I would like to thank my supervisor Prof. Bev Littlewood for his encouragement and assistance with the material of this thesis. My thanks also go to Dr. P. Y. Chan and Mr. John Snell whose software I have re-used in fitting B-splines as part of my implementation of the modified recalibration method. I would also like to thank Prof. Norman Fenton for his support and encouragement during the writing-up period. The work has benefited considerably from numerous critical comments and suggested improvements by colleagues working at the Centre for Software Reliability at City University and from colleagues at other sites involved in several collaborative research projects with which I have been involved.

Contents

List of Tables

List of Figures

Key to Symbols, Abbreviations, and some Terminology

NB Refer also to the keys to the software reliability PFSs and failure data sets on page 84.

r_i	Number of failures of ⟨product,environment⟩ $\mathcal{A}_i$ when subjected to n_i discrete demands	§5.3.1
RV	Random Variable	9
S	U-plot, or its modified versions, when this is regarded as (the graph of) a function	57
$\mathcal{S}$	Probability law for the $\langle U_n \rangle$ process. This is determined in terms of a first PFS used to *define the residual* u_n *from* x^n (using equation (15) on p46 or one of the alternative methods later proposed), and by postulating a second, and possibly distinct PFS, as the *true probability law* for $\langle X_n \rangle$	63
$\mathcal{S}^{\mathcal{Q}}$	$\mathcal{S}$ as above, where the superscript specifies the second of the two PFSs involved in the determination of $\mathcal{S}$	63
$\mathcal{S}^*$	New probability law assumed for $\langle U_n \rangle$ by the recalibration step working from the so-far-accumulated u-plot of the raw PFS	63
SSQR	Sum of Squared Residuals (used in the context of the spline smoother for u-plots)	226
sup	Mathematical supremum or least upper bound of a subset of $\mathbb{R}$, guaranteed to exist for a set which is bounded above	[21, p18]
T	τ thought of as a random variable (whose value is not yet known)	26
T	t thought of as a random variable (whose value is not yet known)	11
t	Software execution time elapsed *since time of some preceding event* (c.f. τ) such as: (i) the most recently observed failure; or (ii) the end of the most recent interval over which some observation has been made. In particular see the following entry	11
t_n	The n^{th} *inter*-failure time of a software product	12,34
T_n	t_n thought of as a random variable (whose value is not yet known)	12,34
U	u thought of as a random variable whose value is not yet known (often subscripted by index of discrete time)	46
u	The u-residual formed by substituting a now realised continuous RV into the c.d.f with which it was previously predicted (often subscripted by index of discrete time). Or variations on this theme as proposed in this thesis. See also item $\mathcal{S}$ above . . .	46
$\mathcal{U}[a,b]$	The *uniform probability distribution* on the interval $[a,b]$	50
v_n	Function used to represent the *modified u-plot* for a discrete or mixed process as the expectation of the ordinary u-plot of a hypothetical continuous process	61
$\{w_i\}$	the vector of weights used in defining a weighted u-plot	66

Chapter 1

Introduction

1.1 Problem Area

The task of predicting the future failure behaviour of software as accurately as is possible, has, over the last three decades or so, become increasingly recognised not only to be inescapable—for project management, product reliability evaluation and certification, and software engineering research reasons—but also to differ substantially from other reliability prediction tasks, [78, 18, 55, 57, 61]. This thesis addresses the area of software reliability assessment and prediction. By the term "software reliability" in this thesis we refer to characteristics (observed or predicted) of the stochastic process of software failures versus software "execution time". This failure process may be a discrete-time process, in the case of "demand-based" systems, if we choose to measure execution time simply by counting the number of demands on the software. (See §2.1.) More usually, the failure process is modelled as a continuous time stochastic "point process" of discrete software failure "events" embedded in a continous time metric representing execution time. (See §2.2.) In either of these two cases, 'reliability' is here interpreted with an emphasis placed on the *rate* at which software failures occur, rather than on how bad are the consequences or cost of particular failures.[1] For a discussion of software incident/failure attributes see e.g. [70, 'How to Measure Incidents, Failures & Faults' on pp31-44]. Throughout this work we have in mind a software *application* of such difficulty, and hence a resulting software *system* of such size and complexity, that it is unrealistic or uneconomic to attempt to complete an exhaustive, deterministic analysis of all the potential modes and causes of software failure. Such an analysis would involve a complete classification of individual inputs and

[1] Of course we could first classify failure events according to such criteria and then use the methods treated in this thesis to analyse the rate of occurence of failures of any selected categories [94, pp348-50].

outputs according to whether or not they would constitute software failure. If this were possible then we would be in a position to think in terms of removing all faults and producing guaranteed *fault-free* software for which reliability would not be an issue. This is generally not achievable : We speak of *fault-free* software in the sense of software which could be safely assumed, usually *in advance* of extensive operational use, to *contain no faults*. This is different from properties expressed in terms such as merely: "there is a *good chance* that this software may contain no more faults"; or "as it reaches the end of its operational life, *having run over several years at many installations* it is at last beginning to look very likely *in retrospect* that this software contained no faults on the date it was released". The scenario of being able to assure *in advance* of use that a software product is fault-free does not accord with current real-life experience and is believed by many to be unlikely to do so for the foreseeable future, if ever. In the circumstances more familiar to developers and users of non-trivial software products, for which complete understanding and full knowledge of flaws in the product is beyond our reach, the case for a probabilistic analysis has been well made elsewhere. See e.g. [50, 56, 78], [72, pp324-7].

Thus the problem domain addressed in this thesis is already a well established area of applied reliability theory in which many of the same operational reliability measures (see §2.2), such as failure rate, are applicable as have been employed in the work on hardware reliability. However it should be understood that it is *software* failures which are considered here. Such failures originate from preceding human errors in the development of complex and purely logical entities. Software failures are thus of a fundamentally different kind from the failures of hardware components caused by the degradation of these through operational use. It has been shown, [69, 50, 52], both theoretically and by observation of empirical reliability data, that we must expect the different natures of these two classes of system failure to be reflected in differences in the resulting pattern of unreliability. Consequently, the specific application to software has stimulated the development of new reliability models and prediction methods having important differences from those of the longer established theory of hardware reliability. Some of the salient features of software affecting its reliability behaviour which have been identified and discussed elsewhere are: the ease with which the software components of a system can be made to be very complex logically; the ease of modification or "softness" of software; the immense diversity between different software products; the large amount of design novelty commonly found in each new piece of software; and the difficulties associated with the interface between the abstract, logical domain of formal languages, and the "real world", physical domain of the majority of software applications. It has been argued [53], that the essential features of what has been called "software reliability theory" would be more accurately encapsulated if the word

"software" was replaced by "complex design". In this case software reliability prediction methods could be taken to apply also to hardware design failures of complex hardware components.

We address the problem of stochastic modelling and prediction or forecasting of the point-process of software failure events embedded in (continuous or discrete) software *execution time*, focusing on two main extensions to existing techniques. The first of these consists of a further development of current techniques of empirical *evaluation* and *improvement* ("recalibration", [2, 59, 6]) of a predictor, involving principally an extension from continuous *inter-failure time* data to the case of the coarser "discrete" *failure-count* data. Results of numerical investigations are presented in support of the techniques proposed. The second extension addressed in this thesis is an entirely theoretical discussion of the potential for using disparate data when predicting software reliability. That is, we examine some ideas and models that might enable us to incorporate, into our predictions of software reliability, sources of data additional to the recorded behaviour of the *failure vs. execution time process* of the *single software product*, whose future reliability we now wish to predict, operating in an execution environment that is statistically representative of its present one. It is this latter source of data alone that is typically used by many existing *software reliability growth models*. We seek to explore ways to supplement this useful data source by using in addition some data of other kinds. To do this we initially examine the potential for use in the software reliability prediction context of certain already existing statistical techniques based on "proportional hazards" and "proportional intensity" regression models. It is proposed that some of these general statistical regression techniques may provide methods of incorporating supplementary data sources (in addition to merely past failure-vs.-time behaviour) into our software reliability predictions under some circumstances. We present arguments for and against the viability of such an approach, distinguishing some different potential applications. We finish by leaving the context of pre-existing general statistical regression models to propose a more specific model for *un*explained random variation between the reliabilities of 'similar' software products or execution environments. We examine how this model might be used to improve the reliability predictions of any particular single member of such a 'family of similar ⟨product, environment⟩ pairs'.

1.2 Layout of Thesis

Chapter 2: Previous Work

§2.1 briefly mentions a simpler modelling context sometimes used in which the software's execution is measured as of a sequence of discrete 'demands'; rather than using the more common continuous-time model for the cumulative amount of execution to which a software product has been subject.

§§2.2,2.3 contain a review of existing techniques for forecasting a process of software failures in continuous software execution time, using records of the early portion of that process as the sole source of data for prediction of subsequent behaviour. The first of these two sections contains a brief survey of some existing models of such a failure process. Some of the basic terminology, used in later sections, relating to reliability of software is introduced and defined here. §2.3 reviews common methods of deriving a reliability prediction algorithm from such a model.

§2.4 discusses techniques for validation of a predictor against a particular failure data set. For reasons which are explained, assessment of predictive performance of each predictive technique assumes a particular significance when it is *software* whose failures are being predicted, and such assessment of any particular software reliability prediction technique is unlikely ever to be completed to a point where that prediction technique can thereafter justifiably be regarded as validated for general use in predicting software failure processes. Particular attention has in the past been paid to evaluation of a succession of short term predictions for the degree to which they possess the property of being "well calibrated" in a sense which can be expressed as a property of a certain sequence of generalised residuals denoted $\langle u_n \rangle$ and used to form the "u-plot". These residuals form the basis of the recalibration methods reviewed in §2.5 and later extended in Chapter 3.

§§2.6 & 2.7 are included as additional background material required for the work contained in §5.2. §2.6 contains a literature survey of some previous work addressing the problem of how data on the past failure-vs.-time behaviour, of the particular software item whose future failure behaviour is required to be predicted, might be supplemented by other data sources of various kinds in the pursuit of more accurate reliability predictions. §2.7 discusses various statistical results and methods based on two general regression models. These have been applied in several diverse contexts in which "survival time" or "lifetime analysis" has been considered appropriate. Such random events and processes range from insurance claims in actuarial work to patient responses following different treatments in medicine. The discussion in §2.7 centers around work found in the statistical, rather than software engineering, literature.

§2.8 considers some existing work on the prediction of failure-count sequences. Failure-count data

amounts to *less* information available for input to a predictor than the more typical assumption of the availability of complete inter-failure time data from the past portion of the point process of software failures in continuous execution time.

Chapter 3: Extension to Discrete Predictions

Chapter 3 further pursues the problem introduced in §2.8 beginning with some observations on difficulties encountered when attempting to carry over, essentially unchanged, to the failure-count context the methods used previously for complete past inter-failure time information. §3.1 discusses difficulties with the conversion of parametric reliability models and their related predictors to the failure-count case.

§§3.2 & 3.3 return to the concept of calibration of a predictor and discuss the associated tool of the u-plot. The notion of an ideal or 'true' predictor is introduced as a way of examining the calibration behaviour that a 'best possibly performing' predictor might exhibit. Some difficulties with u-plot based approaches are revealed in the transition to the failure-count case, when we attempt to define u-plots and construct recalibrated reliability predictors from predictors that concentrate positive predictive probability at discrete values of the variable to be predicted.

As a contribution towards overcoming these problems, §§3.4 & 3.5 consider alternative definitions of the $\langle u_n \rangle$ sequence and of the u-plot which are shown to have desirable properties for general predictors of quantities possessing discrete (or mixed) predictive distributions including, as one special case in particular, predictors based on software failure counts. §3.6 focuses on the potential of such plots for use as tools for obtaining improved, recalibrated predictors. In particular, in §3.6, the use of weighting, and of smoothing under gradient-constraints, are discussed, leading to the proposal of further modifications to the recalibration mechanism. These last modifications are not specific to recalibration of *discrete* predictions and can be applied also to the case of prediction of continuously distributed quantities, such as the original software inter-failure times. §3.7 reviews the extensions of the recalibration idea developed in Chapter 3 and discusses the extent to which these suggest methods of recalibrating other kinds of predictions.

Chapter 4: Data Analysis

Chapter 4 reports a data analysis in prediction from failure-count data using the techniques developed in Chapter 3. The techniques were tested using both simulated and real software failure-vs.-time data. Graphical output is collected in Appendix A.

Chapter 5: Other Sources of Data

Taking the original case of complete information on a single failure-vs.-execution time history as a kind of reference or standard data assumption for the software reliability prediction problem, the work in Chapter 5 can then be seen as progress 'in the opposite direction' from that taken in §2.8 and further pursued in Chapter 3. Thus, Chapter 5 considers the possibility of *augmenting*—rather than depleting by the transition to failure counts—the information entailed by the much treated data assumption of a single sequence of past inter-failure times. After a motivating problem description for Chapter 5 given in §5.1, §5.2 picks up the ideas introduced in §§2.6,2.7 and discusses the question of whether and how it may be feasible to identify and rigorously validate models for software reliability prediction which are able to produce more accurate reliability predictions by incorporating supplementary data sources ("explanatory variables"), rather than being restricted, as sole source of data, to the use of the so-far-observed part of the process of failure events along the execution time axis of the single software item whose reliability is to be predicted. For this purpose of incorporating *additional* sources of data *supplementary* to the inter-failure time sequence, the potential merits of the two related classes of general statistical regression model, *proportional hazards models* and *proportional intensity Poisson process models*, are discussed. These have been used in other contexts for carrying out regression of event times on to associated explanatory variables. At several places throughout §5.2—and in particular in §§5.2.1.4 & 5.2.3—some reasons for the popularity of the common approach of modelling only a single stochastic point process of failures in the case of software are reviewed and some of the obstacles to doing anything more than this are described. In §5.2.1 a number of alternative ways of conceptualising software reliability prediction problems in terms of the standard structures entailed by each of the two classes of general statistical regression models are identified, and it is concluded that the obstacles mentioned apply to greater or lesser extents for each of these, the most favourable application being that of §5.2.1.3 in which different software "operational environments" of a single software product play the role of "individuals" in the terminology of the statistical models.

§5.3 explores of a different approach to incorporating other data into the reliability predictions for a software product and its operating environment. In this case we concentrate on empirical reliability data emanating from other software products or execution environments which are believed to be 'similar' to the product and environment in question, whilst acknowledging that they will have reliabilities which will differ from it. Here our term 'similar' can be thought of as a kind of 'indifference' between distinct failure vs. time processes, whose precise meaning lies in the (conditional) independence assumptions assumed to represent the reliability variation between the different failure

processes. In contrast to the approach taken in §5.2, there is no assumption now that we have access to observable metrics or characteristics of the different ⟨product, environment⟩ pairs of our family, and with which we can attempt to model any *explanation for* or *correlate with* their differences in reliability behaviour.

Chapters 6 & 7: Conclusions and Suggestions for Further Work

Chapter 6 contains the main conclusions and Chapter 7 contains some suggestions for taking some of the ideas further in the future.

Appendices

Appendix A contains the graphical results discussed in Chapter 4. Appendix B contains some mathematical details suppressed for clarity of exposition in earlier chapters.

Chapter 2

Previous Work

2.1 Discrete Demand-Based Reliability Models

Perhaps the simplest mathematical framework in which to model reliability is the demand-based one in which 'time' is represented simply as a count of the number of executions, or demands made on an item of equipment or, in the case of this thesis, an item of software. With each in a succession of demands, we may assume that the software executes the task required of it either successfully or unsuccessfully, the latter case being termed a failure. This is not the most appropriate model for all software, many examples of which are better conceived as operating continuously, but it does have the advantage of a kind of simplicity in that the mathematics of the real line is not required for the model of the time metric. The simplest model for this kind of failure process is the *Bernoulli Trials* process, where the probability p of failure is the same for each demand, and the outcomes of demands which are distinct are always assumed to be statistically independent. See e.g. [22, §4.9],[28, Ch. VI,VIII]. We make use of this simple model of successive executions in §5.3 of this thesis, but otherwise concentrate on point-process models of the failure vs. execution time process. It should be born in mind throughout §§2.2 and 2.3 that most of the discussion of parametric inference and forecasting systems, though couched largely in terms of the mathematically more involved point-process case, can easily be modified (and in fact simplified) to apply to the discrete demand-based model framework.

2.2 Continuous Time Point-Process Reliability Models

Throughout most of this thesis the underlying model with which we are concerned is the slightly more complex 'continuous time' analogue of the above. This consists of a one dimensional stochastic point process. For a formal mathematical definition, the standard one may be assumed. This involves a *sample space*, Ω, consisting of a set of *elementary outcomes*, ω, each of which, in the case of a point process model, consists of a single *realisation* of the process. Thus each $\omega \in \Omega$ is itself some fully specified deterministic arrangement of finitely or countably many[1] indistinguishable points along a half line. A family, Σ, of subsets, E, of Ω, including all straightforwardly defined subsets, each have

Figure 1: One realisation, ω, of a point process, $\langle \Omega, \Sigma, \mathbf{P} \rangle$

an associated probability, $\mathbf{P}[E]$. Σ is assumed to have the closure properties of a *sigma algebra* of sets. The set function, $\mathbf{P}$, satisfies the countable-additivity and other axioms of a probability measure, [82]. In this way a formal framework may be built within which events of interest concerning the arrangement, ω, of indistinguishable points along the half line have coherently defined probabilities of occurrence. Subject to the satisfaction of the axioms mentioned, it follows that any function, $X : \Omega \rightarrow \mathbb{R}$, which is measurable with respect to Σ, is a random variable (RV) having associated c.d.f. $F^X(x) = \mathbf{P}[\{\omega : X(\omega) \leq x\}]$ (or $\mathbf{P}[X \leq x]$ for brevity, following the usual convention), expectation $\mathbf{E}[X] = \int_\Omega X \, d\mathbf{P}$ etc.[2] It is also possible, using Radon-Nikodym derivatives, to define conditional probabilities and expectations in the standard ways, [45, 27], having the standard properties.

The process of formal definition sketched in the previous paragraph produces the general tool of a stochastic point process. In the context of software reliability modelling, the half line is interpreted as an axis of cumulative software *execution time*, τ, measured from time, $\tau = 0$, on the left. See [68, p170] and [70, pp45-53] for further discussion of some more precise practical definitions of this metric. The points of each process realisation, ω, are the locations in cumulative execution time at which successive software failures occur when a particular copy of a single item of software is executed at a given installation site[3] using a given sequence of inputs to the software. The possibility of modification to the software during its execution is not excluded, being in fact central

[1] For the purposes of modelling software failures we could also without loss of realism tighten this to exclude accumulation points of the set of failure times.

[2] Here a general Lebesgue integral of the Σ-measurable function X is used, [21].

[3] Of course, it may be possible to pool the sequences of failures occurring from copies of the same software running at multiple sites into one single point process, provided it is possible to formulate a single aggregated execution time metric with respect to which all of these events can be temporally located.

to the software reliability *growth* models described below.

Of the twenty or so, [17, 71, 26], [78, Chapters 9–11] existing point process models which have been applied to the software reliability prediction problem, the majority are parametric families of stochastic processes. In terms of the above general framework this means that a parameterised family, $\{\mathbf{P}_\theta : \theta \in \Theta\}$, of probability functions is specified by the model. Here θ is a parameter vector of typical dimension about 1–3. For a given probability law $\mathbf{P}$ of a process, i.e. for a given value of the parameter vector θ in the case of a parametric model, several theoretical quantities determined[4] by $\mathbf{P}$ are of particular interest from the point of view of reliability modelling. Four of these are $R(t;\tau|\mathcal{G})$, $f(t;\tau|\mathcal{G})$, $z(\tau|\mathcal{G}_\tau)$, and $M(\tau)$, the *reliability function*, *p.d.f. of time-to-next-failure*, *conditional hazard rate*, and *process mean function*, respectively. Before explaining the meaning and relevance of these quantities, the purpose of the "$\mathcal{G}$" on the right hand side of the conditioning sign will be briefly explained in the following paragraph:

The purpose here is to emphasise that there does exist a rigorous underlying theory which guarantees under very general circumstances[5] (a) the existence of conditional probabilities, conditional expectations, conditional stochastic rates etc and (b) the satisfaction by these quantities of certain general properties, such as that expressed by equations of the form $\mathbf{E}[\mathbf{E}[Y|X]] = \mathbf{E}[Y]$, which will be employed in Chapter 3. Thus, in terms of the abstract model, $(\Omega, \Sigma, \mathbf{P})$, of a point process sketched on p9, $\mathcal{G}$ is a *sub-sigma algebra* of Σ (i.e. a subfamily of the family of sets comprising Σ, such that $\mathcal{G}$ itself satisfies the closure axioms required for it to qualify as a sigma algebra of sets). The interpretation of such a $\mathcal{G}$ is that $\mathcal{G}$ defines a set of possible conditions of partial knowledge about a realisation, ω, of the process. Exact knowledge of ω is equivalent to a complete description of the number and positions of all points along the half line. For each $E \in \Sigma$ (i.e. each $E \subseteq \Omega$ for which $\mathbf{P}[E]$ is meaningful), a positive or negative answer to the question, "Is $\omega \in E$?", represents an element of knowledge about the realisation, ω, of the process. A complete set of answers to all of the corresponding questions for every $E \in \mathcal{G}$ is the abstract representation of the result of a certain kind of partial observation of the arrangement, ω, of points along the half line. Thus, $\mathcal{G}$ consists of those sets $E \in \Sigma$ whose outcome (either $\omega \in E$ or $\omega \notin E$) will be determined by the kind of restricted observation concerned. Each potential state of affairs observed can be thought of as equivalent to a boolean-valued function whose arguments are the member sets $E \in \mathcal{G}$. For example, the act of observing the positions of the earliest (left-most) n points, for some fixed n, however long

[4] For some *conditional* probabilities, rates and expectations, this is a slight simplification since such a quantity is not *uniquely* determined by $\mathbf{P}$. It is close to unique in that any pair of alternative candidates for the conditional quantity (having the same random variable or event on the LHS of the conditioning, "|", sign) are identical "$\mathbf{P}$-*almost everywhere*", i.e. identical *with* $\mathbf{P}$-*probability* 1, [45, 27].

[5] We do not always assume that random variables are of the continuous or discrete classes in later chapters.

this may take, constitutes a particular act of partial observation of a realisation ω. This process of observation would be represented by a sub-sigma algebra $\mathcal{G}$ defined as the family precisely of those "binary" questions about any realisation ω, for which $\mathbf{P}$ is defined and whose answer, "yes" or "no", is logically entailed[6] by knowledge of the positions of the earliest n points of the process (i.e. $\mathcal{G}$ is the family of those subsets $E \in \Sigma$ which represent such questions). Another common partial observation of a realisation is "observation of the process up till time τ", which has associated sigma algebra denoted $\mathcal{G}_\tau$ above.[7] Similarly, observation solely of the value of one random variable, X (i.e. a Σ-measurable function $X : \Omega \to \mathbb{R}$), as the conditioning knowledge, can be expressed in the same form by setting $\mathcal{G}$ equal to $\mathcal{G}(X)$ the sigma algebra generated by X, [27, p160].[8] This notion generalises without difficulty to multivariate random variables, $X : \Omega \to \mathbb{R}^k$, and further to random vectors of random dimension, $X : \Omega \to \bigcup_{k=0}^{\infty} \mathbb{R}^k$, as discussed in [16, Chapter 1]. Having introduced this notation for conditional quantities, we follow common convention in dropping it in most cases in what follows and either adopting looser descriptions of the "conditioning partial knowledge" on the right hand side of the "|" sign or omitting this altogether if it is felt to be clear from context. Sometimes we have adopted the practice of shortening the length of equations on the page by moving the conditioning "$|\mathcal{G}$" or "$|X$" down to form a subscript of the "$\mathbf{P}$" or "$\mathbf{E}$" to which it applies (e.g. see p58, including footnote 19). However, despite the looseness of the notation often used, see [27, pp160–2], [21, pp139–40], [82, pp117-24] for theory assuring the existence and cooperative behaviour of such "conditional" quantities under very general assumptions.

Returning to the quantities defined above, the reliability function, $R(t;\tau|\mathcal{G})$, is defined as the conditional probability $\mathbf{P}\left[C(\tau,\tau+t]=0 \;\middle|\; \mathcal{G}\right]$, where CI denotes the random variable which counts the number of events occurring in the interval, I, and the usual notation is employed for open, closed and half open intervals. Here $\mathcal{G}$ denotes some kind of observation of the behaviour of the realisation prior to (to the left of) time τ, such observation often taking the form of complete knowledge of event times in $[0,\tau]$ or perhaps knowledge only of some event counts as discussed in Chapter 3. The p.d.f., $f(t;\tau|\mathcal{G})$, of time-to-next-failure is the density function corresponding to this reliability function,

$$\begin{aligned} f(t;\tau|\mathcal{G}) &= -\frac{\partial}{\partial t} R(t;\tau|\mathcal{G}) \\ &= \lim_{\Delta t \to 0+} \frac{1}{\Delta t} \mathbf{P}\left[C(\tau,\tau+t]=0 \wedge C(\tau+t,\tau+t+\Delta t]>0 \;\middle|\; \mathcal{G} \right] \end{aligned}$$

[6] It is conventional to assume that the rules used to interpret "logically entailed" here will result in $\mathcal{G}$ having the closure properties of a sigma algebra—Just as it is conventional to assume that any probability function $\mathbf{P}$ is defined on a sigma algebra.

[7] Thus, informally $\mathbf{P}[E|\mathcal{G}_\tau]$ denotes the conditional probability of an event E as this probability is assessed by an observer whose original beliefs held at time $\tau = 0$ about the process corresponded to the probability law $\mathbf{P}$ and who is "now standing at time τ" having fully observed the times of intervening events.

[8] In this case it is usual to substitute "$|X)$" for "$|\mathcal{G})$"

where this limit exists (i.e. where, as is usually the case for the software reliability growth models discussed here, the time-to-next-failure is, given $\mathcal{G}$, continuously distributed). A commonly treated special case of these quantities, reliability function $R(t;\tau|\mathcal{G})$ and time-to-next-failure density $f(t;\tau|\mathcal{G})$, is that in which the conditioning observation[9] $\mathcal{G}$ represents observation *up to the time* $\tau = \sum_{i=1}^{n-1} t_i$ *of the* $n{-}1^{th}$ *failure.* Here the indices, i or n, when appended to the time-to-next-failure variable t denote that the time is measured from a failure time (or from start of execution $\tau = 0$ in the case of t_1) to the time of the following failure, i.e. the variable t_n is the n^{th} *inter*-failure time of the software. This case is often equivalently referred to in terms of *one-step-ahead* prediction of the *inter-failure time process* $\langle T_n \rangle$. See also §2.4.2, and Figure 2 on p34.

The term, conditional hazard rate, is used to denote the stochastic failure rate at time τ of software whose past failure behaviour has been fully observed,

$$\begin{aligned} z(\tau|\mathcal{G}_\tau) &= f(0;\tau|\mathcal{G}_\tau) \\ &= \lim_{\Delta t \to 0+} \frac{1}{\Delta t} \mathbf{P}\left[C(\tau, \tau+\Delta t] > 0 \;\middle|\; \mathcal{G}_\tau\right]. \end{aligned}$$

This quantity may also be referred to in what follows as the conditional program hazard rate or the software failure rate, the implication being, unless otherwise stated, a rate at time τ conditional on complete observation of failure behaviour, $\mathcal{G}_\tau$, prior to τ. $z(\tau|\mathcal{G}_\tau)$ is, of course, itself a real, non-negative valued, *stochastic* process (but not in the case of NHPP failure processes—see below). The failure process mean function is simply the unconditional expected number of failures prior to time, τ, i.e. $M(\tau) = \mathbf{E}[C[0,\tau]] = \int_\Omega C[0,\tau]\, d\mathbf{P}$. Note that $M(\tau)$ is a deterministic function of τ: From the definition of conditional expectation, for fixed τ, t and $\mathcal{G}$, it follows that $R(t;\tau|\mathcal{G})$ and $f(t;\tau|\mathcal{G})$ are $\mathcal{G}$-measurable random variables[10] , $z(\tau|\mathcal{G}_\tau)$ is a $\mathcal{G}_\tau$-measurable random variable, and $M(\tau)$ is a constant. *In the NHPP case only*, we may assume the same for the conditional failure rate function—In which case, we speak of the *process intensity function* and use the symbol λ : $z(\tau|\mathcal{G}_\tau) = \lambda(\tau)$—See §2.7. The mean (MTTF) $\mathbf{E}[t;\tau|\mathcal{G}]$[11] of the time-to-next-failure distribution, frequently used to express the reliability of hardware components, may also be of interest in expressing software reliability. This quantity can be expressed in terms of either the reliability function or the p.d.f. of the time to next failure distribution,

$$\begin{aligned} \mathbf{E}[t;\tau|\mathcal{G}] &= \int_0^\infty t f(t;\tau|\mathcal{G})\, dt \\ &= \int_0^\infty R(t;\tau|\mathcal{G})\, dt\ . \end{aligned} \tag{1}$$

[9] In this case $\mathcal{G}$ will be the sigma algebra generated by the random vector $\langle T_1, T_2, \ldots, T_{n-1} \rangle$.

[10] If a random variable is $\mathcal{G}$-measurable, the interpretation is that its realised value is determined by the knowledge denoted by the sigma algebra, $\mathcal{G}$.

[11] Almost always $\mathcal{G} = \mathcal{G}_\tau$ being intended

It should be born in mind, however, that this mean alone is insufficient for the determination of failure probabilities for many software reliability models, since an exponential assumption for the time-to-next-failure distribution may be very inaccurate. For example, if the time-to-next-failure distribution is of the Pareto family, as is the case for some models, then the mean time to next failure may be infinite even in cases where the reliability is very poor, [50]. In such cases the median time to failure, denoted $m(\tau|\mathcal{G})$, satisfying $R\big(m(\tau|\mathcal{G}); \tau \mid \mathcal{G}\big) = \frac{1}{2}$ may be preferred as a measure of reliability at time τ. This point about mean vs. median time to next failure exemplifies a more general one : There is no single agreed numerical 'reliability measure' of a software product at a given time[12] τ. This is an important consideration in many situations in which the apparent comparative reliability level achieved will depend on how that reliability is expressed [63].

Existing software reliability point-process models can be classified in various ways into categories such as the *exponential order statistic* models, [74]. The exponential order statistic models are based on the idea of a population of faults, in the software, of fixed or parametrically distributed population size. This population size, N say, if assumed fixed, forms one component of the vector θ. If the population size is assumed random, then the parameter vector of its assumed distribution family—frequently for example the mean, λ, of a Poisson distribution—replaces N as the component of θ. These exponential order statistic models then go on to assume that the individual software faults in this population will each first cause a software failure after independently, exponentially distributed cumulative execution times. We may avoid the unwanted complication of repeat manifestation of a single fault by assuming either: (i) that each fault is immediately and correctly removed on first manifestation (with no further fault being introduced in the process) so that subsequent occurrence is impossible ; or (ii) that repeat occurrences of a single fault are correctly diagnosed as such prior to fitting the exponential order statistic model, so that the times of occurrence after the first may be removed[13] from the data-set. The assumption of exponentiality, along with an assumption, in the earliest models, of time-to-failure distributions identical over all faults, have subsequently been relaxed to form new models, e.g. the Littlewood model [53]. In various of these models, the distribution of individual fault manifestation rates over this fault population may be specified either as a deterministic parametric sequence of rates, as a random sample from a single parametric fault-rate distribution, or more generally as the points in a general stochastic fault-rate process. Some of these methods of relaxing the earliest and most simplistic model assumptions which appear at first sight to be distinct, actually turn out to be equivalent, [74], i.e. to result in identical parametric families, $\{\mathbf{P}_\theta : \theta \in \Theta\}$, of point process probability functions. Briefly for example, two equivalent

[12] And this remains the case even supposing we agree to focus on a single probability model of the failure process.
[13] with the entailed loss of statistical information

methods of achieving the same[14] effective extension of model assumptions are: (a) to relax the exponentiality assumption above, and (b) to retain the exponentiality assumption but to replace the assumption of identical manifestation time distributions for all faults in the population, by the assumption that the individual fault rates form an independent, identically distributed random sample from some fault-rate distribution. This second technique is taken one step further in [74], by considering the sequence of *fault rates* themselves to arise as a parameterised stochastic point process.

There are also models, see e.g. [40, 62], which do not represent the software explicitly in terms of its fault population, but consider rather the overall *software* failure rate from the outset, directly modelling the effect of maintenance actions, following each software failure, by means of a random sequence of software failure rates which come into effect sequentially after each successive software failure. In the case of these models, the parameter vector, θ, of the point process of failures is simply identical to the parameter vector of the random sequence of software failure rates. The time to next failure, given the current software failure rate, is assumed to be conditionally exponentially distributed. Also, point process models from the well known NHPP class have been used for software failure process modelling. By means of such models, which can be completely specified simply by stating a parameterised, non-decreasing mean function, $M(\tau;\theta)$ on the half line, $[0,\infty)$, it is effectively assumed that there is no instantaneous effect on program failure rate following the event of software failure (and any subsequent corrective maintenance action). In fact these models are characterised by the assumption that the reliability function, $R(t,\tau|\mathcal{G})$ is a deterministic quantity depending on τ and t only, and not on $\mathcal{G}$, so that $R(t,\tau|\mathcal{G}) = R(t,\tau)$, (provided of course that $\mathcal{G}$ contains information only on the locations of points outside the interval $(\tau,\tau+t]$). Thus for an NHPP model the numbers and patterns of events occurring in disjoint intervals are independent random variables. However, these models too, or at least those of them for which $M(t)$ is bounded by a constant, can be derived in another way such that this effective assumption is not immediately apparent [78, pp268–70]. Note also the comments on p47 concerning the importance of the distinction between a *parametric process model* and a derived *prediction system* in interpreting the term "independent" when applied to events or variables.

[14] In fact the second is more restrictive than the first since the class of distributions which can be produced as a mixture of exponentials is actually a restricted class, [27, p416].

2.3 Forecasting Systems

2.3.1 Observation and Statistical Inference

Statistical inference procedures have been applied to software reliability models in order to provide solutions to the software reliability prediction problem. The most commonly applied procedures: *maximum likelihood "plug-in"* and *Bayesian* prediction are illustrated in [53, 16, 10, 1, 78]. These procedures can be interpreted in terms of the formal model described on p9. Briefly, whatever kind of observation is undertaken, it can be thought of abstractly as a function, $Y : \Omega \rightarrow \mathcal{Y}$ (i.e. a random variable in the usual case where Y is a numeric-valued function), which maps distinct points ω onto a common point y in the observation space $\mathcal{Y}$ whenever these points ω are not distinguished from each other by the particular kind of observation concerned. For example, if observation ceases at time τ, and two realisations, ω_1, ω_2 are identical in the region of the half line to the left of τ, then we will have $Y(\omega_1) = Y(\omega_2)$. For each θ, there is induced[15] a probability function $\mathbf{P}_\theta^{\mathcal{Y}}$ defined for subsets of $\mathcal{Y}$ by $\mathbf{P}_\theta^{\mathcal{Y}}[S] = \mathbf{P}_\theta[Y^{-1}(S)]$.

2.3.2 Maximum Likelihood Inference

A likelihood function is then constructed as a *density*, $L(\theta, y) = \frac{\partial \mathbf{P}_\theta^{\mathcal{Y}}}{\partial \nu}(y)$, with respect to some standard *dominating measure*[16] ν, defined on the observation space $\mathcal{Y}$. In all cases found in the literature listed in the bibliography, and also throughout the body of this thesis itself, ν may be taken to be based on either Lebesgue measure for a continuous observation space (which is appropriate when the observation takes the form of a vector of inter-failure times), or in the case of an observation which is a vector of counts (Chapter 3), a counting measure will do for ν. In either case, and for many other imaginable hybrid observations, the very general Radon-Nikodym Theorem [21, pp139–46] guarantees the existence of the likelihood function, L. A detailed exposition of the procedure for the former case is given in [16, Chapter 1]. Using the well known maximum likelihood technique, the maximum likelihood estimate $\widehat{\theta}$ may be chosen to maximise L, once having observed the value, y, of Y, and then probabilities of the form $\mathbf{P}_{\hat{\theta}}[\cdot|\mathcal{G}]$ used for purposes of prediction based on the observation Y. (Here, if used for predicting future events[17], $\mathcal{G}$ denotes a sigma algebra which at

[15] i.e. determined from the first probability function $\mathbf{P}_\theta$ by the particular form Y of the observation

[16] a *dominating measure* is a measure with respect to which all of the probability measures $\{\mathbf{P}_\theta^{\mathcal{Y}} : \theta \in \Theta\}$ are *absolutely continuous* [21, p139], or, in other words, a measure with respect to which the desired measure-ratios (those suggested by the term *density of* $\mathbf{P}_\theta^{\mathcal{Y}}$ *with respect to* ν) may be taken (without the possibility of zero-divide). I.e. we are requiring that $\nu(S) > 0$ for all subsets $S \subseteq \mathcal{Y}$ to which at least one of the $\{\mathbf{P}_\theta^{\mathcal{Y}} : \theta \in \Theta\}$ assigns a positive probability $\mathbf{P}_\theta^{\mathcal{Y}}[S] > 0$.

[17] The unconditional probabilities $\mathbf{P}_\theta[\cdot]$ may remain of interest, even after $Y = y$ has been observed, for inference purposes such as producing confidence intervals and hypothesis testing concerning the value of θ.

least contains the sigma algebra $\mathcal{G}(Y)$ generated by the function, Y. When $\mathcal{G} = \mathcal{G}(Y)$, the notation "$|y)$" will frequently be used to mean the realised value of this predictor once the value $Y = y$ has been observed.)

When the data Y takes the form of complete observation of the software failure point process prior to current time, τ, then the likelihood function obtained from any one of the parametric models can often be quite a complicated function of the model parameter vector, θ. Hence, whether a maximum likelihood or a Bayesian approach is used for the inference and prediction problem, computer assisted numerical techniques are usually found to be necessary in order to construct predictions, see e.g. [10].

2.3.3 Bayesian Inference

In the case of Bayesian, rather than maximum likelihood, statistical analysis and prediction, an additional model component is constructed in the form of a prior distribution for the parameter Θ, regarded now as a random variable. Thus a probability measure $\mathbf{Q}$, say, is assumed to represent the initial chances of Θ taking values from a suitable space Θ. The net result of this refinement is effectively equivalent to a direct assumption of a probability law $\mathbf{P}$ for the failure process. Here $\mathbf{P}$ is given by the mixture distribution

$$\mathbf{P}[E] = \int_{\Theta} \mathbf{P}_{\theta}[E]\, d\mathbf{Q} \tag{2}$$

i.e. we now have a single derived probability law $\mathbf{P}$ for the process rather than a parametric family. In practice the analysis and computations involved in forming reliability predictions, which as before are based on conditional probability distributions given full or partial observation of the part of the failure process prior to time τ (representing "now"), are frequently carried out by means of an intermediate stage consisting of the derivation of a posterior probability distribution $\mathbf{Q}[\cdot\,|y]$ for Θ given the data y. This relies on Bayes' Rule in a form such as

$$h(\theta|y) = k(y)L(\theta, y)\frac{\partial \mathbf{Q}}{\partial v}(\theta),$$

where $h(\cdot\,|y)$ is a density for the conditional probability measure $A \mapsto \mathbf{Q}[A|y]$ with respect to the dominating measure v on the space Θ. Alternatively, directly in terms of posterior probabilities, there is the form

$$\mathbf{Q}[A|y] = k(y) \int_{A} L(\theta, y)\, d\mathbf{Q},$$

for the posterior probability that θ occupies any measurable subset A of Θ. In these two forms of Bayes' Rule k is a normalizing factor independent of θ chosen such that $\mathbf{Q}[\Theta|y] = 1$. Having determined this posterior distribution for the parameter Θ, predictive probabilities based on the conditional distribution $\mathbf{P}[\cdot|y]$ can be calculated by mixing the corresponding θ-parameterised quantities over the set Θ of possible θ values so that, where $\mathbf{P}_{\hat{\theta}}[\cdot|y]$ would be used in an ML plug-in based forecasting system[18], we now have, under the Bayesian inference system, $k(y)\int_\Theta \mathbf{P}_\theta[\cdot|y]\,d\mathbf{Q}[\theta|y]$ instead. Bayesian further conditioned probabilities $\mathbf{P}[\cdot|\mathcal{G}]$ (i.e. those conditioned on further hypothetical information, such as would be a predictive reliability function or hazard rate function) can be obtained similarly, where here, as in the corresponding ML case, it must be required (for the prediction to be sensible) that the conditioning knowledge represented by $\mathcal{G}$ is consistent with Y having been observed: $\mathcal{G}(Y) \subseteq \mathcal{G}$. In the Bayesian inference case, these would replace the simple ML plug-in conditional probabilities of the form $\mathbf{P}_{\hat{\theta}(y)}[\cdot|\mathcal{G}]$. For example, when the information $\mathcal{G}$ represents the observation of Y, plus the further hypothetical conditioning event E, which may be observed subsequently to have occurred, then in the Bayesian case having observed y, we would use predictive conditional probabilities such as

$$\mathbf{P}[A|E, Y{=}y] = \frac{\int_\Theta \mathbf{P}_\theta[A \cap E|Y{=}y]\,d\mathbf{Q}[\theta|y]}{\int_\Theta \mathbf{P}_\theta[E|Y{=}y]\,d\mathbf{Q}[\theta|y]}$$

in place of the ML version $\mathbf{P}_{\hat{\theta}(y)}[A|E] = \mathbf{P}_{\hat{\theta}(y)}[A \cap E] \,/\, \mathbf{P}_{\hat{\theta}(y)}[E]$.

2.3.4 Computational Considerations

Computationally the maximum likelihood method of obtaining predictive distributions reduces to a (sometimes constrained) optimisation of the likelihood function which may often be simplified by parameter transformations of θ, partial differentiation with respect to components of the (transformed) parameter vector, and transformations of the likelihood function itself such as log likelihood. Obtaining Bayesian based predictive probabilities and distributions involves integrations over the parameter space, Θ.

2.3.5 How to Express Reliability

Predictive probabilities or densities of various random variables may be desired depending on what form the reliability prediction is to take, i.e. what future events are required to be predicted. Typically these will include *predictive* versions of the failure process quantities introduced on p10. By this we mean that these quantities will be defined as for the original parametric model but in terms

[18] $\widehat{\theta}$ being a function of the observation y, through the maximization of $L(\theta, y)$ as explained on p15

now of the distribution $\mathbf{P}_{\hat{\theta}}$, in the ML case, or $\mathbf{P}$ from equation (2), in the Bayesian case. The appropriate one of these two replaces $\mathbf{P}_{\theta}$ as the process probability law used in the definitions of these quantities. There is a narrow definition of 'reliability' in terms of the 'reliability function', i.e. as the probability, under specified conditions, that there will be *no* further failures during a period τ of further execution. In practice, the term reliability is often used rather more loosely. Statistics such as mean and median may be used to attempt to summarize important characteristics of predictive distributions, though care needs to be exercised [50] in the interpretation of these. E.g. mean or median times to next failure are often used as measures of reliability. It is common to speak loosely of system reliability using terms such as: 'a 10 to the minus 5 system'; or 'a system with MTTF[19] 10^7'. Here the units implied would typically be *failures/hour* and *hours* of execution time[20]. It is of course possible to give a precise meaning to these terms, so some care is required in comparing reliabilities. Within a rigorous modelling framework, statements of comparison such as: 'System $\mathcal{A}$ is more reliable than system $\mathcal{B}$'; or 'data set I would yield a higher reliability estimate for this system than would data set II' would need to be made more precise by explaining exactly which mathematical reliability measure they refer to. See §5.3.6 and the comparison of alternative *stopping rules* contained in [64] for further discussion of some situations in which care is needed in distinguishing alternative strict interpretations of the general qualitative concept of 'reliability'.

2.3.6 How Far Ahead to Predict

Related to this question of what precise form of quantity—whether reliability function, median time to next failure, hazard rate, etc—derived from predictive distributions is of interest to the user of a forecasting system, another important issue is the question of *how far ahead* to predict. Predictive versions of quantities such as $R(t;\tau|\mathcal{G}_s)$ where $s < \tau$ may be of interest at to an "observer standing at time s" for example.

2.4 Predictive Quality

2.4.1 The Trustworthiness of Software Reliability Predictions

There are a number of evident reasons for expecting the prediction of software failure behaviour to be a non-trivial problem. These include the diversity of software. The use of the single generic term "software" conceals the potentially infinite variety of the logical components of the solutions to a large

[19] mean-time-to-next-failure
[20] or perhaps, in the discrete *demand-count* setup discussed in §2.1, *failures/demand* and *demands*, respectively

number of vastly different practical problems, developed by differing organizations, using different methods and tools, executing on a range of different physical computers, running under different operating systems, and so on .... Many items from the entire family of all software components differ from each other in both application area and also in the level of difficulty or scale of each problem, and hence in the level of human organisation required to develop the software solution. Also, the fact that software is abstract and (other than for trivial programs) its development involves significant and problematic human-communication[21] tasks means that the laws (if any) which govern its behaviour cannot be expected necessarily to be as regular and stable as those which enable the relatively accurate prediction of the failure behaviour of many physical systems. For these reasons it is essential in the case of *software* reliability prediction not to blindly trust the predictions emanating from any particular predictive technique when it is first applied to any new software component. Instead, the best statistical techniques which are available should be incorporated in methods of *assessing the quality of the predictions* produced by each prediction method when applied to each software component.

2.4.2 Repeated Short-Term Prediction : Prequential Forecasting Systems

In this thesis, as in previous work [2, 58], we concentrate on the quality of *short-term* predictions of software failure-vs.-time behaviour. The techniques used in Chapters 3 & 4 centre on the assessment of the statistical distribution of the vector, $\langle u_n \rangle$, of residuals obtained by substituting an observed quantity in the c.d.f. of its own recent predictive distribution. Following Dawid, [20], we can define a Prequential Forecasting System (PFS) for a random process $\langle X_n \rangle$, $n = 1, 2, \ldots$, by means of functions $\langle F_n^X(x_n; x^{n-1}) \rangle$. The function F_n^X is to be interpreted as the one-step-ahead predictive c.d.f. of X_n as its first argument, having observed the realisation $X^{n-1} = x^{n-1}$ of the process so far[22]. Clearly the function $x \mapsto F_n^X(x; x^{n-1})$ must be a valid c.d.f. (monotonic non-decreasing, right-continuous, with limits 0 and 1 respectively as $x \to -\infty$ or $+\infty$) defined by the PFS for all possible values of x^{n-1} for each n. §3.2 provides further details. We focus on the $\langle u_n \rangle$ residual sequence given in terms of a process realisation by

$$u_n = F_n^X(x_n; x^{n-1}).$$

This, together with the use of *Prequential Likelihood*, including *log Prequential Likelihood Ratio* plots, is in accord with Dawid's "Prequential Principle" of concentrating attention on the comparison between *prediction* and later-observed *observation* of the predicted quantity, when applying a

[21] "Instead of imagining that our main task is to instruct a *computer* what to do, let us concentrate rather on explaining to *human beings* what we want a computer to do." [46]

[22] Here, introducing the notation $\zeta^m \stackrel{\text{def}}{=} (\zeta_1, \ldots, \zeta_m)$ for vectors.

recursive prediction method. Thus we have deliberately avoided being overly distracted by questions of *goodness of fit* and *parameter estimation error* for the *parametric model* on which the forecasting system may be based, concentrating rather on the assessment and improvement of the *performance* of the *forecasting system* as a whole when applied to a particular sequence of observations [20, 2].

2.5 Recalibration

In the case of recursive prediction involving the kind of predict→observe→predict... cycle, discussed in [20], it has been found possible (see [10, 59, 11, 6]) to dynamically modify the predictions in order to attempt to correct certain kinds of systematic inaccuracy which may become apparent with the accumulation of data on the comparison of prediction with subsequent observation. The basic idea used has been to attempt to modify (recalibrate) the predictive distribution of as yet unobserved quantities in a way which is aimed directly at the correction of certain observed kinds of consistent anomaly in the statistical behaviour of the sample of $\langle u_n \rangle$ values forming the so-far-observed portion of the $\langle u_n \rangle$ sequence mentioned in §2.4. Thus it is hoped that the statistical properties of the new $\langle u_n \rangle$ sequence following this modification[23] will be closer to the theoretical ideal statistical distribution of these generalised residuals and that this, in turn, ought to mean that the predictive distributions themselves will have been improved in the process. Further details of this previous work are provided prior to novel extensions of the methods in §3.2. We follow the terminology of [10, 59, 11, 6] in distinguishing a *recalibrated* PFS from the *raw* PFS on which it is based. The term *raw* can usually be thought of as referring to prediction without the use of recalibration. However, it has been found to be demonstrably useful in some previous work to consider recalibrating twice, i.e. adding a second (identical) recalibration step to a PFS which already incorporates one. Therefore, it is strictly more correct sometimes to interpret a *raw* PFS as one to which we could, and perhaps later will, add a recalibration step (or further recalibration step).

2.6 Incorporating Further Data

We turn now to the possibility of incorporating additional information, other than the so-far-observed part of our single point process of failures, in our predictor for the future part of that process. Previous work on explaining and predicting software reliability behaviour in terms of various kinds of other data is briefly surveyed here. Several diverse sources of other data have been considered but we suggest adopting the general terminology of "explanatory variables" denoted $\langle x_s \rangle$ for these in

[23] i.e. the equivalent $\langle u_n \rangle$ sequence for the "recalibrated" forecasting system

an attempt to force these various examples of related work, to the extent that this may be possible, into the general framework introduced in §2.7. The following is a selection from published work in this area.

2.6.1 Correlation Between An Operating System's Failure Rate and Other Time-Varying Operating System Parameters

Iyer and Rossetti [35, 88] investigate reliability variations of a single installation of the VM/SP operating system running in a genuine application environment on an IBM 3081 over a 14 month period. The explanatory variables here are thus time varying, $x = x(\tau)$ for a sample consisting of one sole "individual"[24] (the entire operating system software). All the explanatory variables are internal measures of system activity logged automatically by the system itself. The data is not listed. The analysis is crude in the sense that it treats each explanatory variable in isolation, and does not attempt to model the combined effect of more than one explanatory variable (by, for example, removing the effect of x_1 on the failure rate before investigating any x_2-effect). This is perhaps a wise initial approach to a single data set. Some details of the procedure have been omitted. As far as the details are specified, it appears that the time axis is divided into successive five minute time intervals. For each candidate explanatory variable, say x_1, this set of time intervals is then partitioned into seven subsets according to which of seven strata contains the mean value of $x_1(\tau)$ over a closely preceding time interval of length one hour. Then the proportion of the 5 minute time intervals in each stratum which contain an operating system failure is calculated. The correlation coefficient is used as a measure of strength of relationship between the vector of seven x_1 values and the vector of proportions of the corresponding sets of 5 minute intervals which do contain failure. The same is done independently for the other explanatory variables, x_s. For each separate explanatory variable, x_s, the variation between the seven values of this proportion is typically up to one order of magnitude. The results for one particular explanatory variable, OVERHEAD, appear to be more statistically significant than those for the others. Also, this variable exhibits a greater proportionate variation (about three orders of magnitude) between its different strata. However, on closer inspection it transpires that the authors have used elapsed time, (i.e. calendar time or "real" time) as their fundamental time metric with respect to which all time intervals and reliability observations are defined. They mention that OVERHEAD is a good approximation to execution time for the operating system software. The strong correlation obtained between the value of OVERHEAD and the empirical failure probability now appears merely as confirmation that operating system

[24] This standard statistical term, synonymous with "experimental unit", is explained in §2.7.2

software failure rates are more stable if defined in terms of operating system software execution time rather than in terms of calendar time. This causes one to speculate as to whether the high correlation of empirical failure rate, vs. the *calendar time* metric, with some other explanatory variables (e.g. PAGEIN and SIO) might also be largely explainable in terms of a statistical association between the values of these variables and the current rate of accumulation of operating system execution time with respect to calendar time.

An alternative analysis of this or similar data, were it available, would convert all inter-failure times to operating system execution time, using as good an estimate of this as could be obtained from the OVERHEAD values logged. Then it would be possible to experiment with fitting and checking a PIM[25] model using time-varying explanatory variables $x_s(\tau)$, see [83, 15]. Such an analysis, if it proved successful in the positive sense, would have the advantage of representing reliability as a function of the combined influences of a number of explanatory variables. That is it would attempt to separate the influence of x_1 when looking for an x_2–effect on reliability, and might in theory even be used to examine so called "interactions" between the effects of x_1 and x_2 on reliability.

2.6.2 PHM Regression of *Inter*-failure Times Onto Fault Characterisics

Wightman and Bendell [93] discuss the application of PHM[26] to *inter*-failure times with faults identified as "individuals". Thus they choose t_i rather than τ_i, in the notation which will be used in Chapter 3, as response variable. The authors' approach does not fit straightforwardly within the normal PHM scheme of regressing lifetimes onto individuals' characteristics since the inter-failure time preceding a fault is not actually equal to the true time for which that fault has been "in hazard" (i.e. subject to the risk of manifestation and removal). Also, they suggest using the number of previously observed failures due to other faults as an explanatory variable, which is likewise an unusual way of applying PHM in which the observed survival time of a fault compared to that of other faults, which would normally be thought of as a *response* variable in such an analysis, has become incorporated in the definition of the fault's *explanatory* characteristics. This analysis seems to be based on the idea of defining the sample to be "the first n faults which will occur" rather than an identified set of n known faults, each having explanatory variable values which are measurable in advance of the response variable, time of occurrence. One consequence of this is that any explanatory variables describing characteristics of a fault—with the exception of those such as the rank of the fault in order of occurrence or the times-to-occurrence of preceding faults, which variables are *defined* partially *in terms of response variables*—cannot be known in advance

[25] see §2.7.2, p26
[26] see §2.7.2, p26

and therefore cannot be fed into a predictor since (quite apart from the fact that it is not generally known what faults are present in software) it is certainly very unlikely to be known *which* of those present will give rise to the *next* software failure. So this analysis is different in kind from the other examples of PHM discussed below in §2.7 (such as the medical examples): There is no predefined population of individuals whose explanatory variable values can be stated in advance of observation of the response variable values.

2.6.3 Software Reliability and Exercise Frequencies of Code Containing Each Fault

Andrew [4] extends existing reliability models by including frequencies, obtained by code instrumentation, at which a suitably defined unit of code is exercised. These models are compared to existing models using simulated failure and exercise frequency data.

2.6.4 Software Reliability and Software Module Transistion Rates

Several Markov chain models have been proposed for the effects on software reliability of the movement of the locus of code execution between different software sub-modules. For example, Littlewood [51] obtains asymptotic expressions for parameters relating to reliability using a semi-Markov structural model for transfer of control between submodules of a software product. The data input required for these predictions consists firstly of past failure information for modules, along with some estimate of module interface failure rates. Secondly, control transition rates are required for transitions of execution from one module to another. These would most probably be obtained by code instrumentation. [73] describes a model for the affect on reliability of interactions and stress effects from multiple, simultaneous users transiting sequentially between multiple, shared software modules.

2.7 Two General Regression Models

2.7.1 Generality and Mathematical/Analytical Tractability

Here we focus on candidate probabilistic models for extending the sources of data input to stochastic point process models, introducing two different regression models found in the statistical literature. These are proposed as models for generating software reliability predictions incorporating additional data which supplements the previously recorded failure vs. time behaviour for the specific software

under study. Neither of these models was designed with this application in mind, nor even specifically for engineering reliability analyses. Both are actually general purpose models, more accurately described as classes of models since they are flexible as to the number of different parameters employed: This is a matter for experimentation and choice in a given application—so called *model identification*. These two families of models are very much related and will be distinguished here by the names "Proportional Hazards Model" (PHM), and "Proportional Intensity Model" (PIM). Concerning terminology, note that the term "Proportional Hazards Model" is widely used, although occasionally with a wider meaning than that given below (see e.g. [66, Chapter 9]). The term "Proportional Intensity Model", [49], is less widely used. Both of these model families are suited to general regression analyses, taking advantage of linearity and other mathematically simplifying assumptions appropriate to the exploration of multiple and often little understood relationships or postulated relationahips.

It should be noted that in the case of PHM, the application to software reliability analysis is not new, [93, 79], though in some ways, because of the fundamental stochastic-*process* nature of software failure behaviour, the application of PIM to software reliability would be more natural and less problematic from the point of view of parameter estimation and model fitting, provided that sufficient suitable data had been collected.

2.7.2 Description of PHM and PIM Models

Before describing these models in detail, the elements common to both of them, which characterise them as regression models and suggest the application to software reliability, amongst several other applications, are identified as follows. Both models expect equivalent data from each member of a sample of "individuals" drawn from some population. This data should take the form of values for each of a finite set of attributes which all individuals have in common. (For example, in a medical application, these individuals could be patients suffering from a particular complaint, [85].) Each attribute is possessed by every individual in the population, but to a level or measure which may vary from one individual to another on some numerical (or other) scale. (See [66] for a discussion of various scales of measurement and classification, and [30] for a more developed discussion specifically in the context of software.) One scalar or sometimes vector attribute is singled out and termed the "response" variable or "dependent" variable for the purpose of applying the model. In applications this variable tends to be one whose value (a) can be thought of as being caused, or at least influenced, by the combination of values for the remaining attributes of the same individual, and (b) is desired to be predicted for certain individuals based on observation or measurement of the remaining attribute

values for the individual in question. (As usual, the discovery of a systematic relationship by fitting such a model to data is not necessarily interpreted as confirmation of a specific causal hypothesis.) In an application the response variable is typically not observable until some point later in time than the time of observation of the other attributes, for this individual. The remaining attributes, apart from the response variable, are collectively termed the "independent" variables, "covariates" or "explanatory" variables. If there are more than one of these then the term "multiple regression" is often used. (Returning to the medical example, explanatory variables for each patient might include patient age at initial onset of symptoms, smoker/non-smoker, whether or not there is a known family history of the complaint, income bracket, etc, as well as perhaps the result of a biochemical test conducted on all patients, and/or aspects of the treatment regime such as the administration and dosage of drugs. The response variable might be survival time, time until remission from the illness, or one of several other possibilities.) A regression model is often, but not necessarily always, defined in such a way as to specify either a complete probability distribution or at least the first two moments of a distribution for the response variable of any given individual in terms of the covariate values for that individual. This is the case for both PHM and PIM models. Of course, this information is not uniquely specified for models of a parametric class until a single model from the class has been identified by specifying exact values for all model parameters. In certain classes of regression models called "linear regression", or perhaps "generalised linear regression" models (see [66]), the covariates enter into the distribution of the response variable only via a single weighted sum of the covariate values. The weights in this sum (the vector β below) are constant over the individuals of the population, or at least over disjoint sub-populations termed *strata*. The weights, β, are termed regression coefficients. The PHM and PIM models are of this kind. Such a model assumption is seen to be less restrictive than it might at first appear, on remembering that the attributes incorporated as explanatory variables in the formal statistical regression model can, if required, be produced from *transformations* of one or more primary, directly measurable attributes. In linear regression models the regression coefficients play the role of model parameters to be estimated from data. Note that in general, a regression model may incorporate other parameters to be estimated besides the regression coefficients.

In the type of application to software reliability discussed here, observed reliability data (a vector of inter-failure times) are selected to play the role of response variable. The individuals are then the executable software items, segments or components (perhaps down to the level of individual faults in software) to which these failures may be attributed—Or rather, since it is well understood that observed reliability is not actually a property of the software alone but depends

also on the environment or execution profile under which the software is operated, the individuals will in general be ⟨*software component, usage environment*⟩ pairs. In the most general application of these statistical regression models to software reliability, it is envisaged that the covariates could be any measurable or estimable attributes of either the software itself (including its originating development process), or its execution environment, or some attribute of the dynamic interaction between these two (excluding reliability which has already been nominated as response variable of the model). This is certainly not to suggest that it would be a profitable exercise to attempt to fit such models for software reliability using just any choice of attributes as covariates. Also, because of the general purpose nature of these models, assurance that they have any useful application to the software reliability prediction problem could only be obtained by demonstrating an improvement in predictive quality on several real data sets. Methods of validating the models in a particular application are discussed in §2.7.3.

To fill in the remaining details of the PHM and PIM model assumptions: In the case of the PHM model, the response variable, τ, is a non-negative scalar. The model specifies a probability distribution for τ, described in this case in terms of its hazard rate function,

$$h(\tau) = \lim_{\Delta\tau \to 0} \frac{1}{\Delta\tau} \mathbf{P}[\tau \leq \mathrm{T} \leq \tau + \Delta\tau | \mathrm{T} \geq \tau] \ ,$$

by the equation

$$h(\tau) = e^{x'\beta} h_0(\tau) \tag{3}$$

where x is a $p \times 1$ vector of covariate values, β is a $p \times 1$ vector of regression parameters and h_0 is a hazard rate function known as the "baseline hazard rate". Following the usual convention, 'T', the upper-case τ (distinguished from 'T', the upper-case t), is used to denote a random variable. The interpretation of the PHM model as described here is that no more than one event per individual may be observed. See [Pijnenburg 1991] for an "Additive Hazards Model" analogue of PHM.

The PIM model makes use of a *point process response variable*, $\{\tau_j \ : \ j = 1, 2, \ldots\}$, where $0 \leq \tau_1 \leq \tau_2 \leq \ldots$. This process is assumed to be distributed as a non-homogeneous Poisson process (NHPP)[27], [5], with process intensity function,

$$\begin{aligned} \lambda(\tau) &= \lim_{\Delta\tau \to 0} \frac{1}{\Delta\tau} \mathbf{P}[\tau \leq \mathrm{T}_j \leq \tau + \Delta\tau, \text{ for some } j] \\ &= \lim_{\Delta\tau \to 0} \frac{1}{\Delta\tau} \mathbf{P}[\tau \leq \mathrm{T}_j \leq \tau + \Delta\tau | \mathrm{T}_{j-1} \leq \tau \wedge \mathrm{T}_j \geq \tau], \quad \text{for every } j, \end{aligned}$$

given by

$$\lambda(\tau) = e^{x'\beta} \lambda_0(\tau) \tag{4}$$

[27] see p14

Recall from p14 that the event $\{\tau \leq \mathrm{T}_j \leq \tau + \Delta\tau, \text{ for some } j\}$ here is independent of all earlier or subsequent events.

When fitting either of these models to a sample of n individuals $\{\mathrm{A}_i\}$, indexed by i, the baseline function, h_0 or λ_0, is the same for all individuals, as are the regression parameters, β, of course. Every other term will be indexed by i, i.e. : response variables, τ_i for PHM and $\{\tau_{i,j} \ : \ j = 1, 2, \ldots\}$ for PIM; covariate vector x_i ; and hazard rate $h_i(\tau) = e^{x_i'\beta} h_0(\tau)$ for PHM; process intensity, $\lambda_i(\tau) = e^{x_i'\beta}\lambda_0(\tau)$ for PIM. (Under certain circumstances, such as model checking procedures, the population may be divided into strata between which either the baseline function or possibly the regression parameters are allowed to vary. See p31.)

Fitting procedures which allow the baseline function to be completely unrestricted exist for both model families. Indeed one of the advantages of these models is the degree of separation which can be achieved between the estimation of the regression parameters, β, and the estimation of the baseline hazard rate or process intensity in this unrestricted-baseline case (the so called semi-parametric version of the model). Alternatively, a best fit for the baseline within some parametric family can be obtained. An obvious initial selection of such parametric families for the baseline of the PIM models is provided by those parametric NHPP models which have previously been applied as software reliability models. For example the Weibull, lognormal, negative exponential, and Pareto intensity models amongst others. (See e.g. [74, 78, 48].)

At this point, it is worth mentioning a possible confusion with regard to the naming of PIM and PHM models in terms of the parametric family of their baselines : Suppose a model of time-to-first-manifestation of each member of the population of faults initially present in an item of software employs hazard rate function h_0. Then the associated NHPP model formed as in [74] (by mixing over a Poisson(μ) distributed fault population size) has *intensity* function μf_0 where f_0 is the *pdf*, not the hazard rate function, of the distribution indicated by h_0, i.e. $f_0(\tau) = h_0(\tau)e^{-\int_0^t h_0(u)\,du}$. For example, if the name "Weibull" is given to a model based on time-to-first-manifestation distribution for each fault having hazard rate of the form $h_0(\tau) = ab\tau^{b-1}$, then there is an NHPP model, associated in the above sense, which has intensity function $\lambda(\tau) = \mu ab\tau^{b-1}e^{-a\tau^b}$, and which could therefore reasonably be called the NHPP version of "the Weibull model". Potential terminological confusion arises with the completely different but perfectly legitimate and interesting NHPP model based on the "Weibull intensity" function, $\lambda(\tau) = ab\tau^{b-1}$.

The PIM version of the Weibull model having intensity $\lambda_i(\tau) = e^{x_i'\beta}ab\tau^{b-1}$ is worthy of particular attention for another reason. It lies at the intersection of the PIM and AFT classes of explanatory-variables-NHPP models. Here, by the AFT class is meant the "Accelerated Failure Time" class

of NHPP regression models for which the explanatory variables have the effect of accelerating or decelerating the process of failure occurrence and reliability evolution against time,

$$\lambda_i(\tau) = e^{x_i'\beta}\lambda_0(e^{x_i'\beta}\tau) \tag{5}$$

Models of the AFT class are in general (i.e. for cases other than the Weibull intensity) less tractable for statistical inference than PIM models. There is a corresponding class of AFT survival time models [38] which bears the same relationship to the PHM class as the above AFT NHPP class bears to the PIM class.

The following remarks about flexibility within the PIM model assumptions apply equally to the PHM model. The proportionality assumption

$$\lambda(\tau) = g(x,\beta)\lambda_0(\tau)$$

is fundamental to the PIM models. However the form of the scaling factor, $g(x,\beta) = e^{x'\beta} = \prod_{s=1}^{p} e^{x_s\beta_s}$, does allow some flexibility through transformation of the explanatory variables. Individual explanatory variables can be transformed to give for example a power law $x_s^{\beta_s}$ in place of $e^{x_s\beta_s}$ by a logarithmic transformation. Further, the assumption of independence for the intensity-scaling effects of the different explanatory variables, $g(x,\beta) = \prod_{s=1}^{p} g_s(x_s,\beta_s)$, can be circumvented while remaining within the PIM model structure by introducing artificial new explanatory variables representing more complex explanatory variable "interactions" such as for example, in the case $p = 2$, $g(x,\beta) = e^{x_1\beta_1}e^{x_2\beta_2}e^{x_{12}\beta_{12}}$, where we may define the third explanatory variable as a bivariate function of the first two, for example $x_{12} = x_1x_2$. Model extensions of this kind are discussed more systematically in the context of generalised linear models, [66]. Such a model extension may be suggested by inspection of suitably defined "residuals" (see the final paragraphs of §2.7.3) which result from fitting an initial model which does not represent interactions.

Models with unrestricted baseline function are termed "semi-parametric" to distinguish them from "fully parametric" versions in which the baseline is restricted to some parametric family. One, not too serious, disadvantage of fully parametric versions is that some of the separation between inference concerning the baseline and inference concerning the regression parameters is lost by the parametric restriction on the form of the baseline, [49].

A refinement which can be applied to either model class without seriously impeding the model-fitting and model-checking techniques is the use of time dependent covariates, $x = x(\tau)$, in equation (1) or (2), [38, Chapter 5], [15, 83]. Clearly there may be potential in this case for detecting a relationship between explanatory variables and response variable using a smaller sample of

individuals—observations on the time-variability of x for a single individual supplementing the information obtained from the variability of x between individuals of a sample.

The pros and cons of these two model families for software reliability modelling and prediction are discussed later, in Chapter 5.2.

2.7.3 PHM and PIM Model Fitting and Validation

The PHM and PIM model classes are designed for ease of estimation and model checking. The details of the estimation procedures vary depending on factors such as whether a PHM or PIM model is used, whether the family fitted is semi-parametric (arbitrary baseline) or fully parametric (parametric baseline), with what parametric form, whether all individuals are uncensored in-hazard for the same length of time. Some key techniques from the extensively developed statistical fitting and validation methods for PHM and PIM models are briefly summarised in this section.

Before beginning this summary we give a few comments intended to clarify the notation. Where possible both PIM and PHM model methods are indicated by a single equation. If the sample of m individuals under study are indexed by i, $i = 1 \ldots m$, we use j, $j = 1 \ldots n_i$, to index the sequence of events (failures in our software reliability context) observed in connection with individual i. Thus the set of all observed event times for the entire sample of individuals is denoted by $\{\tau_{ij}\}$ by which we mean $\{\tau_{ij} : j = 1, \ldots, n_i,\ i = 1, \ldots, m\}$. Under the PHM model assumptions, it is clear that, for each individual, $n_i = 0$ or 1 so that in the PHM case all times τ_{ij} are in fact τ_{i1}. Thus, the second subscript can be dropped to give $\{\tau_i\}$ when we refer to the PHM model only. When the expression $\{\tau_{ij}\}$, or $\{\tau_i\}$ for the PHM-only case, appears as a conditioning event on the right hand side of a "|", then the conditioning event is to be interpreted below *only* as a full sequence of observed event times, i.e. *without* assignments of these event times to individuals. So in this specific context, a pooled list of times for all individuals is conceived as given without knowledge of the values of either of the associated suffices ij. (We emphasise that there is no suggestion that such ignorance concerning the allocation of event times to individuals reflects the reality of the information available to the observer: it is purely a hypothetical device for explaining the conditioning used in the formulation and description of the Cox *Partial* Likelihood of equation (6).) We continue to use s, $s = 1, \ldots, p$, to index the *scalar* components of both the regression parameter vector β and the m explanatory variables vectors x_i, $i = 1, \ldots, m$. Double summations and products $\prod_{i,j}$ and $\sum_{i,j}$ mean $\prod_{i=1}^{m} \prod_{j=1}^{n_i}$ and $\sum_{i=1}^{m} \sum_{j=1}^{n_i}$, respectively, where, following the usual convention, $\prod_{j=1}^{0} \cdot = 1$ and $\sum_{j=1}^{0} \cdot = 0$. For each time τ, the "risk set" $\mathcal{R}(\tau)$ is the subset of $\{1, \ldots, m\}$ containing the indices i of precisely those individuals which are in hazard and uncensored at time τ, so that, as far as the observer is

concerned, there remains a risk at time $\tau - \delta\tau$ that an observed event may befall individual i at time τ, for arbitrarily small $\delta\tau$.[28]

It turns out [12, 49] that, roughly speaking, the order and comparative numbers in which different individuals experience events determines $\widehat{\beta}$, without regard to more detailed information about the actual times of occurrence. (Although the times matter more if $x = x(\tau)$.) Given an estimate, $\widehat{\beta}$, of β, a good and perhaps optimal [49] estimate of the baseline function, $\widehat{\lambda}_0$ or $\widehat{h}_0$ can subsequently be obtained. β is frequently estimated using the Cox Partial Likelihood, [13],

$$L(\beta) = \prod_{i,j} \frac{e^{x_i'\beta}}{\sum_{l \in \mathcal{R}(\tau_{ij})} e^{x_l'\beta}} \tag{6}$$

This product contains one factor for each observed event. Each such factor may be interpreted as the conditional probability that it is individual i who experiences an event (individual i's j^{th} uncensored event) at time τ_{ij}, given that at times $\tau_{ij} - \delta\tau$, sufficiently closely preceding τ_{ij}, the individuals $l \in \mathcal{R}(\tau_{ij})$ are precisely those which are "in hazard" (and uncensored), and *given that* an unknown one of these individuals *is to* experience an event at time τ_{ij}. Given the event times, and given the risk sets at each event time, $L(\beta)$ is the probability of the ranks, i.e. the sequence of *tags* rather than *times*, where each event is thought of as tagged with the name of the individual to which it belongs. Thus, for explanatory variables which are constant in time, $L(\beta)$ does not depend on the event times, once the ranks are known. The score vector or gradient vector of the log Cox Partial Likelihood, and the matrix of second derivatives can be calculated in the usual way and used to obtain an estimate, $\widehat{\beta}$, iteratively by the Newton-Raphson method. Asymptotic theory for this estimate of the regression parameter is discussed in [12, 13, 15], and can be used for model checking. The score vector has a particularly simple form

$$\begin{aligned} U(\beta) &= \frac{\partial \log L}{\partial \beta} \\ &= \sum_{i,j} \left\{ x_i - \sum_{l \in \mathcal{R}(\tau_{ij})} \frac{e^{x_l'\beta}}{\sum_{k \in \mathcal{R}(\tau_{ij})} e^{x_k'\beta}} x_l \right\} , \end{aligned}$$

the difference between $\sum_{i,j} x_i$ and its "conditional mean". So $\widehat{\beta}$ actually satisfies the equation $U(\widehat{\beta}) = 0$, or $\sum_{i,j} x_i = \mathbf{E}\left[\sum_{i,j} x_i \,\middle|\, \{\tau_{ij}\} : \widehat{\beta}\right]$, with a suitable interpretation of the conditioning.

Given such an estimate, $\widehat{\beta}$, a non-parametric estimate of the cumulative baseline

$$\Lambda_0(\tau) = \int_0^\tau \lambda_0 ,$$

[28] There is a minor issue here concerning the assumptions about the effect of the exact coincidence of an event with the beginning or end of a period of censoring. However, it is a simply matter to sensibly defined the risk set in such circumstances.

is given by

$$\widehat{\Lambda}_0(\tau) = \sum_{\tau_{ij}<\tau} \frac{1}{\sum_{l\in\mathcal{R}(\tau_{ij})} e^{x_l'\hat{\beta}}} \quad , \tag{7}$$

[49]. (This works for both PIM and PHM.)

It has been remarked elsewhere [2] that, if a software reliability model is primarily intended for use in prediction of software failure behaviour vs. time, then empirical measures of the *accuracy of the resulting predictions* are the "goodness of fit" measures which are of greatest practical interest for that model. Needless to say, the same predictive quality comparison techniques [2], based on u-plots, y-plots and prequential likelihood, as have been used for evaluating predictive quality of software reliability models not containing explanatory variables are available both for assessing the extent of any improvement resulting from the incorporation of explanatory variables and also for making comparisons between explanatory variables models. However, some of the central ideas from the extensive separate literature on model identification and validation[29] techniques specific to the PIM and PHM regression models are briefly surveyed here.

To check for the influence of a specific scalar covariate, note that the models do include the case that one (or more) of the covariates, x_s, say, may be extraneous in the sense that its value has no effect on the distribution of the observations. This possibility is accounted for within the framework of the model by setting the corresponding regression parameter, β_s, to zero. Hence after obtaining an initial fit, it is obviously of interest for each scalar covariate, $s = 1, \ldots, p$, to carry out a formal statistical test of the hypothesis, $H_0 : \beta_s = 0$ which will detect whether each fitted scalar regression parameter is significantly different from zero. This can be done using likelihood ratio test statistics, making use of asymptotic results where appropriate, [13, 15]. Subject to the conditions of the asymptotic theory, approximate χ^2 tests can be used.

A proportionality check for the hazard or intensity functions can be carried out graphically by plotting for each of two or more strata formed by grouping individuals according to bands of covariate values:-

- log empirical cumulative hazard function[30] in the case of PHM; and
- log empirical cumulative event-count, in the case of PIM,

against time. The proportionality assumption of the models predicts an approximately constant

[29] "Validation" as used here refers to investigating the extent to which the various structural model assumptions appear to be supported by the observed data, i.e. this is a slightly different matter from the single issue of the accuracy of the reliability predictions emanating from the model and its associated inference procedure.

[30] Empirical cumulative hazard is obtained for each stratum from a simplified version of equation (7) in which each term in the sum is simply the reciprocal of the number of individuals in the risk set for that stratum at one of its event times i.e. $x_i = 0$ for all i.

vertical separation, particularly towards the right hand end where the variance is smaller. Alternatively Schoenfield [90] showed that a plot for each fixed s, $1 \leq s \leq p$, of the scalar residuals, $\widehat{r}_{is} = x_{is} - \mathbf{E}\left[x_{is} \;\middle|\; \{\tau_i\} : \widehat{\beta}\right]$ against τ_i for the PHM could be used to check the proportional hazards assumption. Note that $\sum_{i=1}^{m} \widehat{r}_{is} = 0$ from the equation $U(\widehat{\beta}) = 0$. Various errors of the model assumptions can be represented by imagining the PHM model structure to hold good but with some components of the regression parameter vector to be in truth time dependent. Now we can regard Schoenfield's plot as some points on the graph of a function $\widehat{r}_s(\tau)$ defined at $\tau \in \{\tau_i\}$ by $\widehat{r}_s(\tau_i) = \widehat{r}_{is}$. If a constant $\widehat{\beta}$ is fitted as above whilst the true β of equation (3) has some scalar component β_s which is actually a functions of time, $\beta_s = \beta_s(\tau)$, then Schoenfield shows that this is likely to be reflected in positive values of $\widehat{r}_s(\tau)$ for those regions in τ where $\beta_s(\tau)$ is larger and negative values of $\widehat{r}_s(\tau)$ for those regions where $\beta_s(\tau)$ is smaller. (See also [81].)

Kay [39] produces residuals $e_i = \int_0^{\tau_i} h_i(\tau)\, dt$ by substituting event times in their own predictive cumulative hazard function in order to verify that the resulting $\{e_i\}$ themselves show an empirical cumulative hazard function[31] which is consistent with a sample of unit exponential random variables, i.e. is approximately linear with unit slope and intercept 0.

2.8 Failure-Count Data

It is frequently found in practice that complete inter-failure time data such as discussed above has not been logged. It is then necessary to devise predictors of software failure behaviour which can make do with only the coarser data that is available. Work exists in [1, 67] on maximum likelihood plug-in predictors for software failures in the form of *failure counts*, i.e. counts of failures occurring during consecutive execution time intervals. The approach has been to attempt to carry over, essentially unchanged, the methods used previously for data in the form of execution time between individual software failures, using identical underlying models for the point process of failures vs. usage time metric. Thus, although the point process model of §2.2 remains the same, the observation function $Y : \Omega \to \mathcal{Y}$ discussed in §2.3, which equally influences the form of the likelihood function, is now a different and "less discriminating"[32] one. In fact, when inter-failure times are replaced by failure counts as the observation, then the observation function Y is sufficiently different that a new observation space $\mathcal{Y}$ and measure[33] υ are required. Further investigation of the software reliability prediction problem in these circumstances is the major motivation for the work below in Chapter

[31] See footnote 30 on p31.

[32] Y fails to distinguish between ω_1 and ω_2 which give the same failure counts in each of a sequence of intervals ($Y(\omega_1) = Y(\omega_2)$) even though the times of failures within those intervals may differ from ω_1 to ω_2

[33] In the measure theoretic terminology of §2.3, Lebesgue measure no longer qualifies as a dominating measure υ for the measure $\mathbf{P}_\theta^{\mathcal{Y}}$ which now concentrates probability mass at discrete points.

3. However, it is believed that the methods developed in Chapter 3 actually have wider application than this.

Chapter 3

Extension to the Case of Discrete Predictions

The purpose of this chapter is to address the modification of the theory and methods described in §§2.2–2.5 so that they may be applied in the case where the data available takes the form described in §2.8 of failure-counts during each of a series of elapsed intervals of software execution time. The main novel content concerns the extension of the methods in §§2.4 and 2.5 on U-plots used for predictive quality assessment and recalibration of predictors.

3.1 Software Failure-Count Data

3.1.1 Different Forms of Failure-vs.-Time Data

The work contained in this chapter is primarily motivated by the need to estimate and predict the reliability of software from *failure-count* observations, when *inter-failure time* observations are not available. These two forms of software failure data are discussed in [1] and [67]. Figure 2

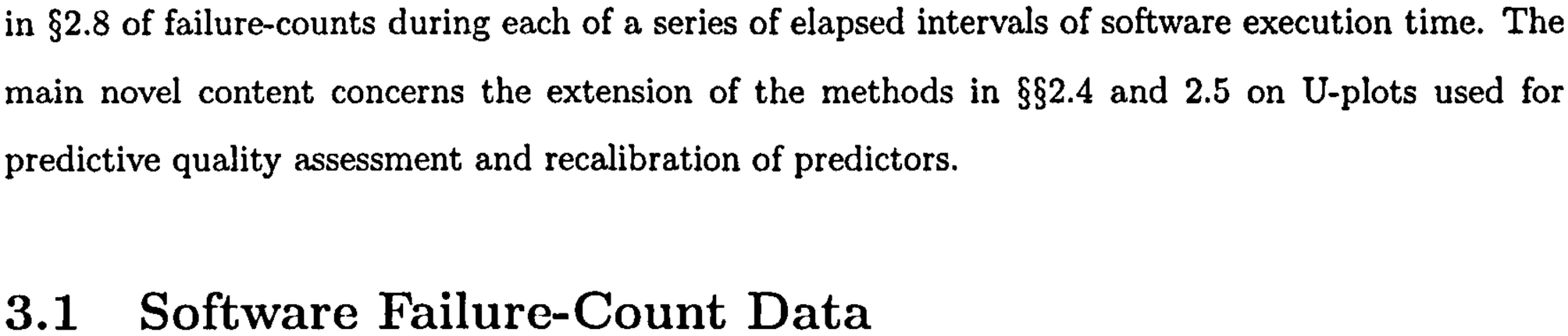


Figure 2: Relationship between complete inter-failure time data and the coarser failure-count data

illustrates the distinction between them. In (a), records of the time elapsed between individual software failures are available to the observer whereas in (b) less complete information is provided in the form of counts of the number of failures occurring during each of a sequence of contiguous time intervals. Thus the word *discrete* in the title of this chapter refers to the observation of quantities which can take discrete values (failure-counts). We are not discussing the discretization of the time metric here. Our underlying stochastic model remains that of a point process in continuous time. The first of the situations illustrated in Figure 2 has been more extensively studied than the second although results based on maximum likelihood are presented in [1] for several models applied to failure-count data.

3.1.2 Inheriting an Inter-Failure Time Model, Unmodified for Use with Failure-Count Data

To model such a failure-count process, it is clearly in principle possible to use identical parametric models to those used to model the complete point process of inter-failure times. Given any parametric family, $\{\mathbf{P}_\theta : \theta \in \Theta\}$, of probability functions for the full point process of inter-failure times, and any deterministic division of the time axis into contiguous intervals, such as that illustrated in Figure 2(b), there is a unique induced parametric family of probability functions for the resulting process $\langle M_n \rangle$ of failure counts. In fact this is a special case of the family $\{\mathbf{P}_\theta^{\mathcal{Y}}; \theta \in \Theta\}$ defined in §2.3 when Y is taken to be the function $Y : \omega \mapsto \langle m_n \rangle$. Any partial observation $Z : \langle m_n \rangle \mapsto z$ of the failure-count process then defines an observation function $Y_1 : \omega \mapsto z$ by composition, $Y_1 = Z \circ Y$, and hence a "discrete likelihood function" $L(\theta, z) = \frac{\partial \mathbf{P}_\theta^{\mathcal{Y}_1}}{\partial v}$ corresponding to any existing parametric model $\{\mathbf{P}_\theta : \theta \in \Theta\}$ for the complete point process of failures. Here, v is some suitable dominating measure[1] on the space $\mathcal{Y}_1$, the image space of the observation function $Z : \mathcal{Y} \rightarrow \mathcal{Y}_1$. This likelihood function can be used to estimate and "plug in" the same parameter vector, θ, as before, but now using only the coarser data (so that a different estimate $\widehat{\theta}$ is obtained). However, we note that:–

- The two likelihood functions for the same model parameters, using the continuous and the discrete-data respectively, usually differ in functional form, (one being obtained from a parameterised family of densities of a continuous random quantity and the other from a parameterised family of discrete probability functions). Therefore there is no guarantee that the analytical and computational problems involved in the remaining stages in the derivation of a predictor, subsequent to the formulation of the likelihood, will bear much resemblance between the

[1] A "counting measure" defined to assign a value 1 to each discrete point in $\mathcal{Y}_1$ will make $z \mapsto L(\theta, z)$ a discrete probability function.

inter-failure time and the failure-count cases.

- It follows that, if a workable prediction algorithm is sought, then it may be preferable *not* to retain identical probabilistic assumptions for the underlying parametric failure point-process $\langle \Omega, \Sigma, \{\mathbf{P}_\theta : \theta \in \Theta\} \rangle$ in taking the step from continuous to discrete data. Ease of analysis[2] is often one of the factors considered in the formulation of a point process model by its original developers. If this is lost in moving from the continuous to the discrete data case, there may be a case for abandoning or modifying the probabilistic model when faced with discrete data. Some modelers have taken this view and modified the underlying probabilistic model in some such way without comment. For example, in the presentation of a discrete-data version of the likelihood for the JM model given in [1], the assumptions about the behaviour of the conditional hazard rate ($z(\tau|\mathcal{G}_\tau)$ on p12) locally within each measurement interval have been modified, simplifying subsequent analysis while approximately preserving (provided the observation intervals are short) the global relationship between failure-rate and cumulative-failures-observed which characterises the JM model.

- In particular, in passing from the continuous to the discrete-data case, two important questions about the tractability of the likelihood function associated with any proposed modification to the probabilistic assumptions are:–

  (a) To what extent is it possible *analytically* to reduce the dimension of the parameter space before carrying out a *numerical search* for the local maxima? (This reduction of dimension being achieved, where possible, by parameter transformation and/or partial differentiation.)

  (b) Can it be demonstrated analytically that the global maximum (rather than a local maximum) of the likelihood function will always be obtained?

- Aside from this issue of model tractability, there may actually be a basis in the *reality modelled* for some modification of the probabilistic model of the underlying failure point-process in the case of failure-count observations. This is because there will often in practice be a relationship between the reporting of software failures and the execution of corrective software maintenance actions which are likely to affect subsequent failure behaviour. For example, it may be realistic to assume that, in the failure-count observation case, attempts to correct any fault occurring during one of the observation intervals illustrated in Figure 2(b) are not made instantaneously following fault occurrence, as many common software failure point-process models assume,

[2] primarily ease of production of a prediction algorithm from a model, in our case

but rather are delayed until the end-time of that interval. Such considerations may perhaps provide further justification—i.e. aside from the mere need for mathematical tractability—for some departure from the exact point-process assumptions used for analysis of inter-failure time data.

3.1.3 Example : The Jelinski-Moranda Model

To illustrate some of the above general points about the transition from inter-failure time to failure count data sequences we briefly, on the next four pages, discuss the case of the JM (Jelinski-Moranda) model [36, 37]. This model is one of the earliest and most simplistic applied to software reliability growth modelling. It is of the *exponential order statistic* class described in §2.2 on p13. The size of the fault population is constant, forming the first model parameter N. The faults are assumed to cause software failure at independently exponentially distributed times τ after the start of program execution and then to be immediately and perfectly removed. The rate parameter ϕ of these N exponential random variables forms the second model parameter, so $\theta = \langle N, \phi \rangle$.

Inter-Failure Time Data

In the case of complete observation of all failure times between $\tau = 0$ and $\tau = l$, where l is a constant observation interval duration, we can represent the observation y as the number, n, and the times, $0<\tau_1<\tau_2<\ldots<\tau_n\leq l$, of failures observed. So this observation $y = \langle n, \tau_1, \ldots, \tau_n \rangle$ lies in a space $\mathcal{Y}$ as described in [16]. The *inter*-failure times $\langle t_i \rangle$ shown in Figure 2 are the differences $t_i = \tau_i - \tau_{i-1}$. The likelihood function is obtained as the parameterised probability density

$$\begin{aligned} L(\theta, y) &= \left(\prod_{i=1}^{n}(N-i+1)\right)\left(\prod_{i=1}^{n}\phi e^{-\phi\tau_i}\right)e^{-(N-n)\phi l} \\ &= \left(\prod_{i=1}^{n}(N-i+1)\right)\phi^n e^{-\left[\sum_{i=1}^{n}\tau_i+(N-n)l\right]\phi} \end{aligned}$$

evaluated at the observation y. The ML parameter estimate $\left\langle \widehat{N}, \widehat{\phi} \right\rangle$ is obtained by maximizing L, which is most easily performed by noting that

$$\log L = -lN\phi + \sum_{i=1}^{n}\log(N-i+1) + n\log\phi - \left[\sum_{i=1}^{n}\tau_i - nl\right]\phi \tag{8}$$

For each fixed $N \geq n$, this function has a unique maximum at $\phi = h(N)$, say, where

$$h(N) = \frac{n}{\sum_{i=1}^{n}\tau_i + (N-n)l} \tag{9}$$

so that by substituting this value in (8) the problem is reduced to that of maximizing the univariate function of N

$$\log L = \sum_{i=1}^{n} \log(N-i+1) - n \log\left(\sum_{i=1}^{n} \tau_i + (N-n)l\right) + n \log n - n \tag{10}$$

If we now define $\zeta = \frac{1}{l}\sum_{i=1}^{n} \tau_i$ and substitute $x = \frac{l}{n}h(N) = \frac{1}{N-n+\zeta}$ in (10), the problem becomes one of maximising $\ell(x)$ for $x \in (0, \zeta^{-1}]$ where

$$\begin{aligned} \ell(x) &= \sum_{i=1}^{n} \log\left(\frac{N-i+1}{N-n+\zeta}\right) + n(\log n - \log l - 1) \\ &= \sum_{i=1}^{n} \log\left(1 + (n-\zeta-i+1)x\right) + n(\log n - \log l - 1) \end{aligned}$$

But ℓ is a sum of logarithms of positive linear functions of x and hence is a sum of concave functions and is thus itself concave. It follows that the ML solution is given by

$$\widehat{x} = \begin{cases} 0, & \text{if } \ell'(0) \leq 0; \\ \zeta^{-1}, & \text{if } \ell'(\zeta^{-1}) \geq 0; \\ \text{one of the two values adjacent to the unique solution of } \ell'(x) = 0, & \text{otherwise.} \end{cases} \tag{11}$$

Dealing with the three cases here in turn:-

$x = 0$ corresponds to the limit $N \to \infty$ and is an HPP model with rate $\frac{n}{l}$ obtained as $\lim_{N\to\infty} Nh(N)$ in (9);

$x = \zeta^{-1}$ is the other end point of the region of feasible x values and corresponds to $\widehat{N} = n$, i.e. all faults have been observed;

$\ell'(x) = 0$ Because ℓ is concave[3], it follows that there must be exactly one such stationary point if the maximum is not at either end point. Then the best x corresponding to *integer* N must be the closest such x on one side or the other of the stationary point.

Failure-Count Data

Moving to the failure count case, assume the interval $[0, l]$ is partitioned into k failure-count observation intervals by $0 = l_0 < l_1 < \ldots < l_k = l$ which are treated as deterministic. To simplify what follows introduce a notation $d_i = l_i - l_{i-1}$ for observation interval length. If the observation consists of failure counts $y_1 = (m_1, \ldots, m_k)$[4] for these intervals we use the notation $c_i = \sum_{j=1}^{i} m_j$ for cumulative failures observed. Thus $c_0 = 0$, and $c_k = n$, retaining the notation n for the total

[3] In fact $\ell'(x)$ is *strictly* decreasing.

[4] The number of scalar observations is no longer random being determined in advance by the number k of failure count intervals, so the space $\mathcal{Y}_1$ of p35 is $\mathbb{N}^k$.

number of failures observed in time l consistently with the previous case (equation (8)). If the same point process model is assumed, then the general method of dealing with failure-count observations mentioned above leads to a multinomial likelihood function with N for the number of trials and $k+1$ possible outcomes for each trial. These $k+1$ outcomes have probabilities $1-e^{-l_1\phi}$, $e^{-l_1\phi}-e^{-l_2\phi}$,..., $e^{-l_{k-1}\phi}-e^{-l\phi}$, $e^{-l\phi}$ respectively. Thus, now

$$\begin{aligned} L(\theta, y_1) &= \frac{N!}{(N-n)!\prod_{i=1}^{k} m_i!} e^{-l\phi(N-n)} \prod_{i=1}^{k} \left(e^{-l_{i-1}\phi} - e^{-l_i\phi}\right)^{m_i} \\ &= \frac{\left(\prod_{i=1}^{n} N-i+1\right)}{\prod_{i=1}^{k} m_i!} e^{\left(-lN+\sum_{i=1}^{k}(l-l_{i-1})m_i\right)\phi} \prod_{i=1}^{k} \left(1-e^{-d_i\phi}\right)^{m_i} \end{aligned}$$

In changing to failure-count observations for this model we have finished up with a different and in this case less tractable likelihood function. This is not surprising since Jelinski and Moranda published their model together with a suggested inference procedure for the inter-failure time case, and it could easily be that the publication of this precise model was in part due to their demonstration of a workable inference procedure. Taking the logarithm, as for the inter-failure-time case, the following form is obtained.

$$\log L = -lN\phi + f(N) + g(\phi) - K$$

where $f(N) = \sum_{i=1}^{n} \log(N-i+1)$, $g(\phi) = \sum_{i=1}^{k} d_i c_i \phi + \sum_{i=1}^{k} m_i \log\left(1-e^{-d_i\phi}\right)$, (the identity $\sum_{i=1}^{k}(l-l_{i-1})m_i = \sum_{i=1}^{k} d_i c_i$ has been used here), and $K = \sum_{i=1}^{k}\sum_{j=1}^{m_i} \log j$. We can obtain

$$\frac{\partial \log(L)}{\partial \phi} = 0 \quad \text{for} \quad N = \frac{g'(\phi)}{l}$$

and, interpolating between integer values of N,

$$\frac{\partial \log(L)}{\partial N} = 0 \quad \text{for} \quad \phi = \frac{f'(N)}{l}$$

but it is clear that this analysis, though feasible, is beginning to get quite involved at this point, due to the awkward form of the ϕ-term in comparison to the previous case.

A Modification of the Underlying Model for the Discrete-Data Case: The above illustrates the differences and frequently, so it has been found, increased difficulty of an analysis based on the likelihood function arising from the failure count case if we stick strictly to the idea of shifting to a failure-count observation function whilst retaining *exactly* the same underlying point process models as have been developed for reliability modelling centering around a full inter-failure time data sequence. It is observed that in [1] the point process model has been tacitly changed so that, in his discussion of the modelling of failure counts based on what he refers to as the JM model, the probabilistic model assumed by Abdel Ghaly is actually subtly different from that used in previous

analyses of inter-failure time data. In fact Abdel Ghaly's model uses probabilistic assumptions which actually depend on the failure count intervals (i.e. the $\langle l_i \rangle$) which are used. His implied point-process model (i.e. that implied by his statement of his version of the likelihood function for failure-count observations) can most succinctly be described by specifying that the conditional program failure rate function, $z(\tau|\mathcal{G}_\tau)$ described in §2.2, takes the same random value[5] when expressed in terms of the past observations at $\tau = l_i$ as it would have under the assumptions of the true JM model, but, unlike the original JM model, the rate now remains *constant* throughout the failure count interval (l_i, l_{i+1}) irrespective of failures occurring during that interval, i.e. we have $z(\tau|\mathcal{G}_\tau) = z(l_i|\mathcal{G}_{l_i})$ for $l_i < \tau < l_{i+1}$. Thus the point process is, conditionally given failure times prior to l_i, a Poisson process throughout the period of the interval (l_i, l_{i+1}). This adjustment of model assumptions is faithful to the fundamental conception of the JM model provided that the failure count intervals are many and short[6], and it results in a likelihood function for a failure count observation which can be expressed as a product of Poisson probabilities

$$\begin{aligned} L(\theta, y_1) &= \prod_{i=1}^{k} \left[\frac{e^{-d_i\phi(N-c_{i-1})} \left(d_i\phi(N-c_{i-1})\right)^{m_i}}{m_i!} \right] \\ &= \left(\prod_{i=1}^{k} \frac{d_i^{m_i}}{m_i!} \right) \phi^n \left(\prod_{i=1}^{k} (N-c_{i-1})^{m_i} \right) e^{\left(-lN + \sum_{i=1}^{k} d_i c_{i-1}\right)\phi} \end{aligned}$$

and which consequently turns out to be considerably more tractable mathematically. It leads to log likelihood

$$\log L = -lN\phi + \sum_{i=1}^{k} m_i \log(N - c_{i-1}) + n \log \phi + \sum_{i=1}^{k} d_i c_{i-1} \phi + K \tag{12}$$

where $K = \sum_{i=1}^{k} \left(m_i \log d_i - \sum_{j=1}^{m_i} \log j \right)$. It turns out that with this form of the likelihood we have returned to an inference problem with, for each fixed $N \geq n$, a unique maximum of $\log L$ given again by an explicit closed form expression $\phi = h(N)$ where now

$$h(N) = \frac{n}{lN - \sum_{i=1}^{k} d_i c_{i-1}} \tag{13}$$

The remaining analysis is very similar to the inter-failure time data case. (13) can be substituted for ϕ in the log likelihood formula (12) to give

$$\log L = \sum_{i=1}^{k} m_i \log(N - c_{i-1}) - n \log \left(lN - \sum_{i=1}^{k} d_i c_{i-1} \right) + n \log n - n + K \tag{14}$$

[5]This value is $z(l_i|\mathcal{G}_{l_i}) = (N - c_i)\phi$.

[6]In fact, as mentioned under one of the "bullet points" on p36, it could be interpreted as a deliberate representation of delayed failure reporting and fault correction, which introduces the possibility of multiple occurrence of a single fault during a single failure-count interval.

and the problem is again reduced to the maximisation of a scalar function of N. Putting $\zeta = \frac{1}{l}\sum_{i=1}^{k} d_i c_{i-1}$ and substituting $x = \frac{l}{n}h(N) = \frac{1}{N-\zeta}$ in (14) gives the function

$$\ell(x) = \sum_{i=1}^{k} m_i \log(1 + (\zeta - c_{i-1})x) + n(\log n - \log l - 1) + K$$

to be maximised from $x \in (0, \frac{1}{n-\zeta}]$. It can be seen, as for the inter-failure time case, that this function is concave on this interval and therefore always has a maximum point which can easily be determined. The cases in which the maximum is at the end points of $x \in (0, \frac{1}{n-\zeta}]$ are easily identified, by considering the sign of $\ell'(x)$. Such considerations lead to the conclusion that the ML fit is an HPP with rate $\frac{n}{l}$ when $\zeta \leq \frac{1}{n}\sum_{i=1}^{k} m_i c_{i-1}$; or, in the opposite extreme case, the ML solution is the "completed debugging" case, in which $\widehat{N} = n$, when $\frac{1}{n}\sum_{i=1}^{k} \frac{m_i}{n-c_{i-1}} \leq \frac{1}{n-\zeta}$. In other cases $\widehat{N}$ can be found numerically as an integer adjacent to the value corresponding to the unique zero of $\ell'(x)$ in the interval $(0, \frac{1}{n-\zeta})$. Equation (13) gives $\widehat{\phi}$ in terms of $\widehat{N}$.

3.1.4 Failure-Count Data, Reliability Prediction, & Prequential Forcasting Systems

We now leave the illustrative JM example and return to the general case of failure-count data, concentrating more specifically on the task of reliability prediction (see §2.3). A problem which is encountered in the attempt to emulate the approach used for inter-failure time observations concerns the difference, in the failure-count case, of the relationship between that which is *observed* (i.e. the terms m_n in the sequence of observations) and the type of *predictions* which are desired to be produced from these observations. Many commonly accepted methods of expressing current reliability, (reliability function, mean-time-to-failure, percentiles of time-to-failure distribution, and conditional failure rate, see p10) are defined in terms of the predictive distribution of the time to *next* failure, given past observations. Thus, in the inter-failure time case, all that is required for the estimation of a conventional mathematical representation of current reliability is a one-step-ahead predictive distribution, $F_n^T(t_n; t^{n-1})$,[7] of the observed process (T_n) itself, since it is the observed quantity T_n whose predictive distribution contains many of the software reliability measures in which there is interest. For this reason tracking the reliability of a software product from inter-failure time observations can be thought of in terms of a repeated cycle of the form: predict T_n, observe $T_n = t_n$, predict T_{n+1}, observe $T_{n+1} = t_{n+1} \ldots$. Expression of the problem in this particular form allows

[7] The sometimes suppressed t^{n-1}-term to the right of the semicolon is made explicit in the notation of this thesis to emphasise that the predictive distribution is thought of as a random entity, being a function of all the previous terms in the process. t^{n-1} denotes conditioning information ($\mathcal{G}$ of P10) in the form of observation of the process realisation ω for exactly as long as it takes for the first $n-1$ failures to occur. Regarding the superscript $n-1$ notation, see footnote 22 on p19.

the construction of a $\langle U_n \rangle$ sequence and hence of prequential likelihood, u-plots, and y-plots, giving measures of predictive accuracy as well as improved, recalibrated predictors [10, 59, 11, 6]. In the failure-count case however, estimation of current reliability expressed in the familiar mathematical terms requires the prediction of a quantity which will never be directly observed, i.e. of the *time* to next failure, given the previously observed *failure counts*. The approach taken to circumvent this difficulty here has been to assume that it is still adequate in the failure-count case to *represent* the current *reliability* in terms of the *predictive distribution* of the *next term in the observed sequence*, i.e. the number of failures which will occur within an immediately subsequent execution time interval of given duration. If this is accepted, then we are again in the situation of one-step-ahead prediction of an observable sequence, i.e. we require to obtain predictive distributions, $\langle F_n^M(m_n; m^{n-1}) \rangle$, of failure counts, for the given intervals. Later, we find that this approach—after a bit of extra work—does indeed enable us to derive something analogous to the recalibration procedure previously used to predict reliability from inter-failure times. However, note that it is *not* our intention to imply by this simplification that something analogous to recalibration *cannot* be performed successfully for more general kinds of prediction. On the contrary, some proposals of ways in which this might be done are included in §3.7. The point is rather that, as a *first step* towards recalibrating from failure-count observations, we attempt to model as closely as possible the successful experience of recalibrating inter-failure time predictions which was based on one-step-ahead prediction of inter-failure time sequences. Hence we begin by producing predictive distributions for the next failure-count in our observation sequence in order to frame the prediction task in the same mathematical form of a Prequential Forecasting System (PFS) [20], or sequence of predictors $\langle F_n^X(x_n; x^{n-1}) \rangle$, the only difference being that $\langle x_n \rangle$ is now a sequence of failure-counts rather than a sequence of inter-failure times.

3.1.5 Three Difficulties of the Failure-Count Case

Having formulated the reliability estimation problem in this way, there remain *three* important further differences from the case of individual inter-failure time observations, when it comes to predictive quality *assessment* and *recalibration.* Only the third of these differences perhaps obstructs the use of prequential likelihood plots for assessment and comparison of predictors, but *all three* present potentially significant problems in the definition of a *u-plot* for assessment and for recalibration. The nature of these problems with u-plots in the failure-count case is briefly sketched here. The problems are then explored in greater detail in the following sections of this chapter, along with some proposed approaches to overcoming them.

1. Varying Length Execution-Time Intervals: The first difference is the impact of the elapsed execution-times $\langle d_n \rangle$ corresponding to each successive failure-count. Any variability with n of these times constitutes an additional deterministic source of variability in the observation sequence, $\langle X_n \rangle$. (The counts of failure occurrences during longer periods of executing the software will tend to be larger, in comparing intervals located at around the same period in the evolution of the software.[8]) However, the justification of a recalibration procedure based on the u-plot relies on some notion of approximate constancy of the distribution of U_n as n varies. (See §3.6.) Variability in interval lengths complicates matters by interfering with this.

2. Constructing U-Residuals from Discrete Predictive c.d.f.s: The second sense in which u-plots based on failure-count data differ from u-plots based on inter-failure time data arises from the *discreteness* of the predictive distributions; or, more precisely, from the eventuality of a random variable assuming a *point value* which has been correctly predicted with *positive probability.* The resulting discontinuities in the predictive c.d.f.s $F_n^X(x_n; x^{n-1})$ are shown in §3.2 to cause corresponding discontinuities and *bias* in the distributions of the U_n. It is later shown in §§3.4 & 3.5 that this effect can be overcome by defining a u-plot in terms of the pairs $\langle F_n^X(x_n-; x^{n-1}), F_n^X(x_n; x^{n-1}) \rangle$. This recaptures, in the discrete case, some of the desirable properties of the familiar u-plot applied to the prediction of continuously distributed processes.

3. Small Size of the Sample of Observed U s Available for U-plot Construction: Another quite simple practical problem is frequently encountered in approaching the recalibration of a set of software reliability predictions based on failure-count data. Recalibration requires the accumulation of a set of observed u_n from predictions of earlier terms in the sequence predicted. In particular, in order for the u-plot, which forms the basis of the recalibration method, to be reasonably statistically stable, a reasonably *large* collection of previous u_ns must have been accumulated so that reliable recalibration can commence. This remark applies particularly to the problem of achieving stability in the *tails* of the recalibrated predictive distribution, since the proportionate accuracy of the tail-probabilities is highly dependent on a small sample of the largest (or smallest) u_n values observed so far. The need for a reasonably plentiful stock of past observation values x_n arises also from the fact that the accumulation of u_n values cannot even *begin* until sufficient early observations have been made to first fit the mathematical process model and thus begin producing the successive raw predictive c.d.f.s in terms of which the u_n residual sequence is defined. In one typical data set

[8] As a first approximation the predictive distribution could be assumed to be, locally, and conditionally on the past, $\mathcal{G}$, Poisson with mean equal to some function of interval length.

seen by the author, 193 failures of a software product had been recorded, but time of occurrence information had been recorded only by counting the number of these 193 failures which occurred during each of 17 contiguous observation time intervals. Had complete inter-failure time data been recorded for these failures, it would have been straightforward to fit reliability growth models to the data, successively adding one additional failure time to the set used for model fitting, in the standard way required to apply an inter-failure time PFS, and hence accumulating a large set of u_n values, which could then have been used to improve the raw model's predictions by incorporating a recalibration step in a new PFS. In the actual case, having instead only the 17 failure counts, there seems far less scope for including a useful recalibration step in the analysis due to the limited size of the sample of u_n resulting from one-step-ahead prediction of such a short sequence of observations $\langle m_n \rangle$. We have found however that the use of a *constrained gradient* of the u-plot smoother, as we will propose in more detail in §3.6.6 below, contributes towards the stabilization of a u-plot based on a small sample of predictions.

One would anticipate that the rate of evolution of software failure behaviour, *per data point* of the sequence to be predicted, is generally likely to be higher for failure count data than for complete inter-failure time data. This might perhaps pose further problems by interfering with the desired approximate stationarity of the u_n sequence when that sequence is to be used for purposes of recalibration, again making things worse for the "would be recalibrator" of failure-count predictions. One rather ad hoc proposal for recalibrating a predictor of rapidly evolving reliability data is to use decaying weights as we suggest in §3.6.3.

Concerning the lengths of failure-count intervals, note that paragraphs 2 and 3 above constitute two *competing* considerations: Shorter intervals, and a consequently larger set of predicted (and then observed) counts, would tend to mitigate the problem identified in paragraph 3. On the other hand, the problem of recalibrating failure-count predictions mentioned in paragraph 2 arises specifically from the *discreteness* of the individual predictive distributions. It will be seen later that the effect of this discreteness in biasing the u_n distribution is far more pronounced if the failure-count intervals are short. Suppose, for example, that, for some interval $[l_{n-1}, l_n]$, three small values, $m_n = 1, 2, 3$, are the only values of the failure count that are predicted with significantly large probabilities. Then, each of these count values will produce a significant *discontinuity* in the associated predictive c.d.f. $\langle F_n^M(\cdot; m^{n-1}) \rangle$. These few large discontinuities will contribute significantly to the bias in the distribution of $U_n = F_n^M(M_n; m^{n-1})$. It is this bias in the distribution of the U-residual which is the obstacle to discrete recalibration discussed in paragraph 2, and further examined in §3.3. However, if the modified u-plot developed in §§3.4 & 3.5 solves this difficulty with discontinuities, then there is

no longer an argument from paragraph 2 in favour of the predictive distributions being continuous, or approximately continuous. It could then be argued strongly from the consideration discussed in paragraph 3 that, for the purpose of producing raw predictions which can be effectively recalibrated, the shortest possible failure-count time intervals should be used when collecting the failure count data.

In the following sections of this chapter, we develop some initial approaches to overcoming the difficulties sketched here, so as to extend the use of the recalibration procedure to failure-count-based software reliability estimation. However, we feel there may well remain further scope for both other analytic refinements, as well as further experimental work to confirm or refute the hypothesized benefits of these various modifications for the efficacy of predictor recalibration in the discrete-data case. Some ideas for further validation work on the approaches developed in the remainder of this chapter are proposed later in chapter 7.

3.2 U-Plots

This section discusses the standard "u-plot", [2], with an examination of the joint probability distribution of the $\langle U_n \rangle$ sequence under combined assumptions about the process being predicted and the prediction system applied. For an informal overview of the purpose and uses of the u-plot, formed from the application of a one-step-ahead prediction system which produces its predictions in the form of predictive distributions, see §§2.4 and 2.5. Further details are given below. In [2, 6, 10] and [58] u-plots are used in order to *assess* the quality of one-step-ahead predictions based on inter-failure time data, and are used further in order to *improve* the predictive quality by *recalibrating* the predictions (see also [11] and [59]). The main substance of this chapter consists of an extension of this kind of technique suitable for application in the failure-count case. As explained on p42, we concentrate on one-step-ahead prediction of the observed sequence itself, with a brief mention of possible extension to longer term prediction and prediction of other quantities in §3.7.

Recall from §2.4.2 that a Prequential Forecasting System (PFS) for a random process $\langle X_n \rangle$, $n = 1, 2, \ldots$, is a sequence of c.d.f. functions $\langle F_n^X(x_n; x^{n-1}) \rangle$ where F_n^X is to be interpreted as the one-step-ahead predictive c.d.f. of X_n as its first argument, having observed the realisation $X^{n-1} = x^{n-1}$ of the process so far. The function $x \mapsto F_n^X(x; x^{n-1})$ must be a valid c.d.f. (monotonic non-decreasing, right-continuous, with limits 0 and 1 respectively as $x \to -\infty$ or $+\infty$) defined by the PFS for all possible values of x^{n-1} and for each n.

In the examples of chapter 4, F_n^X is defined only for $n \geq n_0 = 6$. I.e. "raw" predictions are

begun after 5 terms have been observed. The term "raw" is used in what follows to describe predictors and their associated predictions when these do not rely on a recalibration stage in the total prediction algorithm. In contrast, recalibrated predictions, which will be discussed in §3.6, are begun in the examples of chapter 4 only after the realisations corresponding to 10 raw predictions are first available, i.e. $n_0 = 16$ for our recalibrated predictors.

We restrict to proper c.d.f.s, (i.e. not allocating predictive probability to $X_n = \pm\infty$) in §§3.2–3.6, in order to avoid having to mention too many special cases. There is no real obstacle to extending the procedures to include improper c.d.f.s if desired. Indeed such c.d.f.s are examples precisely of predictive probability concentrated at a point, for which purpose our "modified u-plot" and its associated recalibration procedures described below in this chapter were designed. The usual $\langle u_n \rangle$ sequence [2, 6, 10, 58] can be thought of as a realisation of a process $\langle U_n \rangle$, defined in terms of the observed process, $\langle X_n \rangle$, and the PFS by substitution of successive observations in the predictive c.d.f.

$$U_n = F_n^X(X_n; X^{n-1}), \quad n \geq n_0 \,. \tag{15}$$

Of course, the joint probability distribution of the resulting $\langle U_n \rangle$ process depends on both the PFS $\langle F_n^X(\cdot;\cdot) \rangle$, and on the true[9] probability distribution of the process $\langle X_n \rangle$ to which the PFS is applied.

3.2.1 Common Notation for Continuous, Discrete, and Mixed Case

In order to provide, as far as possible, a common framework for handling the two types of software failure-vs.-time data mentioned in §3.1, we discuss in §§3.2–3.6 the definition and the use for predictor recalibration of a "u-sequence" in the general case of predicting an arbitrary scalar random process in discrete time, i.e. an arbitrary random sequence. This entails allowing for the cases in which the next observation has a predictive distribution which is either *continuous*, or *discrete*, or *mixed*. The first two of these three alternatives correspond respectively to inter-failure time and failure-count PFSs. The inclusion of the third might appear superfluous. It does not seem to introduce any essentially new problems, however, as far as the techniques suggested in §§3.2–3.6 are concerned, and does clarify the relationship between the "u-plots" and the recalibration algorithm discussed in the above references, and those applied in Chapter 4 here to failure-count data. Also there is a

[9] There may be valid objections here to the notion of a 'true', unknown probability distribution for an observed process of software failures. The important point for us here (about the proposition that such a thing exists) is that it enables us to acknowledge explicitly within our mathematical formalisms that the PFS $\langle F_n^X(\cdot;\cdot) \rangle$ we have used to define the $\langle u_n \rangle$ sequence may correspond to a process law that could in principle be improved, and to explore how the nature of the inadequacies of this PFS will determine the stochastic behaviour of the process $\langle U_n \rangle$ of 'residuals' resulting from the application of our imperfect PFS. See §3.2.3 below.

possibility that the general, mixed case might find an application in another context. We represent the probability distribution of a general scalar random variable by its c.d.f. and, to allow for the case in which it does not possess a density function with respect to Lebesgue measure, make use of Lebesgue-Stieltjes integrals (see e.g. [21, especially Chapter 9]) for probabilities and expectations. I.e. we use a Stieltjes-like integral whose formal interpretation involves the use of a function of bounded variation[10] to define a Borel-measure on $\mathbb{R}$ with respect to which the integral can be formally defined using the usual theory of integration with respect to an abstract measure. These integrals reduce to the familiar finite or countably infinite sums in the discrete, failure-count case, and to integrals of expressions involving continuous probability density functions in the case of continuous inter-failure times. However, the available rigorous theory allows extension to the general, mixed-distribution case when this is required.

In the case where the PFS provides predictive c.d.f.s $F_n^X(x_n; x^{n-1})$ which are *continuous* in the argument x_n (presumably indicating a belief on the part of the observer that the corresponding conditional c.d.f. of the true distribution of the process $\langle X_n \rangle$ is likewise continuous), the u-plot and y-plot can be employed in order to test whether the realised $\langle u_n \rangle$ sequence appears consistent with an i.i.d. uniform $\mathcal{U}[0,1]$ joint distribution (see e.g. [2]). This approach is not always appropriate in the *discontinuous* case for reasons which become apparent when we examine, in this section, the distribution of the $\langle U_n \rangle$ process in this more general case.

3.2.2 The Probability Measure Defined by a PFS $\mathcal{P}$

We think of a PFS for the process $\langle X_n \rangle$ as being equivalent to an assumed probability distribution, $\mathbf{P}_{\mathcal{P}}$, for the process, (see [20], p4). If $n_0 > 1$, then x^{n_0-1} will be observed before carrying out any prediction and can be regarded as a directly observable parameter of $\mathcal{P}$ and of all other distributions, probabilities and expectations considered in what follows. Thus $\mathcal{P}$ assigns probabilities[11]

$$\mathbf{P}_{\mathcal{P}}[X_{n_0} \in A_{n_0}, \ldots, X_n \in A_n] =$$

$$\int_{x_{n_0} \in A_{n_0}} \int_{x_{n_0+1} \in A_{n_0+1}} \cdots \int_{x_n \in A_n} dF_n^X(x_n; x^{n-1}) \ldots dF_{n_0+1}^X(x_{n_0+1}; x^{n_0}) dF_{n_0}^X(x_{n_0}; x^{n_0-1}) \quad (16)$$

to cartesian product sets. $\mathbf{P}_{\mathcal{P}}$ represents a probabilistic model for the process, whose conditional distributions are used in making one-step-ahead predictions, even though the distribution $\mathbf{P}_{\mathcal{P}}$ may not be the basis of the *derivation* of the PFS—It is not, for example, in the case of a maximum likelihood "plug-in" PFS. In fact, we note that, whether $\mathcal{P}$ is derived via ML or Bayesian analysis

[10] c.d.f.s are trivially of bounded variation

[11] This repeated-integral construction of the probability measure $\mathbf{P}_{\mathcal{P}}$ assumes that the PFS is such that the intermediate integrals form integrable functions.

of a parametric point process model $\{\mathbf{P}_\theta : \theta \in \Theta\}$, the distribution $\mathbf{P}_\mathcal{P}$ implied by the forecasting system may be quite distinct from any $\mathbf{P}_\theta$, and in particular, events which are independent under $\mathbf{P}_\theta$ for all $\theta \in \Theta$ may be associated i.e. *de*pendant under the distribution $\mathbf{P}_\mathcal{P}$. Thus the *predictive* c.d.f. $F_n^X(x_n; x^{n-1})$ is now assumed to be identified also as a *conditional* c.d.f. $F_n^X(x_n|x^{n-1})$ associated with the joint distribution $\mathbf{P}_\mathcal{P}$. Actually the problem of formalising in a fairly general case the relationship between a PFS $\mathcal{P}$ and its associated or implied stochastic process probability law $\mathbf{P}_\mathcal{P} : \Sigma \rightarrow [0,1]$ remains a topic of research interest among probability theorists [92]. There is also a problem of the non-uniqueness of conditional probability measures defined for a single global process measure $\mathbf{P}_\mathcal{P}$ (see footnote 4 on p10). For the remainder of this chapter we will ignore these matters and attempt to motivate and explain the modified u-plot and recalibration techniques presented on the assumption that our predictive c.d.f. functions may be written $F_n^X(x_n|x^{n-1})$ (i.e. with ";" now replaced by "|") and will satisfy all properties of conditional distributions corresponding to the process probability measure $\mathbf{P}_\mathcal{P}$. Even where this assumption is incorrect, the algorithm for *construction* of the modified u-plots and recalibrators[12] remains precise and unambiguous in terms of any given PFS, $\langle F_n^X(\cdot;\cdot)\rangle$.

3.2.3 Recalibration Conceived in Terms of a *True* PFS?

The procedure of employing the evolving u-plot, during an application of a PFS to a software failure data sequence, for the purpose of "recalibration", (i.e. producing a new and, it is hoped, better PFS for the same data) is not easy to justify formally. Ideally in practice, the raw PFS employed ought to encapsulate the observer's current probability model for the process being observed. However, the addition of a recalibrator to the prediction algorithm necessarily implies a willingness on the part of the observer to entertain probability functions for the process which are at variance with $\mathcal{P}$, the one implicit in the raw PFS. This idea seems awkward to handle theoretically since an observer who is prepared to *learn* from the $\langle U_n \rangle$ and thereby to produce a *recalibrated* PFS, must think in terms of employing two inconsistent probability models for the process $\langle X_n \rangle$: firstly the probability measure $\mathbf{P}_\mathcal{P}$ implicit in the raw PFS $\mathcal{P}$, and secondly a higher level model which is able to include the possibility that $\mathcal{P}$ may differ from the truth and which forms the basis of the recalibrated PFS. Thus the underlying "real-world" mechanism generating the observations may be assumed equivalent to some unknown (and perhaps not unique) 'perfect' PFS $\mathcal{Q}$ for the process $\langle X_n \rangle$, where it is formally accepted that we may find $\mathcal{P} \neq \mathcal{Q}$. At this point we should make a small digression to examine this statement. We have expressed the development which follows, firstly of the probability law

[12] A recalibrator is an algorithm for improving the predictive quality by "recalibration" of each prediction in the light of what can be learned by observing predictive performance so far.

of the $\langle U_n \rangle$-process, and secondly of recalibration and of what recalibration achieves, in terms of this distinction between the PFS $\mathcal{P}$ (and its associated probability law $\mathbf{P}_{\mathcal{P}}$) and some other 'true' or 'real-world' PFS $\mathcal{Q}$ for the process. The question can reasonably be asked of whether it is meaningful to talk of the *true* probability law for something 'real', i.e. for a random process that is found in an application of probability to a real problem. Where the experiment of observing this process is in some sense 'repeatable' under 'identical conditions' then perhaps the law of large numbers can be used to argue that there are true probabilities for events that *could* in principle be estimated to any desired accuracy. The arguments below can be adjusted, to partially accommodate this potential criticism of using a notion of 'true PFS' $\mathcal{Q}$, by arguing instead that the raw PFS $\mathcal{P}$ we are using may not be the best we could ever come up with for predicting this failure process. Then we can describe $\mathcal{Q}$ instead merely as 'some hypothetical better PFS than the one we already have'—Or perhaps more correctly : 'some better PFS which exhibits superiority precisely of a kind which will be detected by examination of u-plots'. We have not explored this issue as carefully as perhaps it could be, but feel confident that such an adjustment to the interpretation of $\mathcal{Q}$ is possible and could be used below to construct slightly different developments of the main arguments. From this point on, however, we will continue the discussion in terms of a probability law $\mathbf{P}_{\mathcal{Q}}$ for the process which is allowed to differ from our working (i.e. numerically implemented[13]) PFS $\mathcal{P}$ and to which we will refer using terms such as the *best, true, ideal, perfect, etc* PFS, to indicate that $\mathcal{Q}$ is supposed to be an *unknown but superior* representation of the true nature of the random behaviour of the empirical software failure process. Note however that at least for the *simulated* failure-data sets in chapter 4 (i.e. for JM1, L1, and LV1 on p84) we *do* have an obvious interpretation of $\mathcal{Q}$ as the probability law that was used to simulate the failure data[14].

3.3 Behaviour of Us from an Ideal PFS

Having agreed to think of the situation in these terms an immediate question is: How should the process $\langle U_n \rangle$ behave if we *do have* the perfect PFS, i.e. if $\mathcal{P} = \mathcal{Q}$? This question is partially answered in the general case by examining the following conditional distribution under this assumption that $\mathcal{P} = \mathcal{Q}$.

$$\underset{\mathcal{P}}{\mathbf{P}}\left[U_n \leq u | x^{n-1}\right] \quad = \quad \underset{\mathcal{P}}{\mathbf{P}}\left[F_n^X(X_n|x^{n-1}) \leq u | x^{n-1}\right]$$

[13] as a general point, the fact that to be useful a PFS has to be both analytically derived and manipulated, and computationally implemented is one good answer to the question: "Why are you not already using the 'best' raw PFS $\mathcal{Q}$?"

[14] Denoted by TRUE in Table 3 on p88

$$= \sup_x \{F_n^X(x|x^{n-1}) : F_n^X(x|x^{n-1}) \leq u\} \qquad (17)$$
$$= G_n(u|x^{n-1}), \qquad \text{say.}$$

Here, we have used the fact that if W is a random variable with c.d.f. F, and m is a monotonic non-decreasing function, then $\mathbf{P}[m(W) \leq a] = \sup_w \{F(w) : m(w) \leq a\}$[15]. We will apply this fact repeatedly in this chapter to obtain the one-step-ahead c.d.f. of U_n under various modifications to its definition. The conditional c.d.f., G_n, takes the form illustrated in Fig. 3(b), where for each n, the pairs (p_{nk}, q_{nk}) correspond to any points a_{nk} of discontinuity of the conditional c.d.f. $F_n^X(x_n|x^{n-1})$ in its first argument x_n. The points a_{nk} are, in the most general case, functions of x^{n-1} and, for fixed x^{n-1}, they are at most countably many in number. Thus[16]

$$F_n^X(x-|x^{n-1}) < F_n^X(x+|x^{n-1}) \quad \text{only at} \quad x = a_{nk}(x^{n-1}),\ k = 1, 2, \ldots ,$$

and at these points, we define

$$\begin{aligned} p_{nk}(x^{n-1}) &= F_n^X(a_{nk}-|x^{n-1}) \\ q_{nk}(x^{n-1}) &= F_n^X(a_{nk}+|x^{n-1}) \\ &= F_n^X(a_{nk}|x^{n-1}). \end{aligned}$$

Figure 3(b) illustrates the behaviour of $G_n(\cdot|x^{n-1})$ which results from two such points of discontinuity in $F_n^X(\cdot|x^{n-1})$. Also indicated is the effect on $G_n(\cdot|x^{n-1})$ of an interval on which $F_n^X(\cdot|x^{n-1})$ is continuous. The c.d.f. of a uniform $\mathcal{U}[0,1]$ random variable is indicated by the broken line for comparison. It can fairly easily be shown from (17) that, if we adopt the symbol G_0 for the c.d.f. of a $\mathcal{U}[0,1]$ random variable and employ the "indicator function" notation

$$I_S(z) = \begin{cases} 1, & \text{if } z \in S; \\ 0, & \text{otherwise,} \end{cases}$$

so that

$$G_0(u) = uI_{[0,1]}(u) + I_{(1,\infty)}(u),$$

then

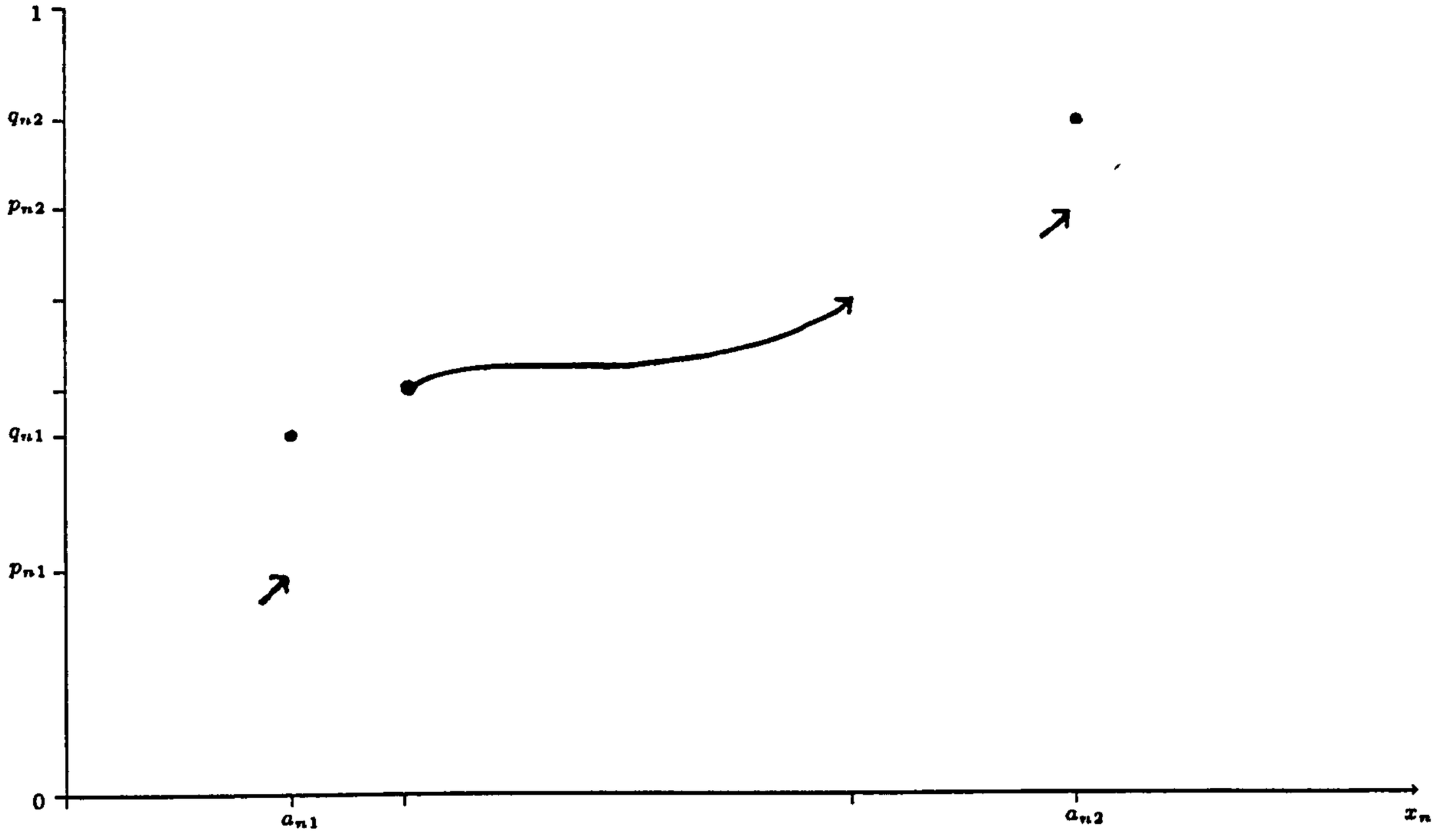
$$G_n(u|x^{n-1}) = \begin{cases} p_{nk}, & \text{if } p_{nk} < u < q_{nk} \text{ for some k;} \\ G_0(u), & \text{otherwise.} \end{cases} \qquad (18)$$

This follows by well known properties of all c.d.f.s : Any $u \in [0,1]$ is either contained in an interval $[p_{nk}, q_{nk}]$ or is the image of a point of continuity of $F_n^X(\cdot|x^{n-1})$. These cases can be considered

[15] which follows from the "continuity" or "σ-additivity" axiom for measures.

[16] It is well known [21] that functions of bounded variation, which include all univariate c.d.f.s, possess at most countably many points of discontinuity, and possess left-handed and right-handed limits everywhere.

a) Conditional c.d.f. $F_n^X(\cdot|x^{n-1})$ of X_n given $X^{n-1} = x^{n-1}$:



b) Conditional c.d.f. $G_n(\cdot|x^{n-1})$ of U_n given $X^{n-1} = x^{n-1}$:

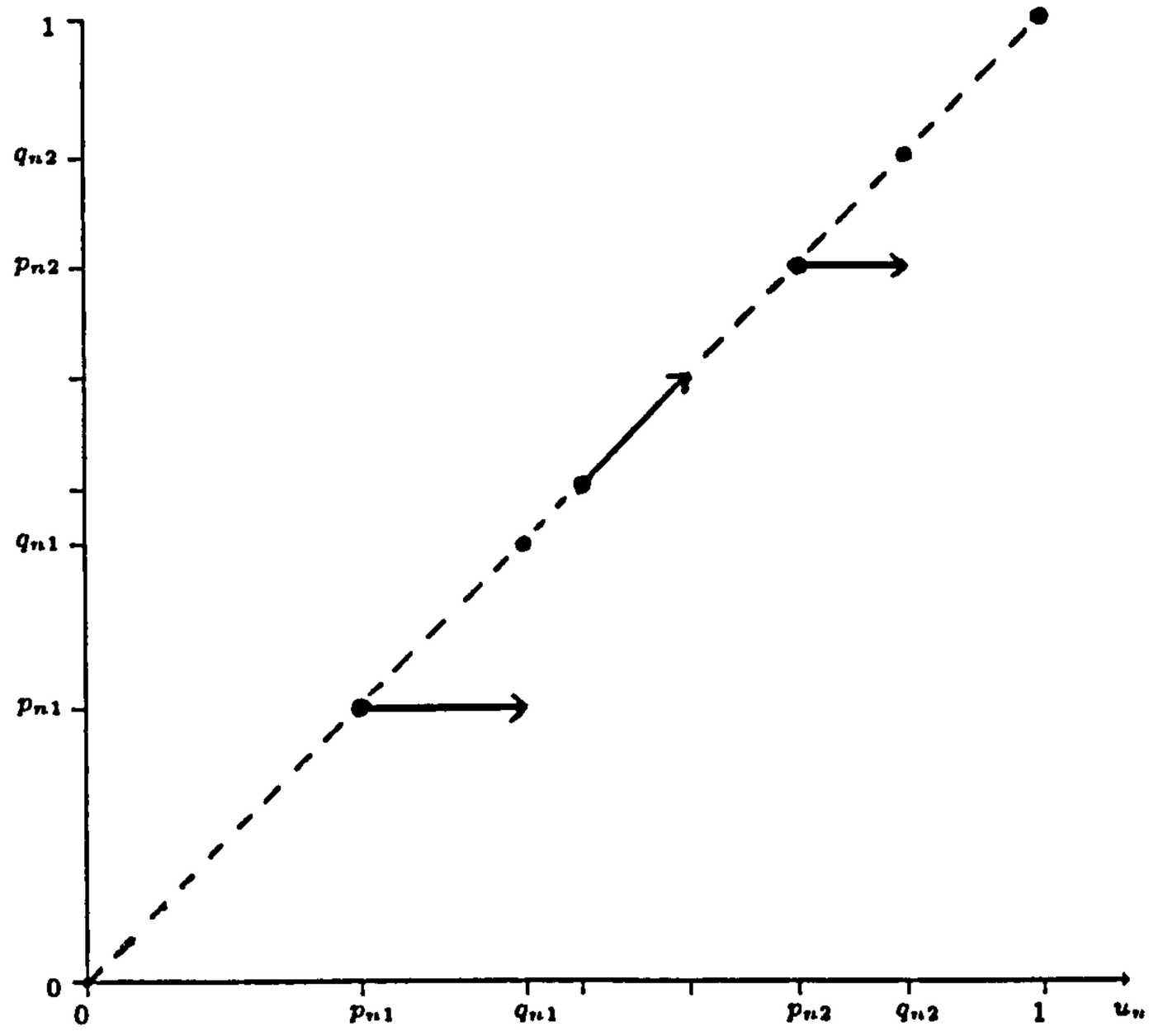


Figure 3: Relationship between predictive c.d.f.s of X_n and U_n

separately, remembering that $F_n^X(\,\cdot\,|x^{n-1})$ is non-decreasing and right-continuous, to arrive at (18). This c.d.f. can be expressed as a sum of a finite or countable number of terms[17]

$$\begin{aligned} G_n(u|x^{n-1}) &= G_0(u) - \sum_k (u - p_{nk}) I_{(p_{nk},q_{nk})}(u) \\ &= G_0(u) - \sum_k \Phi_{nk}(u), \qquad \text{say.} \end{aligned} \tag{19}$$

3.3.1 Biasing Effect of Discontinuities on the U-Distribution

It is now apparent that the conditional distribution of U_n given $X^{n-1} = x^{n-1}$, which is uniform if the predictive c.d.f. is continuous, receives a bias towards larger values of U_n if there are discontinuities in F_n^X, i.e. single points of positive predictive probability, $\mathrm{P}_p\left[X_n = a_{nk}|x^{n-1}\right] > 0$. Perhaps, if the discontinuities are small, the uniform distribution will be a reasonable approximation to $G_n(\,\cdot\,|x^{n-1})$ giving justification to the use of standard u-plots, y-plots and the Kolmogorov-Smirnov test statistic to assess the correctness of the hypothesis that the distributions $\mathcal{P}$ and $\mathcal{Q}$ above are equal. In the application where $\langle X_n \rangle$ is a software failure-count process however, it is hoped that, at least after the initial stages of testing, high predictive probabilities will be assigned to particular discrete values, (e.g $X_n = 0$ or $X_n = 1$), making the assumption that $G_n(u|x^{n-1})$ is the $\mathcal{U}[0,1]$ c.d.f. under the hypothesis $\mathcal{P} = \mathcal{Q}$ more difficult to justify, (see Fig. 3).

3.3.2 The Mean as an Indication of the Seriousness of this Bias

An indication of the level of severity in this effect of discontinuities in the predictive c.d.f. on the conditional distribution of U_n is provided by the conditional expectation

$$\begin{aligned} \mathop{\mathrm{E}}_{p}[U_n|x^{n-1}] &= \int_{u\in\mathbb{R}} u\, dG_n(u|x^{n-1}) \\ &= \int_{u\in[0,1]} u\, du - \sum_k \int_{u\in(p_{nk},q_{nk}]} u\, d\Phi_{nk}, \qquad \text{by (19)} \\ &= \int_0^1 u\, du - \sum_k \left(\int_{p_{nk}}^{q_{nk}} u\, du + q_{nk}(\Phi_{nk}(q_{nk}+) - \Phi_{nk}(q_{nk}-)) \right) \\ &= \tfrac{1}{2} - \sum_k \left(\frac{q_{nk}^2 - p_{nk}^2}{2} + q_{nk}(p_{nk} - q_{nk}) \right) \\ &= \tfrac{1}{2} + \tfrac{1}{2} \sum_k (q_{nk} - p_{nk})^2, \\ &= \tfrac{1}{2} + \tfrac{1}{2} \sum_k \mathop{\mathrm{P}}_{p}\left[X_n = a_{nk}|X^{n-1} = x^{n-1}\right]^2 \end{aligned} \tag{20}$$

which further demonstrates the concluding statement of the previous paragraph.

[17] in the general case of a completely arbitrary continuous, discrete, or mixed scalar predictive c.d.f. $F_n^X(\,\cdot\,|x^{n-1})$

3.4 Alternative Definitions of U

The question arises of whether the definition (15) of the $\langle U_n \rangle$ process could be altered, in cases where F_n^X can possess discontinuities in its first argument, so as to provide distribution properties which are less dependent on n, (through variation in the p_{nk}, q_{nk}), and consequently more amenable to statistical testing. This section discusses properties of some alternative $\langle U_n \rangle$.

3.4.1 An Alternative Definition which Reverses, Rather than Removes, this Bias

The asymmetry of the deviation of $G_n(u|x^{n-1})$, as a function of u, from the function $G_0(u)$, apparent in Figure 3, suggests an alternative definition of $\langle U_n \rangle$ obtained on replacing (15) by

$$U'_n = F_n^X(X_n - | X^{n-1}), \quad n \geq n_0 \,,$$

the left-handed limit $F_n^X(X_n - | X^{n-1}) = \lim_{x \to X_n -} F_n^X(x|X^{n-1})$. In terms of the conditional probabilities of the joint distribution, $\mathcal{P}$, this amounts to changing the definition of u_n, given the realisation x^n, from

$$u_n = \mathop{\mathrm{P}}_{\mathcal{P}}\left[X_n \leq x_n | X^{n-1} = x^{n-1}\right] \tag{21}$$

to

$$u'_n = \mathop{\mathrm{P}}_{\mathcal{P}}\left[X_n < x_n | X^{n-1} = x^{n-1}\right]. \tag{22}$$

The argument of §3.2, slightly amended, now leads to the conditional c.d.f. for U'_n on the hypothesis $\mathcal{P} = \mathcal{Q}$

$$\begin{aligned} G'_n(u|x^{n-1}) &= \sup_x \{F_n^X(x|x^{n-1}) : F_n^X(x-|x^{n-1}) \leq u\} \\ &= \begin{cases} q_{nk}, & \text{if } p_{nk} \leq u < q_{nk} \text{ for some } k; \\ G_0(u), & \text{otherwise,} \end{cases} \\ &= G_0(u) + \sum_k (q_{nk} - u) I_{[p_{nk}, q_{nk})}(u) \,. \end{aligned}$$

With this definition it turns out that U'_n is biased downwards

$$\mathop{\mathrm{E}}_{\mathcal{P}}\left[U'_n | x^{n-1}\right] = \tfrac{1}{2} - \tfrac{1}{2} \sum_k (q_{nk} - p_{nk})^2. \tag{23}$$

It requires only a trivial sign change in the derivation of (20) to demonstrate this.

We mention that another way of viewing the $\langle U'_n \rangle$ is by means of the $\langle U_n \rangle$ for a different process. It can be seen that if $\langle Y_n \rangle$ is a process deterministically related to $\langle X_n \rangle$ by functions continuous and strictly monotonic-decreasing in their first argument $y_n = d_n(x_n; x^{n-1})$, with the induced probability

distribution $\mathcal{P}^Y$, then U'_n from the $\langle X_n \rangle$ process with PFS $\mathcal{P}$ is simply $1 - U_n$ from the $\langle Y_n \rangle$ process with PFS $\mathcal{P}^Y$.

3.4.2 Elimination of the U-Bias

There does not seem to be much to choose between (21) and (22), but at least viewed together they suggest a way of eliminating the bias from the conditional distribution of U_n by simply redefining it as the mean of these two values

$$\begin{aligned} u_n &= \tfrac{1}{2}\left(F_n^X(x_n - | X^{n-1}) + F_n^X(x_n | X^{n-1}) \right) \\ &= \mathop{\mathbf{P}}_{\mathcal{P}}\left[X_n < x_n | X^{n-1} = x^{n-1} \right] + \tfrac{1}{2} \mathop{\mathbf{P}}_{\mathcal{P}}\left[X_n = x_n | X^{n-1} = x^{n-1} \right]. \end{aligned} \tag{24}$$

This results in a conditional c.d.f. when $\mathcal{P} = \mathcal{Q}$

$$\begin{aligned} G_n(u | x^{n-1}) &= \sup_x \{ F_n^X(x | x^{n-1}) : \tfrac{1}{2}\left(F_n^X(x - | x^{n-1}) + F_n^X(x | x^{n-1}) \right) \leq u \} \\ &= \begin{cases} q_{nk}, & \text{if } \tfrac{1}{2}(p_{nk} + q_{nk}) \leq u < q_{nk} \text{ for some } k; \\ p_{nk}, & \text{if } p_{nk} < u < \tfrac{1}{2}(p_{nk} + q_{nk}) \text{ for some } k; \\ G_0(u), & \text{otherwise,} \end{cases} \end{aligned}$$

illustrated in Figure 4, and a conditional expectation, $\mathbf{E}_{\mathcal{P}}\left[U_n | x^{n-1} \right] = \frac{1}{2}$, which is independent of n and unaffected by discontinuities in the predictive c.d.f., making this definition of $\langle U_n \rangle$ preferable to the previous two. In Figure 4 the PFS remains as in Figure 3(a): it is the change from equation (21) to (24) which is responsible for Figure 4 replacing 3(b).

3.4.3 A Formalisation in Terms of Integration by Parts

In fact, (20), (23) and the last result can be rigorously obtained in the general mixed-distribution case directly from the integration by parts formula for Lebesgue-Stieltjes integrals,

$$\int_S \mathrm{G}_+ \, d\mathrm{H} + \int_S \mathrm{H}_- \, d\mathrm{G} = \int_S d(\mathrm{GH}), \tag{25}$$

where the function definitions are $\mathrm{G}_+ : x \mapsto \mathrm{G}(x+)$, $\mathrm{H}_- : x \mapsto \mathrm{H}(x-)$, and $\mathrm{GH} : x \mapsto \mathrm{G}(x)\mathrm{H}(x)$ and G and H are defined and of bounded variation on a (finite or infinite) interval containing S [21][18].

[18]The Lebesgue-Stieltjes measure [21, p156] implied by the symbol dG above is defined independently of the values of the function G at its points of discontinuity, being determined rather in terms of the one-sided limits G_+ and G_- (It is, trivially, invariant also under addition of a scalar constant to G). This measure may be employed as a rigorous formalisation of the construction of a probability distribution given any function having the basic properties of a c.d.f. (although it equally has application to the definition of signed and non-finite measures). It can be obtained by a general *measure extension* procedure once it has been defined on finite intervals by definitions such as $\mu_G((a, b]) = \mathrm{G}_+(b) - \mathrm{G}_+(a)$, etc. The measures, integrals, and the integration-by-parts equation (25) are guaranteed to apply at least to all Borel subsets S of the interval over which the functions G and H are defined.

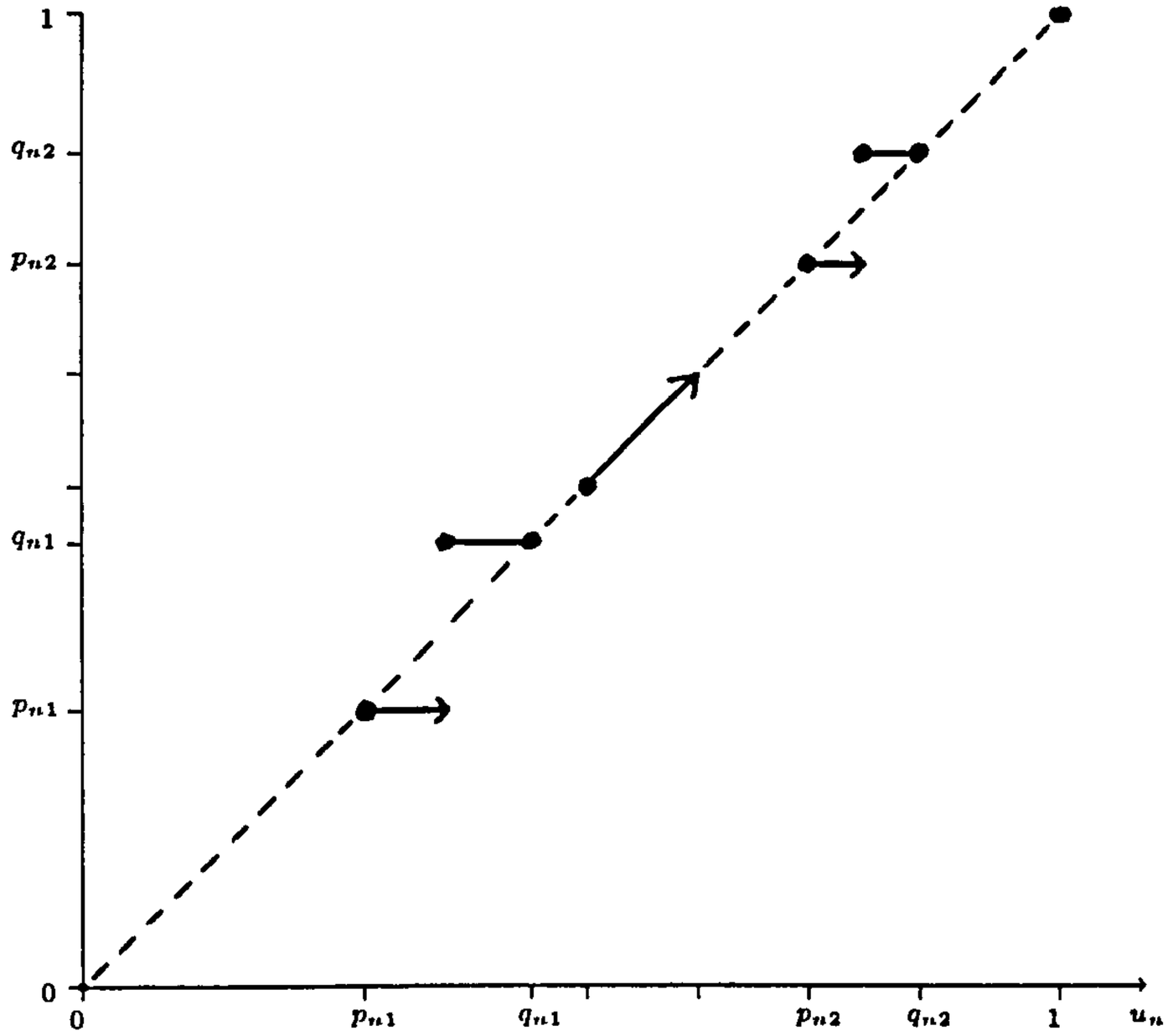


Figure 4: Conditional c.d.f. of unbiased version of U_n

For any function, H, of bounded variation over an interval containing S, it follows by putting G = H in (25) that

$$\int_S \mathrm{H}_+ \, d\mathrm{H} + \int_S \mathrm{H}_- \, d\mathrm{H} = \int_S d\mathrm{H}^2 \, .$$

Also, from basic definitions, [21],

$$\begin{aligned} \int_S \mathrm{H}_+ \, d\mathrm{H} - \int_S \mathrm{H}_- \, d\mathrm{H} &= \int_S \mathrm{H}_+ - \mathrm{H}_- \, d\mathrm{H} \\ &= \sum_k \int_{\{a_k\}} \mathrm{H}_+ - \mathrm{H}_- \, d\mathrm{H}, \qquad \{a_k : k = 1, 2, \ldots\} \text{ being the discontinuity points of H in } S, \\ &= \sum_k (\mathrm{H}(a_k+) - \mathrm{H}(a_k-))^2 \end{aligned}$$

and it follows that

$$\begin{aligned} \int_S \mathrm{H}_+ \, d\mathrm{H} &= \tfrac{1}{2} \left\{ \textstyle\int_S d\mathrm{H}^2 + \sum_k (\mathrm{H}(a_k+) - \mathrm{H}(a_k-))^2 \right\} \\ \int_S \mathrm{H}_- \, d\mathrm{H} &= \tfrac{1}{2} \left\{ \textstyle\int_S d\mathrm{H}^2 - \sum_k (\mathrm{H}(a_k+) - \mathrm{H}(a_k-))^2 \right\} \end{aligned}$$

which if applied to the function $F_n^X(\cdot|x^{n-1})$ over $S = \mathbb{R}$, gives the results obtained above for the conditional expectations of U_n under the three alternative definitions.

Alternatively these expectations can be obtained by inspection of the graphs of the c.d.f.s (see Figures 3 and 4 using the well known result that for a c.d.f. F of any non-negative random variable

X

$$\int_0^\infty x dF(x) = \int_0^\infty R(x)dx \tag{26}$$

where $R(x) = 1 - F(x)$ is the reliability function of X. Thus the mean of the random variable is equal to the area bounded by the vertical axis "$x = 0$", the line "$y = 1$", and the graph of F (where discontinuities would be joined by vertical lines.) The fact that this identity (26) continues to hold for arbitrary non-negative random variables can actually be shown from (25), and some additional reasoning, on putting $\mathrm{G}(x) = x$ and $\mathrm{H}(x) = 1 - F(x)$. (Essentially the same result was also quoted as equation (1) on p12.) This enables the three results about bias to be verified by inspection of the graphs of the conditional c.d.f.s of U_n.

3.4.4 A Randomised Definition of U Removing Both Bias *and Discontinuity*

A further modification to the definition of $\langle U_n \rangle$, reminiscent of procedures used for handling discontinuities in other contexts (e.g. [41, §20.22]), is to define U_n to be a point selected *randomly* from within the interval $[F_n^X(x_n - | x^{n-1}), F_n^X(x_n | x^{n-1})]$, instead of using an end-point or the mid-point of this interval as in the three definitions discussed so far. For this purpose, assume that $\langle \xi_n \rangle$ is an i.i.d. uniform $\mathcal{U}[0,1]$ process which is assumed independent of the $\langle X_n \rangle$ process (a realisation of $\langle \xi_n \rangle$ could be generated by the observer), and redefine

$$u_n = (1 - \xi_n) F_n^X(x_n - | x^{n-1}) + \xi_n F_n^X(x_n | x^{n-1}). \tag{27}$$

To obtain the resulting revised c.d.f., $G_n(u_n | x^{n-1})$, on the hypothesis $\mathcal{P} = \mathcal{Q}$, we fix a number u, and consider probabilities, conditional on $X^{n-1} = x^{n-1}$, of events in the space of (X_n, ξ_n) as follows:–

$$\begin{aligned}
E &: \quad (1 - \xi_n) F_n^X(X_n - | x^{n-1}) + \xi_n F_n^X(X_n | x^{n-1}) \leq u; \\
E_1 &: \quad F_n^X(X_n | x^{n-1}) \leq u; \\
E_2 &: \quad F_n^X(X_n - | x^{n-1}) \leq u < F_n^X(X_n | x^{n-1}) \wedge \xi_n \leq \frac{u - F_n^X(X_n - | x^{n-1})}{F_n^X(X_n | x^{n-1}) - F_n^X(X_n - | x^{n-1})}.
\end{aligned}$$

Then $E = E_1 \cup E_2$, $E_1 \cap E_2 = \emptyset$, $\mathbf{P}[E_1]$ is given by the RHS of (18), and

$$\mathbf{P}[E_2] = \begin{cases} (q_{nk} - p_{nk}) \times \dfrac{u - p_{nk}}{q_{nk} - p_{nk}}, & \text{if } u \in [p_{nk}, q_{nk}) \text{ for some } k; \\ 0, & \text{otherwise.} \end{cases}$$

It follows that

$$G_n(u | x^{n-1}) = \mathbf{P}[E] = \mathbf{P}[E_1] + \mathbf{P}[E_2] = G_0(u), \tag{28}$$

i.e., with this "randomised" definition, U_n is conditionally a uniform $\mathcal{U}[0,1]$ random variable, given x^{n-1}.

In one sense this final definition of the process $\langle U_n \rangle$ seems preferable to those preceding it in that it has recaptured the uniform-conditional-c.d.f. property of U_n, which held in the case of continuous predictive distributions but at first appeared to have been lost in attempting to extend the application to a PFS producing arbitrary predictive distributions. It is also apparent, however, that in this last definition a random component—whose value is meaningless as far as the performance of the PFS is concerned—has been added to the $\langle U_n \rangle$. This must have some detrimental effect on the usefulness of, for example, a u-plot used as a means of capturing graphically something of the nature of the bias in the output from a PFS applied to a particular data set. Also, if the sequence, $\langle U_n \rangle$, were incorporated in a recalibration procedure, it is to be expected that any resulting recalibrated PFS would be subject to extra noise effects from the $\langle \xi_n \rangle$.

In the next section, a "Modified U-Plot" is suggested which makes use of the *idea* of the above $\langle U_n \rangle$, *without* requiring a *realisation* of $\langle \xi_n \rangle$. Thus the doubts mentioned in the previous paragraph do not arise. A further theoretical interpretation of this modified plot is attempted in §3.6, and in Chapter 4 results are presented based on this plot, after additional modifications have been developed also in §3.6 and Appendix B.

3.5 Modified U-plot

As mentioned at the end of §3.4, a plot can be produced via the definition (27) of $\langle U_n \rangle$ without having to generate a noise sequence $\langle \xi_n \rangle$. The definition of this plot can be presented as an extension of the definition already familiar for the continuous F_n^X case by, rather artificially, thinking of the familiar definition in terms of *posterior c.d.f.s* of the $U_i; i = 1, \ldots, n$. Later on p78, it is suggested that this device indicates an approach to recalibration of predictions in other more general contexts. In the familiar continuous case, (27) reduces to $u_n = F_n^X(x_n|x^{n-1})$ which is the standard definition of u_n. Then, the standard u-plot, regarded as a function, is the sample distribution function of $\langle u_i \rangle_{i=1}^n$ (which we continue to write as u^n.) This function, $S_n : \mathbb{R} \to [0,1]$, can be expressed using the Heaviside notation as

$$S_n(u) = \sum_{i=n_0}^{n} w_i H(u - u_i), \qquad \text{where } w_i = \tfrac{1}{n-n_0+1},$$

which trivially can be rewritten

$$S_n(u) = \sum_{i=n_0}^{n} w_i G_i(u|x^i), \tag{29}$$

where $G_i(\cdot|x^i)$ is used to denote the c.d.f. (under the assumption that $\mathcal{P} = \mathcal{Q}$) of U_i, conditional on $X^i = x^i$. Expressed in this form the definition will extend to the general mixed F_n^X case, i.e. (29) can be used as the basis of a Modified U-Plot defined for any scalar PFS applied to any data with U_n defined now by (27), (and therefore possibly not an observable quantity). Thus $G_n(\cdot|x^n)$ will be the c.d.f. of a uniform $\mathcal{U}[p_{nk}, q_{nk}]$ random variable, whenever $x_n = a_{nk}$ for some k, and a constant random variable otherwise. This follows from the fact that (27) defines U_n such that U_n, conditioned on $X^m = x^m$, is distributed $\mathcal{U}[0,1]$ for any $m < n$, and $\mathcal{U}[F_n^X(x_n-|x^{n-1}), F_n^X(x_n|x^{n-1})]$ for any $m \geq n$, (provided $\mathcal{P} = \mathcal{Q}$). So instead of thinking of U_n as a deterministic function of X^n, we think of it as having a posterior c.d.f. given $X^n = x^n$ which, depending on x^n, may or may not turn out to be a Heaviside function. The "u-plot" is defined in terms of these posterior c.d.f.s which take the form

$$G_i(u|x^i) = \begin{cases} 1, & \text{if } u \geq F_i^X(x_i|x^{i-1}); \\ \frac{u - F_i^X(x_i-|x^{i-1})}{F_i^X(x_i|x^{i-1}) - F_i^X(x_i-|x^{i-1})}, & \text{if } F_i^X(x_i-|x^{i-1}) \leq u < F_i^X(x_i|x^{i-1}); \\ 0, & \text{if } u < F_i^X(x_i-|x^{i-1}). \end{cases} \tag{30}$$

The resulting S_n is equal to a mixture of cumulative probability distribution functions, and must therefore itself be a c.d.f. Moreover if $\mathcal{P} = \mathcal{Q}$, then S_n is a random function with, for u any fixed number,[19]

$$\begin{aligned} \underset{X^n}{\mathbf{E}}[S_n(u)] &= \sum_{i=n_0}^{n} w_i \underset{X^n}{\mathbf{E}}[G_i(u|X^i)] \\ &= \sum_{i=n_0}^{n} w_i \underset{X^i}{\mathbf{E}}[G_i(u|X^i)] \\ &= \sum_{i=n_0}^{n} w_i \underset{X^{i-1}}{\mathbf{E}}\left[\underset{x_i|X^{i-1}}{\mathbf{E}}\left[\underset{\epsilon_i|X^i}{\mathbf{P}}[U_i \leq u]\right]\right] \\ &= \sum_{i=n_0}^{n} w_i \underset{X^{i-1}}{\mathbf{E}}\left[\underset{\epsilon_i, x_i|X^{i-1}}{\mathbf{P}}[U_i \leq u]\right] \\ &= \sum_{i=n_0}^{n} w_i \underset{X^{i-1}}{\mathbf{E}}[G_i(u|X^{i-1})] \\ &= \sum_{i=n_0}^{n} w_i \underset{X^{i-1}}{\mathbf{E}}[G_0(u)], \qquad \text{by (28)} \\ &= G_0(u). \end{aligned} \tag{31}$$

So it has been shown that a function, S_n, can be defined by means of (27) and (29) which—regarded as a Modified U-Plot—extends to the case of a general PFS, the definition used in [2], whilst

[19] The notations $\underset{V}{\mathbf{E}}[\cdot]$, $\underset{V}{\mathbf{P}}[\cdot]$, $\underset{V|W}{\mathbf{E}}[\cdot]$ and $\underset{V|W}{\mathbf{P}}[\cdot]$, where V and W are random variables, are used here to mean $\mathbf{E}[\cdot]$, $\mathbf{P}[\cdot]$, $\mathbf{E}[\cdot|W]$ and $\mathbf{P}[\cdot|W]$, respectively, explained in Chapter 2 p10. Thus, strictly speaking, given that we are assuming for the joint process (X_n, ζ_n) an underlying probability law with respect to which expectations and probabilities are defined, the "V" in these expressions is superfluous but is included as a reminder that the numerical value of the random variable or expression inside the square brackets is fully determined by a realisation of the pair (V, W).

preserving the property of expectations lying on the 45°-line under the hypothesis that $\mathcal{P} = \mathcal{Q}$. It is also of interest to see how this plot relates to that which would have been obtained had the $\langle \xi_n \rangle$ been explicitly generated. We can reexpress (29) as

$$\begin{aligned} S_n(u) &= \sum_{i=n_0}^{n} w_i \mathop{\mathbf{P}}_{\xi^n|x^n}[U_i \leq u] \qquad (U_i \text{ still defined by (27)}) \\ &= \sum_{i=n_0}^{n} w_i \mathop{\mathbf{P}}_{\xi^n|x^n}[U_i \in (-\infty, u]] \\ &= \sum_{i=n_0}^{n} w_i \mathop{\mathbf{E}}_{\xi^n|x^n}\left[I_{(-\infty,u]}(U_i)\right] \\ &= \sum_{i=n_0}^{n} w_i \mathop{\mathbf{E}}_{\xi^n|x^n}[H(u - U_i)] \\ &= \mathop{\mathbf{E}}_{\xi^n|x^n}\left[\sum_{i=n_0}^{n} w_i H(u - U_i)\right]. \end{aligned} \tag{32}$$

So the definition (29), (30) is equivalent to assigning $S_n(u)$ to be the *expectation* of the value which would have been obtained from the same realisation x^n, by *actually generating* a realisation of ξ^n, and using the sample c.d.f. of the resulting u^n values obtained from (27). Equations (31) and (32) and the attempt at the end of this section to further emphasize the relationship to the continuous case by relating the Modified U-Plot to the u-plot of a hypothetical underlying process for which continuous predictive c.d.f.s would be appropriate, suggest using some measure of "distance" between S_n and G_0 as a comparative measure of predictive performance of different PFSs on a common data set, just as has been done, in for example [2], for continuous inter-failure time prediction. However, the present work provides no basis for carrying out a formal statistical hypothesis test of $\mathcal{P} = \mathcal{Q}$, from the Modified U-Plot alone[20] in the general mixed (or discrete) case since, although the point-wise expectation of $S_n(u)$ has been shown by (31) to be u, still, the effect on the distribution of the Kolmogorov distance for plot (29), has not been analysed when $H(u - u_i)$ is replaced by equation (30)'s $G_i(u|x^i)$, for those u_i which are incompletely observed. The Kolmogorov distance is therefore interpreted as an indicator of predictive quality in the same way as in [2], *although* now without the force of a formal statistical hypothesis test. As to the *sign* of the deviation, it should be born in mind that failure counts are a kind of reciprocal observation of inter-failure times, so that *optimistic* predictors of failure-count, (those which over-estimate future software reliability), will have u-plots which deviate *below* the 45°-line—which would indicate pessimism if the predictions had been of inter-failure times, as in [2].

In appearance the plot described above is piece-wise linear, having gradient changes at points

[20] Of course a formally correct *randomised* test could be obtained using the u-plot produced by actually generating the ξ_n of equation (27).

$u = p_{ik}$ and q_{ik} for those i with $x_i = a_{ik}$ for some k; and having discontinuities of size w_i at $u = u_i$ for other i. Its slope at u, obtained from (29), is

$$\frac{dS_n}{du} = \sum_{\substack{n_0 \le i \le n \\ u \in (p_{ik}, q_{ik}), \\ \text{where } x_i = a_{ik}}} \frac{w_i}{(q_{ik} - p_{ik})},$$

where defined. If in the application of a PFS, every observed value x_i has been correctly predicted on the basis of x^{i-1}, with non-zero probability, then the plot here described will be a continuous function. This is the likely situation with software failure-count data[21]. Examples of plots obtained are presented in Appendix A.

3.5.1 An Alternative Interpretation in Terms of a Hypothetical Continuous-Valued Process

Before going on to describe its use for recalibration in §3.6, we mention an alternative way of viewing the $\langle U_n \rangle$ sequence defined by (27), which gives more meaning to the introduction of the random quantity ξ_i. The basic idea here is to suggest that there may be a quantity underlying the observations which can be represented as a process $\langle Z_n \rangle$ whose conditional c.d.f.s are continuous. The $\langle X_n \rangle$ could then be thought of as that part of $\langle Z_n \rangle$ which is available to the observer, i.e. as the result of incomplete observation of the process $\langle Z_n \rangle$. ξ_n would represent the information lost in the step from Z_n to X_n. Suppose that the following conditions hold:–

(i) $\langle X_n, Z_n \rangle$ is a two dimensional random process, $\langle Z_n \rangle$ having marginal joint distribution $\mathcal{P}^Z$;

(ii) For each n, X_n is a deterministic function of Z^n by means of a sequence of functions $\langle v_n \rangle$ each non-decreasing in its first argument

$$X_1 = v_1(Z_1),\ X_2 = v_2(Z_2, X_1),\ \ldots, X_n = v_n(Z_n, X^{n-1}),\ \ldots;$$

(iii) The marginal probability distribution for $\langle X_n \rangle$, which is determined by $\mathcal{P}^Z$ and the functions $\langle v_n \rangle$, is $\mathcal{P}$.

Figure 5 illustrates the function $v_n(\cdot, x^{n-1})$ which maps the interval $[b_{n0}, c_{n0}]$ onto the single point a_{n0}, the interval (c_{n0}, b_{n1}) monotonically onto (a_{n0}, a_{n1}), and $[b_{n1}, c_{n1}]$ onto the point a_{n1} where it is assumed that a_{n0} and a_{n1} are two adjacent points at which positive predictive probability is

[21] Exceptions do exist, such as ML plug-in based PFSs in which the size of the initial fault population is a finite (or at least allowably finite) model parameter: then a failure count may be observed after having a zero assigned predictive probability. The JM model on p37 is an example.

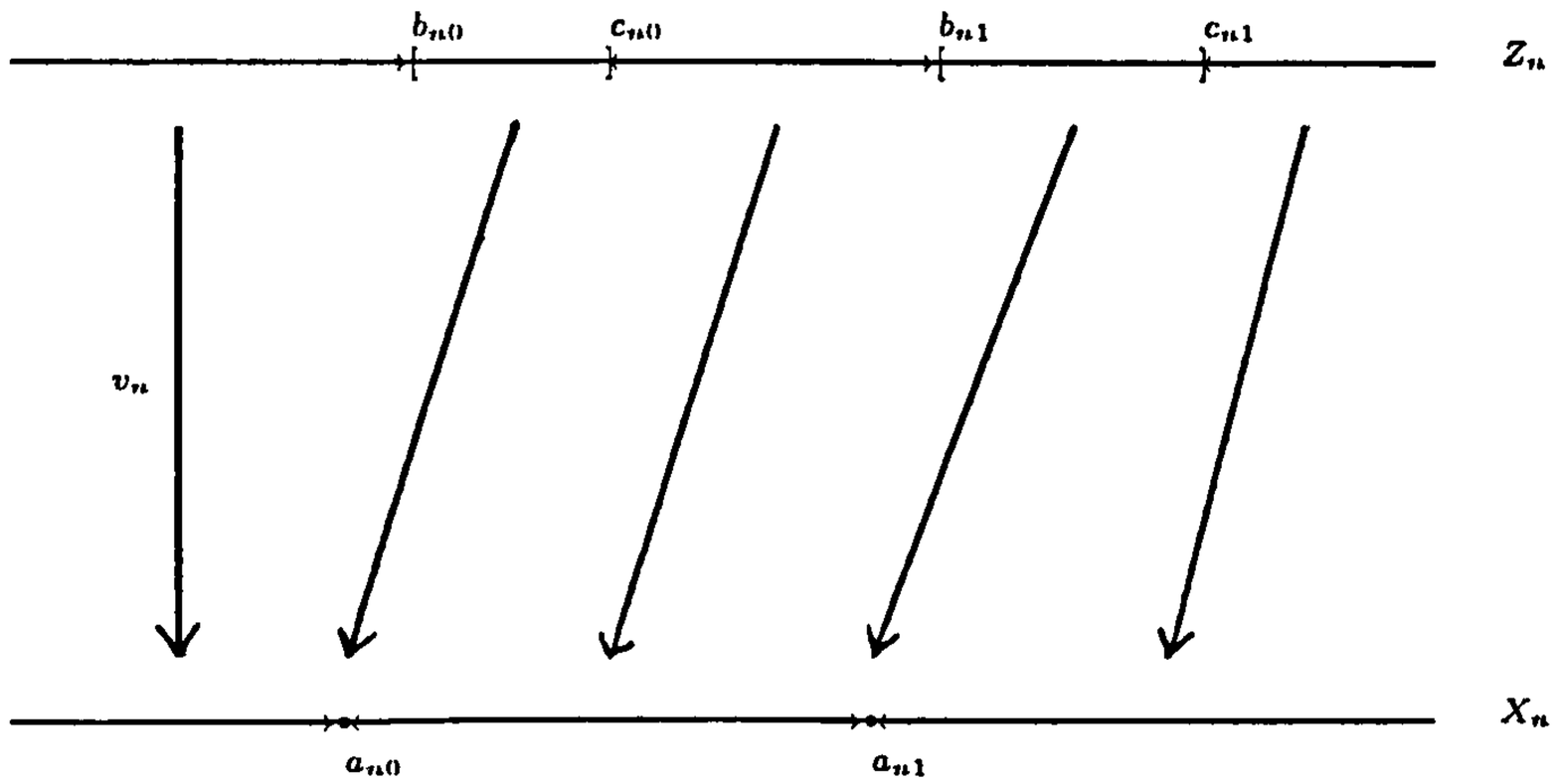


Figure 5: Interpretation of ξ_n in terms of a hypothetical underlying continuously distributed quantity Z_n

assigned, by $\mathcal{P}$, to X_n given that $X^{n-1} = x^{n-1}$. The points b_{ni} and c_{ni} are allowed to depend on x^{n-1}. It now follows from the assumption that $\mathcal{P}^Z$ has continuous predictive c.d.f.s

$$F_n^Z(z_n|z^{n-1}) = \underset{\mathcal{P}^Z}{\mathbf{P}}\left[Z_n \leq z_n | Z^{n-1} = z^{n-1}\right],$$

that the associated $\langle U_n^Z \rangle$ process (defined according to the original form (15) with Z replacing X), has conditional c.d.f.s $\mathbf{P}_{\mathcal{P}^Z}\left[U_n^Z \leq u | Z^{n-1} = z^{n-1}\right] = G_0(u)$. It then follows that (30) above also defines the conditional c.d.f. of U_n^Z given x^n, i.e. (27) defines U_n having the same conditional distribution given x^n as has U_n^Z, defined according to the standard definition, (15), for the hypothetical process $\langle Z_n \rangle$,

$$\underset{\mathcal{P}}{\mathbf{P}}[U_n \leq u | x^n] = \underset{\mathcal{P}^Z}{\mathbf{P}}\left[U_n^Z \leq u | x^n\right].$$

This suggests that, provided it is agreed to accept the existence of such an unobserved process $\langle Z_n \rangle$, then U_n defined by (27) can be identified with U_n^Z, defined in the original manner, for the $\langle Z_n \rangle$ process. Thus the modified u-sequence is nothing more than the familiar u-sequence for prediction of the process $\langle Z_n \rangle$ using the PFS $\mathcal{P}^Z$. And the modified u-plot $s \mapsto S_n(s)$ is the point-wise posterior expectation of the ordinary u-plot (29) of the incompletely observed $\langle Z_n \rangle$ process with respect to this "extended" PFS $\mathcal{P}^Z$.

3.6 Further Modifications for Recalibration

In this section the use of the Modified U-Plot described in §3.5 for improving failure-count predictions by recalibration techniques is discussed. There are certain problems with the idea of recalibrating

failure-count predictions which are mentioned—for example, the problem of unequal duration of failure-count time intervals. Arguments for attempting to use some kind of recalibration technique despite these problems, are the consistent success achieved in the inter-failure time case, [59], and the problem mentioned in [2] of the apparent non-existence of widely applicable "best" raw software reliability growth models. In view of such problems this section is intended to comprise an investigation of some techniques for recalibrating a PFS, complemented by a data analysis in Chapter 4 indicating the effectiveness of these techniques. Some success in recalibration of failure-count predictions is evident in the numerical results of §4.2 for both simulated and real failure data. In fact, both the definition of the Modified U-Plot and its method of use for the purpose of recalibrating predictive distributions could be applied as described here in the general case of arbitrary *mixed* predictive distributions. However numerical results have been obtained only for the case of the purely discrete predictive c.d.f.s appropriate for prediction of failure-count processes. For the other special case in which the predictive c.d.f.s are purely continuous, both the Modified U-Plot and the technique of recalibration given in this chapter become equivalent to those discussed in [11] and [59], where they are applied to inter-failure time data. In addition, the two further enhancements to the u-plot, described below, and intended to improve its use in recalibration, remain applicable in the general mixed case and, in particular, might be useful in the case of inter-failure time prediction—although perhaps slightly less so than they are shown to be in the failure-count context by the numerical results of §4.2. The second of these two enhancements—the technique of introducing tighter constraints on the derivative of the smoothed plot—is probably most beneficial when the number of past predictions (i.e. the number of "u"s observed) is small, which will more commonly occur with failure-count data than with individual inter-failure times.

The description of the recalibration technique which will be given later in this section, including the two optional modifications mentioned, originates from the following general understanding of the role of recalibration in improving predictive distributions, (c.f. [6, 10, 11, 20, 59, 7] and [56]):

3.6.1 A PFS $\mathcal{P}$ as : (i) a Probability Model for $\langle X_n \rangle$; or (ii) a Transformation between $\langle X_n \rangle$ and $\langle U_n \rangle$

An observer generates one-step-ahead predictive c.d.f.s $\langle F_n^X(\cdot|x^{n-1}) \rangle$ for a process, $\langle X_n \rangle$, using a raw PFS which, as explained in §3.2, is here assumed equivalent to a joint distribution, $\mathbf{P}_{\mathcal{P}}$, for $\langle X_n \rangle$, with the conditional c.d.f.s of $\mathbf{P}_{\mathcal{P}}$ equal to the predictive c.d.f.s from the PFS. However, the fact that the observer is prepared to consider a recalibrated version of this PFS indicates that the observer is treating $\mathcal{P}$ as a provisional probabilistic model only, and retains a "higher level" model of the

situation. According to this higher model $\langle X_n \rangle$ arises from a true[22] joint probability distribution, $\mathbf{P}_{\mathcal{Q}}$, which is unknown and from which $\mathbf{P}_{\mathcal{P}}$ may differ. Thus, on moving to the higher model, the role of $\mathcal{P}$ is changed from being the probabilistic law for the process $\langle X_n \rangle$. Rather, $\mathcal{P}$ can be loosely regarded as defining a transformation, known to the observer (via (27) in our generalised-u case), between the processes $\langle X_n \rangle$ and $\langle U_n \rangle$. Therefore any value of $\mathcal{Q}$, the unknown true joint distribution for $\langle X_n \rangle$, should induce a similarly unknown joint distribution, $\mathcal{S}^{\mathcal{Q}}$ say, for the $\langle U_n \rangle$ process

$$\langle \langle X_n \rangle, \mathcal{Q} \rangle \overset{\mathcal{P}}{\longleftrightarrow} \langle \langle U_n \rangle, \mathcal{S}^{\mathcal{Q}} \rangle. \tag{33}$$

The result of this transformation was considered in §§3.2 & 3.4 only for the special case $\mathcal{Q} = \mathcal{P}$,

$$\langle \langle X_n \rangle, \mathcal{P} \rangle \overset{\mathcal{P}}{\longleftrightarrow} \langle \langle U_n \rangle, \mathcal{S}^{\mathcal{P}} \rangle,$$

say.

3.6.2 Recalibration Viewed as Replacement of the model for $\langle U_n \rangle$

The recalibrated PFS can now be viewed as resulting from a decision to replace the PFS $\mathcal{S}^{\mathcal{P}}$ for $\langle U_n \rangle$—which would be logically consistent with belief in $\mathcal{P}$ for $\langle X_n \rangle$—by a crude PFS, $\mathcal{S}^*$ say, for $\langle U_n \rangle$ which is chosen by the observer independently of any specific modelling assumptions which were involved in the original justification of the raw PFS $\mathcal{P}$ for $\langle X_n \rangle$. Having replaced $\mathcal{S}^{\mathcal{P}}$ by $\mathcal{S}^*$ as the probabilistic model for $\langle U_n \rangle$, the recalibrated PFS, $\mathcal{P}^*$ say, for the $\langle X_n \rangle$ process becomes determined by a requirement for consistency of the probabilistic model representations of the PFSs for the two processes $\langle X_n \rangle$ and $\langle U_n \rangle$,

$$\langle \langle X_n \rangle, \mathcal{P}^* \rangle \overset{\mathcal{P}}{\longleftrightarrow} \langle \langle U_n \rangle, \mathcal{S}^* \rangle.$$

(N.B. Here, the definition of each *realisation* $\langle u_n \rangle$ in terms of $\langle x_n \rangle$ has remained unchanged, still being given by (27) using the unchanged raw PFS, $\mathcal{P}$.)

Some Requirements for a Fully Rigorous Argument

The above description is intended to motivate the recalibration techniques described below by emphasizing the central task of *choosing* a PFS $\mathcal{S}^*$ for the $\langle U_n \rangle$ process independently of the assumptions involved in the raw PFS, $\mathcal{P}$. However, it is not strictly accurate—particularly for the cases where definition (27) is used for a PFS $\mathcal{P}$ which can assign concentrated predictive probability. In order to provide a rigorous development (which has not been done here) along the same lines, in the general case, the following problems would need to be dealt with:–

[22] But see remarks in §3.2.3.

(i) The equivalence of a PFS (i.e. sequence of c.d.f.s $\langle F_n^X(\cdot|\cdot)\rangle$) to a joint probability distribution, as mentioned on p48, is not 1–1. Many sequences of conditional c.d.f.s may define the same joint distribution. We *can* assume, provided the sequence $\langle F_n^X(\cdot|\cdot)\rangle$ may correctly be regarded as conditional c.d.f.s consistent with the probability measure $\mathbf{P}_\mathcal{P}$ (defined by equation (16)), that, by standard properties of conditional probability[23] the sequence of functions $\langle F_n^X(\cdot|x^{n-1})\rangle$ which arise for a given realisation of $\langle X_n\rangle$ will be, with $\mathcal{P}$-probability one, identical to that arising from any given *other PFS consistent in the same way with the joint distribution,* $\mathbf{P}_\mathcal{P}$. However, the phrase "with probability 1" may be less reassuring than usual in our situation here since this probability is $\mathbf{P}_\mathcal{P}$-probability, and we are *not* assuming $\mathbf{P}_\mathcal{P}$ to be the *true* distribution of $\langle X_n\rangle$[24].

(ii) Even after fixing upon a single sequence of c.d.f. functions corresponding to $\mathcal{P}$, the transformation between x^n and u^n provided by (27) is not deterministic, because of ξ_i, in cases where one of the x_i has been predicted with positive probability. It is also the case in general that many n-vectors x^n may map onto one u^n since a c.d.f. function $F_i^X(\cdot|x^{i-1})$ is not necessarily 1–1. All we can say with certainty about the general case is that any sequence of functions corresponding to $\mathcal{P}$ defines a correspondence between any x^n and a set of possible u^n, and between any u^n and a set of possible x^n.

The form of the relationship between the PFS $\mathcal{S}^*$, (sequence of functions $\langle G_n^{\mathcal{S}^*}(u_n|u^{n-1})\rangle$, say), and the recalibrated PFS $\mathcal{P}^*$, (say $\langle F_n^{*X}(x_n|x^{n-1})\rangle$), can be obtained by noting that, given the observation $X^{n-1} = x^{n-1}$, the two events: $X_n \leq x_n$, and $U_n \leq F_n^X(x_n|x^{n-1})$ are identical. (Here there is a problem from (ii) above if $F_n^X(\cdot|x^{n-1})$ is constant on an interval extending to the right from x_n.) Then, provided we read the condition "$\cdot|u^{n-1}$" as "given that u^{n-1} lies in a set consistent only with $X^{n-1} = x^{n-1}$", we can derive

$$\begin{aligned} F_n^{*X}(x_n|x^{n-1}) &= \underset{\mathcal{P}^*}{\mathbf{P}}\left[X_n \leq x_n|x^{n-1}\right] \\ &= \underset{\mathcal{S}^*}{\mathbf{P}}\left[U_n \leq F_n^X(x_n|x^{n-1})|u^{n-1}\right] \end{aligned}$$

---

[23] A countable intersection of probability-one events has probability one. So, it suffices to show, that for each *individual* integer n and (using also right-continuity of c.d.f.s) for each *rational* first argument x_n (to the left of the '|'), the two predictive c.d.f.s will agree (as functions of the vector x^{n-1} to the right of the '|') with $\mathbf{P}_\mathcal{P}$-probability 1. This weaker statement will follow after verifying that assumption (16) for a process measure $\mathbf{P}_\mathcal{P}$ implies that, for any *fixed* number x_n, the c.d.f. $F_n^X(x_n|\cdot)$, when regarded as a function of the vector argument to the right hand side of the '|', satisfies the properties of a 'Radon-Nikodym derivative' of the measure $E \mapsto \mathbf{P}_\mathcal{P}\left[\{\omega : X_n(\omega) \leq x_n \wedge X^{n-1}(\omega) \in E\right]$ with respect to the measure $E \mapsto \mathbf{P}_\mathcal{P}\left[\{\omega : X^{n-1}(\omega) \in E\right]$. Now rely on the theorem [21, p139] that two equivalent Radon-Nikodym derivatives can only differ on a zero-measure subset of the domain of their argument variable—in this case of the variable x^{n-1}, where the measure concerned in our case here (i.e. the second of the above two measures) is clearly just $\mathbf{P}_\mathcal{P}$-probability.

[24] In measure theory terminology, we might even *not* have absolutely continuity [21, p139] of $\mathbf{P}_\mathcal{Q}$ relative to $\mathbf{P}_\mathcal{P}$, in which case having to ignore a "$\mathbf{P}_\mathcal{P}$-probability 0" collection of exceptional cases might represent a significant weakness if that collection of process realisations has *positive* $\mathbf{P}_\mathcal{Q}$-probability.

$$= G_n^{\mathcal{S}^*}(F_n^X(x_n|x^{n-1})|u^{n-1}), \tag{34}$$

which states the familiar result [59] that the recalibrated predictive c.d.f. of X_n given x_{n-1} is obtained by the function-composition of the observer's predictive c.d.f. $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$ for U_n with the raw predictive c.d.f. $F_n^X(\cdot|x^{n-1})$ for X_n given $X_{n-1} = x_{n-1}$. This recalibration procedure is the same as that used in the references cited on p62, provided that the standard u-plot given in those references is interpreted as the observer's predictive c.d.f. $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$ of U_n. Thus in the case of inter-failure time prediction, the PFS $\mathcal{S}^*$ has been defined by setting the predictive c.d.f.s $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$ equal to the sample c.d.f. of all the u_i values observed up to current time (or a smoothed version of this). For the case of an arbitrary raw PFS we generalise this technique by setting $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$ equal to the modified u-plot function, S_{n-1}, defined by (29) and (30). Again here it is assumed understood that references to the "observed u^{n-1}" (as well as conditioning of the form "$\cdot|u^{n-1}$" in probability statements and predictive c.d.f.s) must be regarded as a shorthand for "observation that u^{n-1} lies within the set of possible values consistent with the observed value of x^{n-1}".

A small point about definition (34) is that there are some special circumstances in which—to be logically consistent—we must allow the recalibrated predictive distribution $F_n^{*X}(x_n|x^{n-1})$ to be 'improper'. E.g. positive "$\mathcal{S}^*$-predictive probability" concentrated at U_n=1 can cause $F_n^{*X}(x_n|x^{n-1})$ to assign positive probability to the prediction X_n=∞, even when the corresponding raw distribution $F_n^X(x_n|x^{n-1})$ is not improper in this sense. In other respects we *do* have good c.d.f.-behaviour of $F_n^{*X}(\cdot|x^{n-1})$ (monotonic non-decreasing, and right-continuous) inherited from similar behaviour of *both* $F_n^X(\cdot|x^{n-1})$ and $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$. However, when we consider the effect of definition (34) on *left* hand limit values, we find that the situation becomes complicated due to the lack of left-continuity for general c.d.f. functions. In particular, to take one interesting case which crops up in the numerical examples of chapter 4, when we consider the extreme, large values of X_n and U_n, we find that we can, for the recalibrated predictor, rely only on

$$\lim_{x_n\to\infty} F_n^{*X}(x_n|x^{n-1}) = \begin{cases} G_n^{\mathcal{S}^*}(s|u^{n-1}), & \text{if } F_n^X(x_n|x^{n-1}) \text{ attains } s \text{ for some finite } x_n; \\ G_n^{\mathcal{S}^*}(s-|u^{n-1}), & \text{otherwise,} \end{cases}$$

where $s \stackrel{\text{def}}{=} \lim_{x_n\to\infty} F_n^X(x_n|x^{n-1})$. So, even for a 'proper' raw predictor (a predictor with s=1), the presence of a discontinuity $G_n^{\mathcal{S}^*}(1-|u^{n-1}) < G_n^{\mathcal{S}^*}(1|u^{n-1})$ corresponding to concentrated predictive probability of the PFS $\mathcal{S}^*$ at the value U_n=1 will produce an *improper recalibrated* predictor whenever the *proper raw* distribution $F_n^X(x_n|x^{n-1})$ represents an unbounded (above) RV. That is to say, *a proper, unbounded raw predictive distribution becomes improper upon recalibration whenever the PFS $\mathcal{S}^*$ assigns positive predictive probability to the extreme event U_n=1 corresponding to X_n=∞.* In

this case, the logically consistent interpretation of (34) is simply that the recalibrated predictor $F_n^{*X}(\cdot|x^{n-1})$ predicts a value of ∞ for X_n with a positive probability equal to that with which $G_n^{\mathcal{S}^*}(U_n|u^{n-1})$ predicts the extreme event U_n=1. There are other related special cases, such as that a zero $\mathcal{S}^*$-predictive probability on a neighbourhood of U_n=1 can result in a proper (or bounded) recalibrated predictor where the original raw predictor lacked the same property of being proper (or bounded). Another case, perhaps more difficult to interpret in the application to failure count or inter-failure time prediction, would be the event that $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$ might concentrate predictive probability at the small end extreme U_n=0. Logically, according to (34), this appears to demand the interpretation that the recalibrated predictor is assigning positive probability to a value for X_n which is off the small end of the scale of values predicted by the raw PFS—perhaps in the form of a prediction that $X_n = -\infty$ with positive probability. Depending on how the recalibrated probabilities embodied in the PFS $\mathcal{S}^*$ are obtained, various of these awkward cases can be argued to be improbable or impossible to occur. In particular, the gradient constraints described in §3.6.6 can be used to eliminate the possibility of $\mathcal{S}^*$-probability concentrated at point values. These considerations seem similar to those concerning recalibration of small probabilities discussed below in §3.6.5.

3.6.3 Weighting the U-Plot Sum when it is Used for Recalibration

One alternative to the use of equal weights in (29) is for the observer to represent, by the assignment of *different* weights w_i, any intuitive belief they may hold that certain of the previous U_i should be modelled under $\mathcal{S}^*$ as being more strongly associated with U_n than others. Thus it is possible to arrive at a predictive distribution of U_n under an alternative $\mathcal{S}^*$ by taking S_{n-1}, from (29), as the predictive c.d.f. of U_n, but now with the w_i varying in some systematic way to reflect the intuition of the observer. In obtaining the results of §4.2, weights were used which decrease exponentially the less recent the prediction $F_i^X(\cdot|x^{i-1})$ from which u_i (or rather its posterior c.d.f. $G_i(\cdot|x^i)$, given by (30)) was obtained, so that

$$G_n^{\mathcal{S}^*}(u|u^{n-1}) = \sum_{i=n_0}^{n-1} w_i G_i(u|x^i), \qquad \text{where } w_i = \frac{r^{n-1-i}(1-r)}{1-r^{n-n_0}}, \quad 0<r<1. \tag{35}$$

Thus a decision can be taken to be influenced more heavily by the most up-to-date observation of how the realisation $\langle u_n \rangle$ of the process $\langle U_n \rangle$ is behaving. Figure 6a on p67 provides some empirical evidence to support this approach. In this figure the functions

$$S_n(u) = \sum_{i=n-9}^{n} \tfrac{1}{10} G_i(u|x^i), \qquad n = 15, 25, 35, 45, 55$$

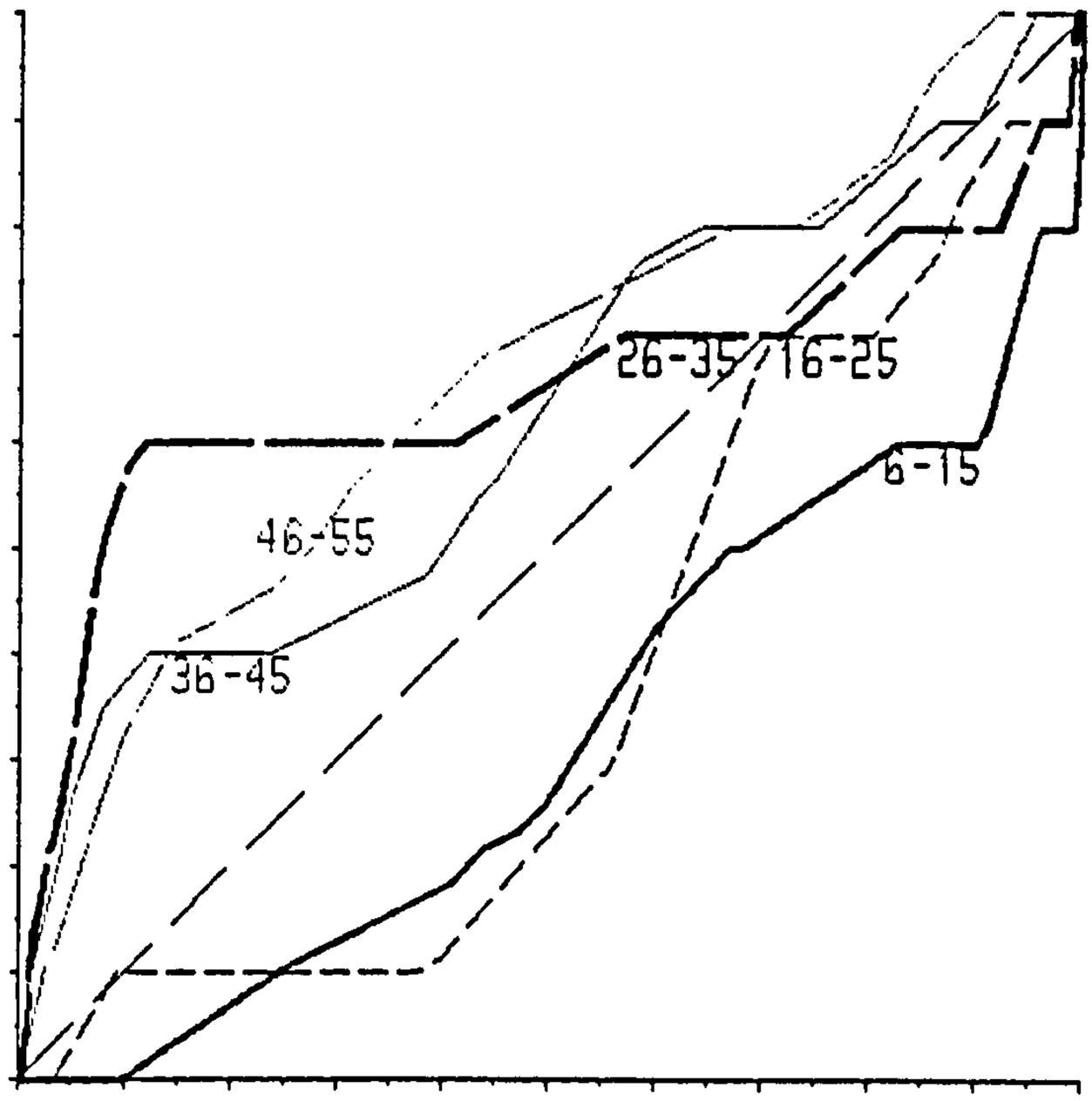


Figure 6a: Evidence of trend in $\langle U_n \rangle$ for DJMAG PFS applied to SS3 data

are shown for the DJMAG PFS (see p85) applied to the SS3 data set (see p84 and Table 2 on p86). Interpreted as empirical approximations to the marginal distributions of the U_i values taken from each of these 5 disjoint i-sequences, the corresponding 5 plotted functions seem at least consistent with some kind of systematic trend over time in the distribution of the $\langle U_n \rangle$. Hence, we can informally regard Figure 6a as evidence which suggests an explanation for the effectiveness—as applied to this particular *combination* of *data* and *PFS* (see Chapter 4)—of recalibrators based on the exponential sequence of weights given in equation (35).

Of course, any formal test of such a "trend" hypothesis for the $\langle U_n \rangle$ would need to take account of the "censoring" involved here: The $\langle U_n \rangle$ of §3.4.4, adopted throughout §3.5, were defined in such a way (27) as to be incompletely determined by observation of the $\langle x_n \rangle$. The indeterminacy of our randomized U_n-sequence, as described in §§3.4.4–3.5.1, creates complications in the task of statistically formalizing a test for trend. These complications are related to the similar difficulties mentioned on p59 of formally testing the significance of deviations of the Modified U-Plot from the ideal uniformity of U_n-distribution corresponding to the 45°-line. We briefly, in the following section, investigate the utility, under such difficulties, of a 'y-plot' as a test for trend, illustrating the discussion in terms of testing for the trend that, on visual inspection at least, appears to be present in the $\langle u_n \rangle$ data as represented in Figure 6a.

3.6.4 'Y-plot' Tests for Trend in the Case of Discrete[25] Predictions

A commonly used [59, 54, 10], [7, p28], [65, p140] statistical test for systematic trend in a *standard* $\langle u_n \rangle$ sequence (i.e. a fully observable $\langle u_n \rangle$ sequence emanating from a *continuous* PFS applied to, for example, a sequence of one-step-ahead inter-failure time predictions) is based on a transformation of the vector of observed $\langle u_n \rangle$ values whose sample c.d.f. is plotted as a 'y-plot'. Let us assume there are K of these observed u_n and index them $u_{n_0}, u_{n_0+1}, \ldots u_{n_0+K-1}$ (remembering that prediction begins at observation sequence index $n_0 > 1$). Then the y-plot is formed of the transformed sequence

$$y_n = \frac{\sum_{i=n_0}^{n} \log(1-u_i)}{\sum_{i=n_0}^{n_0+K-1} \log(1-u_i)} \qquad n = n_0, \ldots, n_0+K-2 \tag{36}$$

Here, each y_n is a function of the entire sequence $\langle u_i \rangle_{i=n_0}^{n_0+K-1}$ of observed U s; so notice that, unlike the u_n residual whose value depends only on 'the past'[26], the value of the 'transformed

[25] The reasoning used here applies equally to Mixed Predictions

[26] on x_i for $i \leq n$

residual' y_n is dependent, via the denominator of (36), on the *number* of subsequent predictions that are to be made before the statistical test for trend will be applied (predictions of those X_i with $n < i \leq n_0+K-1$), as well as on the eventual *realised values* of these later-predicted elements of the $\langle X_n \rangle$ sequence. Under the usual null hypothesis that the $\langle U_i \rangle_{i=n_0}^{n_0+K-1}$ are independently distributed from an identical uniform probability distribution (corresponding to a PFS of the $\langle X_n \rangle$ which is 'perfect' or 'true', *and* which nowhere concentrates predictive probability mass at a point value—see §3.3 on p49), the $\langle y_i \rangle_{i=n_0}^{n_0+K-2}$ become distributed as the order statistics of a random $\mathcal{U}[0,1]$ sample of size one less than the size K of the $\langle u_i \rangle_{i=n_0}^{n_0+K-1}$ sample[27]. Significant deviation of this $\langle y_i \rangle_{i=n_0}^{n_0+K-2}$ sample from such uniformity, as measured by the Kolmogorov-Smirnov distance (sometimes shortened to 'KS-distance', or 'Kolmogorov-distance') of the sample c.d.f. of these $\langle y_i \rangle$, has been used elsewhere to test for systematic trend in the sequence $\langle u_i \rangle_{i=n_0}^{n_0+K-1}$. (Littlewood and Keiller [59] point out that any such trend might be interpreted in terms of a failure of the PFS to capture correctly some trend in the reliability data—in their case an inter-failure time sequence—to which the PFS is being applied.)

The immediate difficulty with transporting this y-plot technique into the generalised context of our Modified U-Plot is that we have resorted to a *randomized* u-residual (27) each time that predictive probability has been concentrated at a point value x, immediately prior to this value being realised, $X_n{=}x$. Then the corresponding u_n in (27) is incompletely observed. So long as there is *at least one* such randomized u_n, for which we will finally have only a *posterior distribution*—not an exact *numerical value*, then *all* of our $\langle y_i \rangle_{i=n_0}^{n_0+K-2}$ become RVs $\langle Y_i \rangle_{i=n_0}^{n_0+K-2}$ for which we, likewise, have only posterior distributions with which to work. (In fact, at the end of our sequence of observations, we have a joint, posterior distribution $\langle Y_i \rangle_{i=n_0}^{n_0+K-2} \mid \langle x_i \rangle_{i=n_0}^{n_0+K-1}$ for the Y-vector, whose components will be statistically associated.) Recall from §3.5 that, in this generalised discrete- or mixed- prediction case, under a hypothesis of 'perfection' of our PFS (notated as $\mathcal{P} = \mathcal{Q}$ in §§3.2–3.5), and prior to obtaining any observation evidence of the form $\langle X_n \rangle = \langle x_n \rangle$, our randomized U_n are distributed i.i.d. uniformly $\mathcal{U}[0,1]$. Under the same 'PFS perfection' hypothesis, but now conditionally *at the end* of the prediction/observation process, given observations $\langle x_n \rangle_{i=n_0}^{n_0+K-1}$, the updated distribution of our randomized U_n says that they are now conditionally independently, *non*-identically, uniformly distributed

$$U_n \mid \langle X_i \rangle_{i=n_0}^{n_0+K-1} \sim \mathcal{U}[p_{nk}, q_{nk}] \tag{37}$$

[27] Briefly, under the null hypothesis, a $u \mapsto -\log(1-u)$ transformation creates i.i.d. exponentially distributed RVs. Thinking of these as the spacings of successive points of an HPP process in time, we may consider the time interval from time zero, up to the time of the K^{th} of these points, and condition on its length. It is a well known property [14, p27] of the HPP process that the positions of the intermediate $K-1$ points within this interval are then conditionally distributed as order statistics of a $(K-1)$-sized random sample from a uniform distribution.

where k is determined by the value of the observed x_n and may below be dropped from the notation without ambiguity. See Figure 3(a) on p51. For the data illustrated in Figure 6a, we have tabulated these 'observed u-intervals' $[p_n, q_n]$ as the central columns in Table 1 on p72. Clearly the development of a 'y-plot-based' test for trend in the randomized $\langle u_n \rangle$ would require that some thought should be applied to the consequences, of this posterior U_n-distribution (37), for the induced posterior distribution of the $\langle Y_n \rangle$ vector, defined by (36), under the same null hypothesis of a perfect PFS. Firstly, we can see easily from (36) that y_n is a monotonically increasing function of each u_i with $n_0 \le i \le n$, and a monotonically decreasing function of each u_i with $n < i \le n_0+K-1$, yielding the *bounds* of the posterior marginal distribution of $Y_n \mid \langle x_i \rangle_{i=n_0}^{n_0+K-1}$:

$$y_n^{\min} = \frac{\sum_{i=n_0}^{n} \log(1-p_i)}{\sum_{i=n_0}^{n} \log(1-p_i) + \sum_{i=n+1}^{n_0+K-1} \log(1-q_i)} \tag{38}$$

$$y_n^{\max} = \frac{\sum_{i=n_0}^{n} \log(1-q_i)}{\sum_{i=n_0}^{n} \log(1-q_i) + \sum_{i=n+1}^{n_0+K-1} \log(1-p_i)} \tag{39}$$

With a little more effort[28], the mean value of this same posterior Y_n-distribution

$$\mathrm{E}\left[Y_n \;\middle|\; \langle x_i \rangle_{i=n_0}^{n_0+K-1} \right] = \frac{1}{\text{volume}(C)} \int_C \left[\sum_{i=n_0}^{n} \log(1-u_i) \Big/ \sum_{i=n_0}^{n_0+K-1} \log(1-u_i) \right]$$

may be obtained by numerical integration, where C is the K-dimensional hyper-cube defined by $p_i \le u_i \le q_i$, $i = n_0, \ldots, n_0+K-1$. The numerical values for these three $y_n^{\min}$, $\mathrm{E}[Y_n \,|\, \text{data}]$, $y_n^{\max}$, obtained using the same data set that was used to produce Figure 6a, are tabulated as the righthand columns of Table 1 on p72. These three columns are plotted in Figure 6b in the form of three corresponding sample c.d.f.s (but note the cosmetic adjustment mentioned in footnote 30). That is, for each value of the *ordinate* P, if we draw a horizontal line across the figure at that P-value, it will intersect the three graphs shown at the minimum, mean, and maximum, respectively, of the posterior distribution of the *point at which the y-plot* (i.e. the sample c.d.f.—were the exact values comprising the sample $\langle Y_n \rangle_{n=6}^{54}$ known) corresponding to our unobservable, randomized $\langle U_n \rangle_{n=6}^{55}$ would intersect the same horizontal line. As in the case of the Modified U-Plot, which we earlier defined similarly as the posterior mean of an unobservable plot[29] associated with randomised residuals, it is not obvious

[28]The dimension of the integral may be high, but the integrand is a well-behaved, bounded, smooth function, monotonic in each argument u_i.

[29]similarly, but not quite equivalently: Even if we restrict to equal weights w_i in the definition (29), (30) on p57 of the Modified U-Plot, still we used there (see also (32) on p59) the *posterior expectation of the sample c.d.f.* of an 'incompletely observable sample'; whereas the middle plot in Figure 6b corresponds rather to the *sample c.d.f.* of the *posterior expected order statistics* of an 'incompletely observable sample'.

how to interpret the middle plot of Figure 6b in a manner enabling a formal statistical test for trend to be obtained. As for the U-plot case, we provide no answer, in this thesis, to the general problem of formally interpreting our "E[Y]-plot" in the discrete- (or mixed-) data cases. Fortunately, however, in the case of *this particular* data set, the deviation from the 45°-line is sufficiently pronounced that even the extremes (38), (39)—between[30]which the unobservable c.d.f. of the true Y_n-sequence must of course lie—can be used to *formally* reject a hypothesis of uniformity for the $\langle Y_n \rangle$ with a high degree of confidence. The Kolmogorov-Smirnov distance (KS-distance, or KS-statistic) [41, §30.49, p477], [80, 75] is one test statistic commonly used in this context. In the case of the data illustrated in Figure 6b, we know that the KS-statistic (of the unobservable sample c.d.f. of the randomized $\langle Y_n \rangle_{n=6}^{54}$ values) must be at least .2283. We are assured of this because the most 'vertically distant' point on our "$y_n^{\min}$ c.d.f." from the 45°-line may be easily verified from the numbers in Table 1 to occur immediately before the step positioned at the value $y_{28}^{\min}$ on the horizontal axis. This is the 23$^{\text{rd}}$ $y^{\min}$-value (because we began prediction at $n = n_0 = 6$), i.e. it is the step *up to* level $\frac{23}{49}$ on the P-axis (the vertical axis of the graph). So, without reference to the posterior y-plot distribution *shape* or to the computed values of the posterior y-means, but knowing only the *extremes* $y^{\min}$ and $y^{\max}$ of this distribution, our lower bound on the KS-distance is the distance between the points $(y_{28}^{\min}, \frac{22}{49})$ and $(y_{28}^{\min}, y_{28}^{\min})$ on Figure 6b. Coincidentally, this lower bound of $0.6773 - \frac{22}{49} = 0.2283$ is *equal* (to 4 significant digits) to the 1%-critical value of the Kolmogorov-Smirnov test[31] for a sample of this size (y_n, $n = 6, \ldots, 54$, so the sample size is 49 [80, 75]). We can conclude, in the case of this data shown in Table 1 and Figure 6a, that there is a formally confirmed deviation of our 'randomised y-plot' below the 45°-line, which is *at least as significant* as the 1%-level. The $\langle Y_n \rangle$ are therefore statistically 'too large' to be accepted at the 1%-level as a sample from a parent distribution uniform on the interval [0,1]. This in turn means that the randomised $\langle U_n \rangle$ show a *significantly decreasing trend*, as Figure 6a earlier seemed to suggest that they might, so that the DJMAG predictions obtained with this discrete SS3 data set are tending towards *increasing pessimism* (smaller u-values) as time progresses.

---

[30]We found that, if one attempts to plot at this resolution exactly the steps of a step function (and at the same time distinguish the three plots by means of dashed line styles), then the graphing package produces an unsatisfactory line. So Figure 6b was plotted by joining with straight line segments only the *front* (i.e. upper, left) corners of each 'step' (of the sample c.d.f. functions, for each of the three quantities that are tabulated in the 'Y-values' column of Table 1). I.e. there is a small discrepancy (maximum vertical size $\frac{1}{49}$: so just visible at the resolution of this figure) between the plots shown and the true region known to contain the sample c.d.f. graph of the unobservable $(y_n)_{n=6}^{54}$ sequence. This pictorial discrepancy is present *only* on the plots of Figure 6b, and does not affect either the numerical values in Table 1, or the calculation of the lower bound on KS-distance, both of which we obtained exactly (to the number of digits shown in Table 1), using the exact, step-function sample c.d.f.s., as required by the rigorous KS test procedure.

[31]two-sided, single-sample

There have been several other tests proposed for identifying trend in the spacings of a point-process. For example, the 'Laplace test' [7, 65], or the various tests mentioned in [14, Chapter 3]. We will not further pursue the notion of formal U_n-trend-tests for our randomized $\langle U_n \rangle$ in this thesis, contenting ourselves with the initial thoughts and example given above about extending the 'y-plot' method, and with our proposal that the possibility of systematic trend in the $\langle U_n \rangle$ provides one potential justification for considering the use of unequal weights w_i in a Modified U-Plot (29) on p57 when it is to be used for recalibrating a PFS.

Failure Counts		U-values		Y-values		
Index	Count					
n	m_n	p_n	q_n	$y_n^{\min}$	$\mathrm{E}[Y_n]$	$y_n^{\max}$
6	4	0.4070	0.6019	0.007289	0.0116	0.01744
7	2	0.0954	0.2457	0.008709	0.0146	0.02270
8	10	0.9966	0.9990	0.08971	0.1152	0.1489
9	3	0.2432	0.4402	0.09407	0.1221	0.1587
10	4	0.4743	0.6686	0.1039	0.1361	0.1775
11	5	0.6817	0.8260	0.1214	0.1592	0.2070
12	7	0.9120	0.9615	0.1586	0.2047	0.2617
13	10	0.9917	0.9971	0.2325	0.2900	0.3586
14	5	0.4965	0.6702	0.2442	0.3045	0.3752
15	8	0.9021	0.9533	0.2822	0.3477	0.4227
16	8	0.8645	0.9306	0.3157	0.3852	0.4630
17	5	0.3774	0.5505	0.3246	0.3955	0.4739
18	5	0.4002	0.5747	0.3343	0.4064	0.4854
19	14	0.9991	0.9997	0.4508	0.5286	0.6099
20	7	0.5525	0.6966	0.4663	0.5447	0.6255
21	9	0.8022	0.8850	0.4963	0.5752	0.6549
22	3	0.0339	0.0917	0.4973	0.5763	0.6558
23	7	0.5776	0.7190	0.5145	0.5934	0.6719
24	7	0.5649	0.7077	0.5313	0.6101	0.6872
25	7	0.5536	0.6976	0.5478	0.6262	0.7019
26	3	0.0466	0.1195	0.5493	0.6276	0.7030
27	10	0.9211	0.9604	0.5975	0.6741	0.7459
28	*13*	*0.9860*	*0.9939*	*0.6773*	*0.7496*	*0.8150*
29	3	0.0296	0.0818	0.6784	0.7506	0.8156
30	3	0.0379	0.1005	0.6798	0.7517	0.8164
31	6	0.4118	0.5718	0.6924	0.7628	0.8254
32	8	0.7210	0.8298	0.7200	0.7874	0.8462
33	1	0.0018	0.0130	0.7201	0.7875	0.8462
34	2	0.0211	0.0730	0.7212	0.7883	0.8467
35	1	0.0045	0.0288	0.7216	0.7886	0.8468
36	2	0.0396	0.1226	0.7234	0.7900	0.8476
37	1	0.0084	0.0487	0.7240	0.7904	0.8478
38	0	0.0000	0.0111	0.7242	0.7905	0.8479
39	6	0.7523	0.8669	0.7560	0.8179	0.8702
40	4	0.3832	0.5769	0.7694	0.8286	0.8782
41	4	0.3916	0.5858	0.7832	0.8397	0.8862
42	1	0.0153	0.0792	0.7844	0.8405	0.8866
43	3	0.2387	0.4342	0.7933	0.8472	0.8911
44	4	0.4488	0.6441	0.8101	0.8603	0.9005
45	7	0.8997	0.9547	0.8630	0.9035	0.9350
46	3	0.2377	0.4329	0.8726	0.9102	0.9392
47	1	0.0196	0.0967	0.8742	0.9112	0.9396
48	4	0.4764	0.6707	0.8935	0.9252	0.9493
49	3	0.2738	0.4797	0.9048	0.9330	0.9541
50	0	0.0000	0.0244	0.9052	0.9332	0.9541
51	2	0.1302	0.3108	0.9117	0.9373	0.9562
52	5	0.7296	0.8605	0.9475	0.9635	0.9750
53	1	0.0300	0.1353	0.9501	0.9649	0.9755
54	1	0.0332	0.1463	0.9529	0.9664	0.9760
55	5	0.8191	0.9184			

Table 1: Investigation of Y-Plot as Measure of Trend in the U-Data of Figure 6a

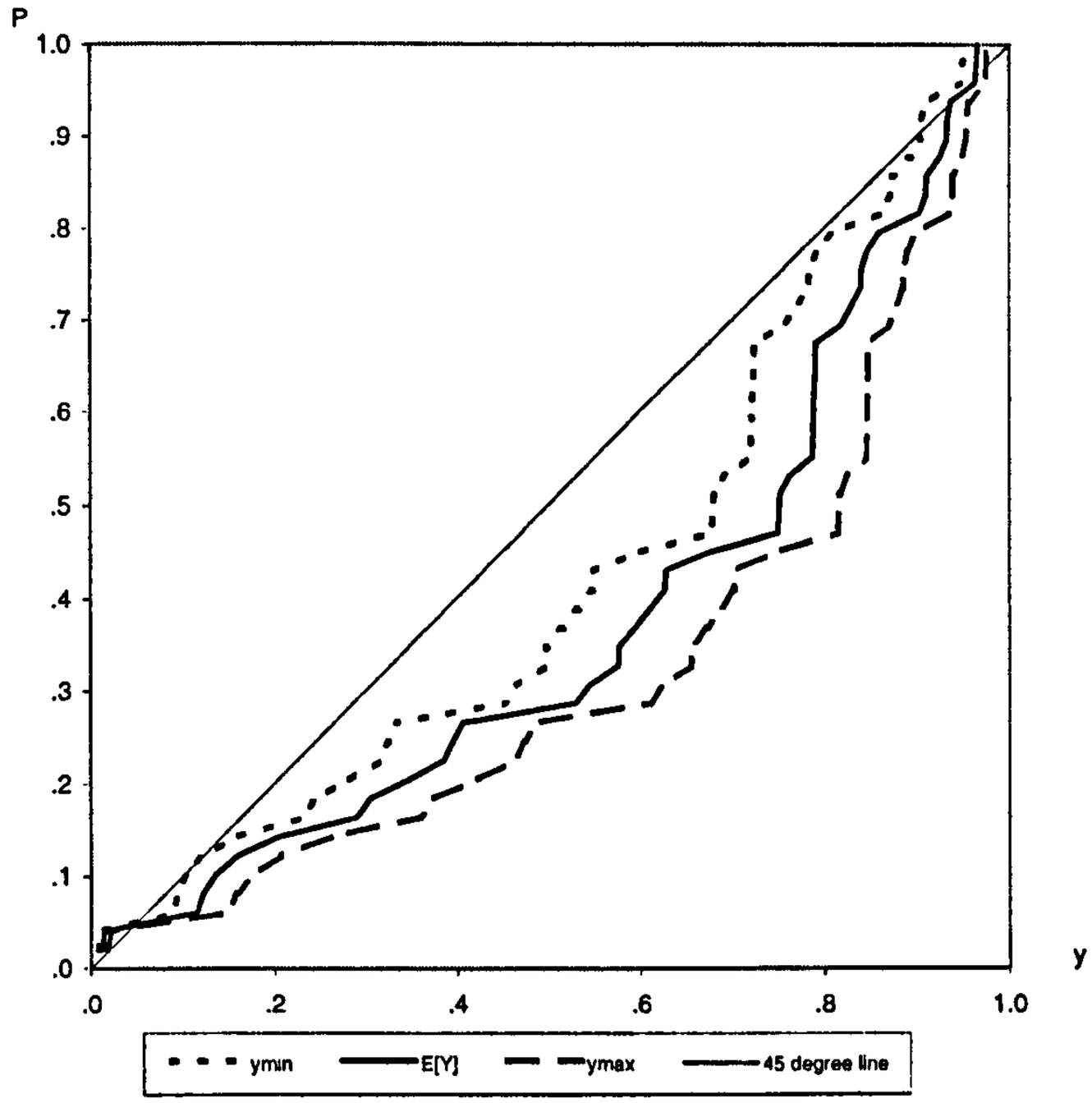


Figure 6b: Distribution of 'y-plot' for DJMAG PFS applied to SS3 data

3.6.5 A Difficulty of Recalibrating Very Small Probabilities

It perhaps seems unreasonable to assume that the predictive c.d.f., $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$, should possess discontinuities, or discontinuities in derivative, at the points, mentioned in §3.5, where S_{n-1} does. For this reason we have applied the same smoother, based on the use of B-splines and used in [11], to S_{n-1} before using it as $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$ in order to recalibrate. The results obtained are presented in §4.2.

A modification of this smoother has been implemented, in an attempt to overcome the following weakness in the recalibrator as so far described. A large random variability in the recalibrated predictive probability of *unlikely* events results from the above method of defining $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$. This may be especially noticeable in the case of discrete predictive c.d.f.s. The importance or otherwise of this variability depends in part on whether it is considered desirable to predict unlikely events as accurately as likely ones, with probabilities which are *proportionately* not too far in error. If this is so, then the random variability can be partially corrected by the smoothing procedure as it stands, but further improvement has been obtained (as measured by discrete prequential likelihood). If a "small" (in probability terms) subset, A say, of $[0,1]$ is considered and if the number $n-1$ of observed u_i is not large, then there is a high probability that the proportion of the u_i in u^{n-1} which

fall in A is not at all representative of the 'true probability' of such an event occurring, nor, more importantly, of the observer's subjective "recalibrated" probability (i.e. under his alternative model $\mathcal{S}^*$) that the *next* U_n is about to fall in A. This is simply because, in percentage-error terms, a reliable estimate of the probability of an unlikely event is not given by the proportion of occurrences in a small number of trials[32]. For example, in the context of discrete predictions it is frequently the case that the recalibrated predictive probability of one discrete outcome for X_n, given x^{n-1}, corresponds to the predictive probability, based on $\mathcal{S}^*$, of U_n lying in such a small subset of $[0,1]$. (This can be verified from the u-plots in Chapter 4.) To take a hypothetical example in which X_n takes non-negative integer values only, suppose A is the interval $[0,.01]$ and the event $X_{16}=0$ or 1 given x^{15} corresponds to U_{16} lying in A (i.e. suppose $F_{16}^{X}(1|x^{15})=.01$). Then the current recalibration procedure would assign a recalibrated predictive probability of

$$F_{16}^{*X}(1|x^{15}) = \begin{cases} 0, & \text{if } u_i > .01 \text{ is observed for } n_0 \le i \le 15 \\ w_i \text{ at least}, & \text{if } u_i \le .01 \text{ is observed for some } n_0 \le i \le 15 \end{cases}$$

either of which events (and especially the first) has a significant probability even if the raw PFS is perfect, (in which case $F_{16}^{X}(1|x^{15})=.01$ would be the "perfect" prediction.) This example illustrates why $\mathbf{P}_{\mathcal{S}^*}\left[U_n \in A|u^{n-1}\right]$ cannot safely be obtained empirically from S_{n-1} when $\mathbf{P}_{\mathcal{S}^*}\left[U_n \in A|u^{n-1}\right]$ is small and n is not large. Attempting to do this results in large random fluctuations in the *proportionate* adjustment from $\mathbf{P}_{\mathcal{S}^P}\left[U_n \in A|u^{n-1}\right]$ to $\mathbf{P}_{\mathcal{S}^*}\left[U_n \in A|u^{n-1}\right]$ during the recalibration step.

3.6.6 A Solution using a Gradient-Constrained U-Plot for Recalibration

To get round this problem the smoothed $G_n^{\mathcal{S}^*}(\cdot|u^{n-1})$ can be restricted to belong to a family with upper and lower bounds on first derivative. Another way of saying the same thing is that bounds on the ratios of recalibrated and unrecalibrated predictive probabilities of all events of the $\langle U_n \rangle$ process, one-step-ahead, can be imposed

$$\epsilon \underset{\mathcal{S}^P}{\mathbf{P}}\left[U_n \in A|u^{n-1}\right] \le \underset{\mathcal{S}^*}{\mathbf{P}}\left[U_n \in A|u^{n-1}\right] \le \frac{1}{\delta} \underset{\mathcal{S}^P}{\mathbf{P}}\left[U_n \in A|u^{n-1}\right], \tag{40}$$

say, where $0 < \epsilon, \delta < 1$[33]. Although the introduction of the bounds ϵ and δ has been justified by considering unlikely events, $U_n \in A$, where A is small, it follows (in the general case by considering the Radon-Nikodym derivative for the two measures $A \mapsto \mathbf{P}_{\mathcal{S}^*}\left[U_n \in A|u^{n-1}\right]$ and

[32] The coefficient of variation [44] of the proportion of successes in n Bernoulli trials with parameter p is $\sqrt{\frac{1-p}{np}} \sim (np)^{-\frac{1}{2}}$ as $p \to 0$.

[33] One further extension which has not been implemented in obtaining the results of Chapter 4, would be to allow ϵ and δ to decrease as time progresses since, as n increases, the larger sample u^{n-1} will allow $\mathcal{S}^*$, defined without the last refinement, to produce more accurate predictive probabilities for U_n lying in smaller sets A.

$A \mapsto \mathbf{P}_{\mathcal{S}^{\mathcal{P}}}[U_n \in A | u^{n-1}]$) that this necessarily entails imposing the same bounds on the ratio of these two predictive probabilities for all events $A \subseteq [0,1]$. It will be noticed that, for a discrete PFS, the bounds ϵ and δ can be directly interpreted in terms of the logPLR[34] plot of recalibrated versus unrecalibrated predictors. The logPLR plot (see for example Figure 14b) will have successive ordinates, y_n say, which must satisfy

$$y_{n-1} + \log \epsilon \leq y_n \leq y_{n-1} - \log \delta.$$

The details on the method of imposing these bounds on the slope of the smoothed function $G_n^{\mathcal{S}^*}(\cdot | u^{n-1})$ used for recalibration are given in Appendix B. This method and the use of unequal weights w_i, are the two enhancements mentioned on p62 at the beginning of this section, and can be introduced into the recalibration procedure either separately or in combination.

3.7 Implications for Recalibration of Other Predictions

This chapter concludes with a review of the ideas which have been used to extend the method of recalibration to arbitrary one-step-ahead scalar predictive distributions. By choosing to view the recalibration method in terms of prediction of the outcomes of sequences of *events*, rather than the values of sequences of random variables, it is suggested that recalibration might be further extended, including as an example an application to the direct recalibration of failure rate predictions.

3.7.1 Recalibrating a Raw Predictor of a Sequence of *Analogously Predicted, Equi-probable* Events

Suppose we concentrate now on a sequence of individual events $\langle E_n \rangle$ i.e. we have a discrete time metric n, which may if we like be derived in some way from an underlying continuous time model $\langle \Omega, \Sigma, \mathbf{P} \rangle$, and we know that at discrete-time $n = 1$ we will have observed whether of not E_1 occurred, etc. Thus we assume that $E_n \in \mathcal{G}_n$ where $\mathcal{G}_n$ denotes *observation up to 'time' n* (so that $\mathcal{G}_{n-1}$ is a sub-sigma algebra of $\mathcal{G}_n$ for each n). Then an interpretation of the phrase "well-calibrated" for a sequence of one-step-ahead probabilistic predictions (assumed derived from a particular PFS $\mathcal{P}$) as to the occurrence or otherwise of each of these events is usually thought of [19] as something like

$$\frac{1}{n}\sum_{i=1}^{n} I_{E_i} \quad \text{is close to} \quad \frac{1}{n}\sum_{i=1}^{n} \mathop{\mathbf{P}}_{\mathcal{P}}[E_i | \mathcal{G}_{i-1}], \tag{41}$$

[34] This is the analogue for two competing PFSs of the likelihood ratio for two competing parameter values from a parametric family of models. See (48) on p83 for a definition.

or perhaps even

$$\frac{1}{n}\sum_{i=1}^{n} I_{E_i} - \frac{1}{n}\sum_{i=1}^{n} \underset{\mathcal{P}}{\mathbf{P}}[E_i|\mathcal{G}_{i-1}] \to 0 \qquad \text{as } n \to \infty, \tag{42}$$

in some form of stochastic convergence[35]. Here, as on p50, we use indicator function notation, so that I_{E_i} is the random variable whose value is

$$I_{E_i} = \begin{cases} 1, & \text{if event } E_i \text{ occurs;} \\ 0, & \text{otherwise.} \end{cases}$$

In practice, of course, there is usually only one unique realisation of the process available, and, for that data sequence, well-calibratedness is simply an empirical property. Then the concept of stochastic convergence of (42) is not applicable. Although, such theoretical considerations might still be useful in providing arguments to say how close is 'close' in (41); or, when comparing two PFSs on the same event-sequence, to say whether one has 'significantly' outperformed the other in the sense of (41).

Given a sequence $\langle E_n \rangle$ of *events* which have each been *analogously* predicted with *equal probability* p, this kind of definition of "well-calibrated" has been used to form the basis of a definition of *re*calibration for current predictive probabilities of as yet unobserved events. By "analogously" predicted, we mean ideally that[36]

- the events E_n of the sequence are themselves in some sense analogous to each other,
- the states of knowledge $\mathcal{G}_{n-1}$ or observation on the basis of which each event has been predicted are in some sense analogous relative to the event being predicted for each event in the sequence,
- the conceptual and probabilistic model, inference procedure, and resulting forecasting system $\mathcal{P}$ (which may have been composed into a single automated algorithm) are likewise similar or identical for the prediction of all events of the sequence.

When these conditions hold, it is intuitively appealing, and forms the basis of the methods discussed in [2, 6, 10, 58, 19] that the definition (41) of well-calibratedness should be extended, for such sequences of equi-probable event predictions, and provided n is sufficiently large, to an equation which *assigns* the left hand side of (41)

$$\underset{\mathcal{P}^*}{\mathbf{P}}[E_{n+1}|\mathcal{G}_n] = \frac{1}{n}\sum_{i=1}^{n} I_{E_i} \tag{43}$$

as a recalibrated predictive probability of the next event E_{n+1} in the sequence.

---

[35] In [19] it is shown that a PFS which may be identified with the *true* conditional probabilities of a stochastic process measure $\mathcal{P}$ (see §2.2) is well-calibrated in this second sense using "$\mathcal{P}$-almost sure" convergence of the limit.

[36] or, at least, that the triples $(E_n, \text{data}_{n-1}, \text{prediction-method}_n)$ are believed to be evolving systematically with n

3.7.2 Recalibration of a Predictor of *Random Variables* Interpreted as Recalibration of Predictors of some Associated *Event*-Sequences

This notion of the property of being *well*-calibrated has been used in [2, 6, 10, 58] as the basis of a procedure for *re*-calibration of a probabilistic predictor, as discussed in §3.2. The work contained in these references in fact describes the recalibration of a sequence of predictive distributions $\langle F_n^X(\cdot|X_{n-1})\rangle$ of *continuous* scalar *random variables*, $\langle X_n\rangle$—rather than a sequence of predictive event-probabilities—within an iterative *predict→observe→predict...* set-up which is in other respects identical. The methods used in the references to define a property of well-calibratedness and also to recalibrate predictions of such a continuous quantity can easily be expressed in terms of analogous methods for the less complex *event* sequences introduced in this section. It is only necessary to focus on a few particular event sequences from amongst all such sequences which could be defined along the lines of $E_n \stackrel{\text{def}}{=} \{\omega : X_n(\omega) \in A_n\}$ for some set A_n (where the set[37] A_n is allowed to be defined in terms of observations $\mathcal{G}_{n-1}$). The method used is effectively to construct for all fixed pairs of numbers $p_0 \in [0,1]$ and $p \in [0, 1-p_0]$, a sequence $\langle E_n(p_0,p)\rangle$ of events *defined in terms of the pair of percentiles* $100p_0\%$ and $100(p_0+p)\%$ of the *raw predictive distribution* of X_n. That is, to define the event $E_n(p_0,p) = \{\omega : a_n < X_n \leq b_n\}$, where $F_n^X(a_n|x^{n-1}) = p_0$ and $F_n^X(b_n|x^{n-1}) = p_0 + p$; $n = 1, 2, \ldots$. Then a recalibrated prediction of the event E_{n+1} assigns, by (43), the left hand side of (41) to be $\mathbf{P}_{\mathcal{P}^\bullet}[E_{n+1}(p_0,p)|x^n]$, the recalibrated probability of E_{n+1} given that $X^n = x^n$. A recalibrated predictive *c.d.f.* for the *random variable* X_{n+1} is then the c.d.f. which assigns this recalibrated probability to events $E_{n+1}(p_0,p)$ for all p_0 and p. Note that it is only sequences of events defined in a this way, in terms of a pair of percentiles which are *fixed for all* n, that are correctly recalibrated according to (43) by this method. Other sequences of equi-probable (i.e. equi-probable in the view of the evolving predictions of the raw PFS) events must inevitably be assigned recalibrated probabilities which *do not satisfy* (43)[38]. But note also that, fortunately, for a raw predictor of a *continuous scalar* process $\langle X_n\rangle$, it is trivial to show that, for each n, all of the recalibrated probabilities produced[39] are not only intuitively sensible (in the sense of (43)) predictive probabilities for each such sequence of events, but also are "coherent" [19, 31] to the extent that they constitute a genuine predictive probability *distribution* of the *random variable* X_n, i.e. the additivity and other axioms are satisfied by the recalibrated probabilities assigned to these

[37] To be clear: the question of '*which set* A_n is' may be determined by past observations, but the *outcome* as to whether or not A_n contains X_n will not be determined until X_n is observed.

[38] In the terminology used earlier, we can say that we here interpret 'analogous' events to mean events identically defined in terms of percentiles of the successive c.d.f.s of the raw, continuous PFS.

[39] by selecting different values of p_0 and p

events by the predictive c.d.f. This is a consequence of the fact that, to define a probability distribution for a scalar quantity, it is only necessary to specify some non-decreasing function satisfying the properties of a c.d.f. And we can see that our recalibration procedure—defined by applying (43) to event sequences defined from raw percentiles—turns out (use (43 with $p_0 = 0$)) to simply 'adjust' the raw predictive c.d.f. by composing with a monotonic function as in (34), using (35) with equal weights w_i.

3.7.3 The Problem of 'Missing Events' in the Failure-Count Case, or more Generally the *Mixed* c.d.f. Case

Viewed in the above terms, the problem arising in the extension to mixed or discrete predictive c.d.f.s, can be expressed as a shortage, for some p_0, p pairs, of such past events $E_i(p_0, p)$ which we would wish to have been first predicted, and then observed, by the time a prediction of $E_{n+1}(p_0, p)$ is required. Thus in the general non-continuous case, for some or perhaps all $i \leq n$, the desired percentiles p_0, $p_0 + p$ of the past predictive distributions $F_i^X(\cdot|X_{i-1})$ do not exist. In the case of failure-count observations, this is because the richness of the family of events of the *mathematical process model*, being at this stage naturally the result of an attempt to model the richness of the family of *potentially observable* events in the real world, has been reduced by the move to the cruder, less discriminating failure-count observations. (For failure-counts there may be the additional and separate problem for recalibration, that as well as the absense of exact $100p$-percentiles of the raw predictive distributions for some p and $i = 1, 2, \ldots, n$, there will also be a smaller number n of past predictions than would have been the case if individual inter-failure time data had been collected—Refer back to paragraph 3 on p43, and footnote 32 on p74.)

3.7.4 Solution by: (i) Enriching the Mathematical Process Model; and (ii) Using Posterior Expectations in Place of Full Observations

In §§3.4,3.5 the problem of this deficiency of the supply of previously predicted and subsequently observed events for use on the right hand side of (43) has effectively been overcome by expanding the probabilistic model to include additional events in as "natural" and plausible a way as could be found. Since the selection of the events to be represented within the original model was determined by those events whose occurrence, or not, is fully determinable in the real-world observation process, it follows that the price paid for this extension necessarily involved the problem of unobservable or "fictitious" events having been added. Therefore, in order to include these new events on the right

hand side of (43), a substitute is sought for the now unobservable term I_{E_i}. For this purpose it can be considered that, although such an event E, added in a mathematically natural way to the model, is not observable, this is far from saying that it is thought of by the observer as being stochastically *independent* of all events whose occurrence or otherwise *will be observed* subsequent to the probabilistic prediction of E by an extended raw prediction system which we might apply to the extended model. Hence the added event E may have a *posterior probability*, given a period of intervening observation, which differs from the probability with which it was predicted prior to that period of observation. Then a tentative proposed solution to the problem of the non-observability of E is to replace the now unobservable random variable I_E in (43) by its posterior expectation, given the most up-to-date intervening observations. It is stressed that term "posterior expectation", as used rather loosely here, should not be interpreted with respect to the raw PFS process model $\mathcal{P}$ only, since the whole purpose of the recalibration step is to replace that badly calibrated model with a better one by some kind of ad-hoc remodelling of the process, or some aspect of it, in an alternative, and probably rather cruder way.

To sum up this way of perceiving the recalibration method used for discrete or mixed prediction in §§3.4-3.6, there are two steps to circumventing the problem of the shortage or absence at stage n of historically equi-probably predicted, analogous events $\langle E_i(p, p_0)\rangle_{i=1}^{n}$ compared to the more straightforward situation with the prediction of *continuously distributed* quantities:-

- Embed the process model in a larger one which is richer in events by adding non-observable events. Ideally but not necessarily these could concern some unobservable aspects of the conceptual model of the real world.

- For those terms I_E on the right hand side of (43) for which E is such an added event, replace I_E by the posterior probability $\mathbf{P}^*[E|\text{observations to date}]$. Here, this conditional probability is to be thought of as a probabilistic prediction or, more likely, 'backwards' estimate of the chances of E's occurrence with respect to some process model which may differ in some respects from the raw model being recalibrated—i.e. $\mathbf{P}^*$ here is something analogous to $\mathcal{S}^*$ on p63.

The above way of describing the recalibrator of discrete and mixed predictions developed in §§3.4-3.6 is mentioned in the hope that it may suggest methods of generalising the recalibration technique beyond situations in which the one-step-ahead predict→observe→predict... cycle applies. There is interest in longer term predictions of software reliability and it is tentatively suggested that perhaps the method of using up-to-date posterior probabilities of events in the "recalibration set" $\{E_1, E_2, \ldots, E_n\}$ in place of their observed indicator function values I_{E_i} may provide some kind of

weaker recalibration technique for longer term predictions. We have not persued this idea further here.

3.7.5 Direct Recalibration of Failure-Rate Estimates

That these methods do suggest ways of recalibrating quantities other than the predictive distribution of the next term in the observation sequence $\langle X_n \rangle$ itself is exemplified by the following proposed method of recalibrating point predictions of instantaneous program failure rate (leaving aside the question of how these quantities $r_{n+1|n}$ say, themselves may have been obtained) in the failure-count observation sequence case described in §3.1. In the notation of that section, it may be considered that a raw prediction system, by producing a sequence $\langle r_{i|i-1} \rangle$ of analogously predicted rates at times $l_i; i = 1, 2, \ldots, n$, has produced equal predictive probabilities of a sequence of unobservable events

$$E_i = \left\{ \omega : C\left(l_i, l_i + \frac{\varepsilon}{r_{i|i-1}} \right] > 0 \right\} \tag{44}$$

for $\varepsilon > 0$ small. (Where the operator C is defined on p11.) Then, taking $m_i \frac{\varepsilon}{d_i r_{i|i-1}}$ as a crude posterior probability[40] of occurrence of E_i gives

$$\frac{1}{n} \sum_{i=1}^{n} m_i \frac{\varepsilon}{d_i r_{i|i-1}} \tag{45}$$

or more generally

$$\sum_{i=1}^{n} w_{ni} m_i \frac{\varepsilon}{d_i r_{i|i-1}} \qquad \text{where} \qquad w_{ni} > 0, \quad \sum_{i=1}^{n} w_{ni} = 1$$

as a recalibrated predictive probability of E_{n+1}. This in turn is equivalent to a recalibrated program failure rate estimate

$$\begin{aligned} r^*_{n+1|n} &= \frac{r_{n+1|n}}{\varepsilon} \mathop{\mathbf{P}}_{p^*}[E_{n+1}|\mathcal{G}_n] \\ &= \sum_{i=1}^{n} \frac{w_{ni} m_i}{d_i r_{i|i-1}} r_{n+1|n} \end{aligned}$$

Of course, this is just an attempt to suggest a further example application of the ideas developed in §3.7, whose performance would need to be tested in practice. I.e. we should devise accompanying assessment methodologies for the comparison of the performance of these two failure rate predictors. As a minimal requirement, for example, we might hope that that $r^*_{n+1|n}$ is at least better calibrated than the raw $r_{n+1|n}$ with respect to the sense of calibration used in its own definition. If we use for

[40] Depending on the sizes of these numbers, other estimates such as $1 - \left(1 - \frac{\varepsilon}{d_i r_{i|i-1}}\right)^{m_i}$ might perhaps be preferred, but we could perhaps equally think of ε as sufficiently small to make this unnecessary.

this purpose the same sequence $\langle E_i \rangle$ of (44) in an equation like (41) we find that this comparison is equivalent to a comparison of

$$\frac{1}{n}\sum_{i=1}^{n} r_{i|i-1}\frac{\varepsilon}{r_{i|i-1}} = \varepsilon \qquad \text{and} \qquad \frac{1}{n}\sum_{i=1}^{n} r^*_{i|i-1}\frac{\varepsilon}{r_{i|i-1}} \tag{46}$$

to see which is closest to the empirical estimate (45), where of course the common ε factor, from the derivation, is redundant. (If the right-hand term in (46) is closer (than is ε) to (45), then, in our chosen sense, the recalibrated rate predictor r^* is 'better calibrated' than the raw r.)

Chapter 4

Failure-Count Data Analysis

4.1 Description of Analysis

This section contains results from a number of numerical examples, each consisting of three components: *data* comprising a sequence of failure counts; a *raw PFS*, (one-step-ahead predictor), applied to this sequence; and a *recalibrator* used to modify the raw predictions. The main purpose here is to explore the effect on the predictions of varying the third component, for given combinations of the first two. Accordingly most of the tabulated and graphical results are grouped by failure-count sequence and raw PFS.

The tabulated results, (see Table 3 on p88), consist of values of two scalar indicators of overall predictive quality for complete sequences of predictions. The first of these, "*K-dist*", is simply the maximum vertical distance between the modified u-plot described in §3.5 (with no smoothing, and weights w_i all equal), and the line segment joining the points $(0,0)$ and $(1,1)$. The meaning of this quantity is discussed on p59. The second scalar indicator is a χ^2–type measure of "distance" between prediction and realisation,

$$\chi^2 = \sum_{i=n_0}^{n} \frac{(X_i - \mathrm{E}_{\mathcal{P}}[X_i|X^{i-1}])^2}{\max\{1, \mathrm{E}_{\mathcal{P}}[X_i|X^{i-1}]\}}, \tag{47}$$

where X_i is the failure count in the i^{th} observation interval and, as before, the subscript $\mathcal{P}$ indicates that the expectation involved is the mean belonging to the predictive distribution function, $F_i^X(\cdot|X^{i-1})$, defined by the PFS. (N.B. These notations $\mathcal{P}$ and F_i^X, are not always exactly consistent with the notation of previous sections since the PFS now concerned may incorporate a recalibrator—in which case $\mathcal{P}^*$ and F_i^{*X} used in §3.6 are represented here by $\mathcal{P}$ and F_i^X.) $\mathrm{E}_{\mathcal{P}}[X_i|x^{i-1}]$ is the same one-step-ahead predictive expectation plotted on the graphs collected in Appendix A. Like

K-dist, the scalar χ^2 is given no precise statistical interpretation here since, with the conditioning on previous observations, it is not a standard statistic of goodness-of-fit of a model to data. The decision to constrain the denominator of each term in this χ^2 sum to be at least 1 was taken in order to reduce the sensitivity of this measure to those values of x_i for which the mean $\mathrm{E}_p\left[X_i|x^{i-1}\right]$ of the distribution used to predict x_i is small. Many of the predictive expectations are small in practice, (a fraction of one failure in some of the later intervals when the reliability of the software has improved), so χ^2 is viewed merely as a rough indication of comparative performance of various recalibrators. By analogy with existing theory on more standard uses of χ^2 statistics to test model fit, it seems at least possible that a more reliable performance indicator might be obtained by pooling the predictive means and count observations from a series of intervals for which the predictive means are small. However, in view of the fact that, often, different PFSs disagree as to whether or not this is the case, this would require some thought if the pooling were to be implemented in such a way as to allow consistent comparisons between the performance of a number of different PFSs applied to a single data sequence. We have not explored any such refinements to our primitive χ^2 predictive quality measure and do not assume it to be amenable to formal statistical tests of comparative predictor performance.

In all the examples presented, only those intervals from the 16^{th} onwards were included in the analysis of predictions since the predictions for intervals 6 to 15 were made using the raw (not recalibrated) PFS only—in order to accumulate sufficient observations to form a plot, $u \mapsto S_n(u)$, with which to begin recalibration. So the index variable i in both the above summation, (47), and the summation, (29), used to define the plot from which K-dist is obtained, ranges from $n_0 = 16$ up to the index, n, of the final observation interval.

The remaining numerical results of prediction/recalibration are presented graphically in Appendix A. These take the form of superimposed graphs which can be used to compare different recalibrators on a common data set and raw PFS. The quantities plotted are: (i) *predictive expectation* of number of failures in next interval against interval index, (ii) *log discrete prequential likelihood ratio* against interval index, and (iii) the *modified u-plot*. These graphs are labelled (a), (b), and (c) respectively in all cases. The first is the expectation used in defining the χ^2 distance in (47), and the third is the plot whose maximum vertical distance from the 45°-line is "K-dist". The second is the natural logarithm of the discrete prequential likelihood ratio of the recalibrated vs. raw PFS,

$$\log \mathrm{PLR}(i) \quad = \quad \log \prod_{j=16}^{i} \frac{\mathrm{P}_{p^*}\left[X_j = x_j | X^{j-1} = x^{j-1}\right]}{\mathrm{P}_{p}\left[X_j = x_j | X^{j-1} = x^{j-1}\right]} \tag{48}$$

$$= \sum_{j=16}^{i} \log\left\{F_j^{*X}(x_j|x^{j-1}) - F_j^{*X}(x_j-|x^{j-1})\right\} \qquad (49)$$
$$- \sum_{j=16}^{i} \log\left\{F_j^{X}(x_j|x^{j-1}) - F_j^{X}(x_j-|x^{j-1})\right\}, \quad i = 16, 17, \ldots,$$

using the "*" superscript to distinguish recalibrated from raw predictive c.d.f., as on p63. (N.B. There are also a few logPLR plots used for comparing raw PFSs – see Fig. 29b.) The numerical value of this function at a particular interval index, i, is affected by the arbitrary decision to take the 16^{th} failure count as the starting point for producing recalibrated predictions. More important, therefore, in the interpretation of the log PLR plot is an apparent trend over any sub-sequence of consecutive predictions: upwards indicates that the recalibrated PFS is performing better, and downwards indicates the contrary, over the particular time period concerned. Similarly a trend in the difference between two of the graphs can be taken as an indication that one recalibration procedure is out-performing the other during that period. (See [20] for further details on PLR.) In several of the examples with an unsmoothed recalibrator, a failure-count value is observed after having been predicted with probability zero. When this occurs all subsequent log PLR values are $-\infty$, which is denoted graphically by truncating the graph at the point where this occurs. All prematurely terminated plots in this section have that interpretation.

The *real data sets* employed for prediction are:–

- SS3 – A set of times between failure of a word processing system with many copies in actual use. See Musa [77]. Here we have grouped the data to produce failure counts during intervals of length $d_i = 10^6$ seconds;
- SYS1 – Times between failure of a real time command and control system, also taken from [77], and grouped here into intervals of length $d_i = 1000$ seconds;
- AAA – Data from many operational copies of an aerospace software product, grouped here into intervals of length $d_i = 400$ hours.

The individual inter-failure times of the first two of these data are sets are listed in [1, 65], and on the web at www.dacs.dtic.mil/databases/sled/swrel.shtml.

The following *simulated data sets* were also used:–

- JM1 – Failure-count data generated from the Jelinski-Moranda model (see §3.1.3 on p37) with parameters $N = 106$, $\phi = 7 \times 10^{-5}$, and interval length $d_i = 1000$;

- L1 – Failure-count data generated from the Littlewood model [53] with parameters $N = 200$, $\alpha = 1$, $\beta = 5 \times 10^4$, and interval length $d_i = 1000$;

- LV1 – Failure-count data generated from the Littlewood-Verrall model [62], [65, p105] with parameters $\alpha = 1.5$, $\beta_1 = 30$, $\beta_2 = 5$, and interval length $d_i = 500$.

The *raw PFSs* employed are as follows:–

- DD – the ML plug-in PFS using the likelihood function from the Duane model [24];

- DJMAG – the ML plug-in PFS using the likelihood function from an approximate version of the Jelinski-Moranda model used by Abdel Ghaly, (see [1, p61], and equations (12-14) in §3.1.3 of this thesis);

- DJMNHPP – the ML plug-in PFS using the likelihood function from the NHPP version [32] of the JM model.

In addition to these three, the word "TRUE" sometimes occurs in Table 3 as a PFS applied to those of the failure data sequences which are obtained by simulation. This indicates that the predictive c.d.f. used is "the truth" in the sense that it is equal to the conditional c.d.f. associated with the model used to simulate the failure data. This has been included, wherever feasible, (i.e. for those examples based on failure data *simulated* from a known model, provided the necessary computation is straightforward—specifically, for the JM1 and L1 data sets), as a kind of upper limit on the improvement over raw predictions that can be expected from any recalibration procedure. It has been assessed as a PFS, using the same scalar and graphical tools. See e.g. Figs. 24abc in which the graph labelled "TRUE" corresponds to the use of a one-step-ahead predictive distribution $F_i(m_i; m^{i-1})$ which is binomial with parameters $N - c_{i-1}$ and $1 - e^{-d_i \phi}$, where $N = 107$, $\phi = 7 \times 10^{-5}$, $d_i = 1000$.

The *recalibration procedures* applied are based on various versions of the modified u-plot as described in Chapter 3 and can be distinguished apart by:–

(i) the value of r used to specify the *weights* in equation (35);

(ii) whether or not a *smoother* is used—the label "BSPL50dx" in Table 3 indicates that the smoother of Appendix B is used with $x_j = m_j$, as described on p229;

(iii) the values of the *bounds*, ϵ and $\frac{1}{\delta}$, on *slope* of the smoothed plot, wherever a smoother is used.

The following is a key to the plotted data and results in Appendix A:

Table 2: Key For Data and PFS

Data	Raw PFS	
SS3		Fig. 12
	DD	Fig. 13a †
	DD	Fig. 13b †
	DD	Fig. 13c †
	DD	Fig. 14a
	DD	Fig. 14b
	DD	Fig. 14c
	DD	Fig. 15a †
	DD	Fig. 15b †
	DD	Fig. 15c †
	DJMAG	Fig. 16a †
	DJMAG	Fig. 16b †
	DJMAG	Fig. 16c †
SYS1		Fig. 17
	DD	Fig. 18a †
	DD	Fig. 18b †
	DD	Fig. 18c †
	DD	Fig. 19a
	DD	Fig. 19b
	DD	Fig. 19c
AAA		Fig. 20
	DD	Fig. 21a †
	DD	Fig. 21b †
	DD	Fig. 21c †
	DD	Fig. 22a
	DD	Fig. 22b
	DD	Fig. 22c

†Plot included for raw PFS
‡Plot included for the "true PFS"

Continued...

Table 2: Key For Data and PFS (continued)

Data	Raw PFS	
JM1		Fig. 23
	DJMAG	Fig. 24a †‡
	DJMAG	Fig. 24b †‡
	DJMAG	Fig. 24c †‡
	DJMAG	Fig. 25a
	DJMAG	Fig. 25b
	DJMAG	Fig. 25c
	DD	Fig. 26a †‡
	DD	Fig. 26b †‡
	DD	Fig. 26c †‡
	DD	Fig. 27a
	DD	Fig. 27b
	DD	Fig. 27c
L1		Fig. 28
	DJMNHPP, DD, DJMAG	Fig. 29a †‡
	DJMNHPP, DD, DJMAG	Fig. 29b †‡
	DJMNHPP, DD, DJMAG	Fig. 29c †‡
	DD	Fig. 30a †‡
	DD	Fig. 30b †‡
	DD	Fig. 30c †‡
	DJMNHPP	Fig. 31a †‡
	DJMNHPP	Fig. 31b †‡
	DJMNHPP	Fig. 31c †‡
LV1		Fig. 32
	DJMAG	Fig. 33a †
	DJMAG	Fig. 33b †
	DJMAG	Fig. 33c †
	DJMAG	Fig. 34a
	DJMAG	Fig. 34b
	DJMAG	Fig. 34c

†Plot included for raw PFS
‡Plot included for the "true PFS"

4.2 Results

Table 3: Results of Failure-Count Prediction

Data source	Interval length	Raw PFS	Recalibrator				Pred. Perf.	
			r	Smoother	ϵ	δ	K-dist	χ^2
SS3	10^6	DD					.207	85.6
SS3	10^6	DD	1.0				.237	79.0
SS3	10^6	DD	0.9				.125	72.1
SS3	10^6	DD	0.7				.117	90.9
SS3	10^6	DD	1.0	BSPL50dx	0.0	0.0	.239	73.9
SS3	10^6	DD	0.9	BSPL50dx	0.0	0.0	.122	70.6
SS3	10^6	DD	0.7	BSPL50dx	0.0	0.0	.106	88.5
SS3	10^6	DD	1.0	BSPL50dx	0.3	0.3	.246	74.5
SS3	10^6	DD	0.9	BSPL50dx	0.3	0.3	.154	70.3
SS3	10^6	DD	0.7	BSPL50dx	0.3	0.3	.132	74.4
SS3	10^6	DD	0.7	BSPL50dx	0.3	0.0	.084	83.4
SS3	10^6	DD	0.7	BSPL50dx	0.0	0.3	.160	76.1
SS3	10^6	DJMAG					.245	75.3
SS3	10^6	DJMAG	1.0				.185	73.4
SS3	10^6	DJMAG	0.9	BSPL50dx	0.3	0.3	.145	71.9
SYS1	10^3	DD					.132	63.7
SYS1	10^3	DD	1.0				.082	67.9
SYS1	10^3	DD	0.9				.038	66.9
SYS1	10^3	DD	0.7				.057	70.2
SYS1	10^3	DD	1.0	BSPL50dx	0.0	0.0	.084	67.2
SYS1	10^3	DD	0.9	BSPL50dx	0.0	0.0	.039	66.7
SYS1	10^3	DD	0.7	BSPL50dx	0.0	0.0	.042	69.5
SYS1	10^3	DD	1.0	BSPL50dx	0.3	0.3	.082	67.3
SYS1	10^3	DD	0.9	BSPL50dx	0.3	0.3	.040	66.8
SYS1	10^3	DD	0.7	BSPL50dx	0.3	0.3	.036	68.7
SYS1	10^3	DD	0.7	BSPL50dx	0.3	0.0	.035	68.8
SYS1	10^3	DD	0.9	BSPL50dx	0.3	0.0	.040	66.7
AAA	400	DD					.145	22.9
AAA	400	DD	1.0				.127	21.7
AAA	400	DD	0.9				.095	23.7
AAA	400	DD	0.7				.117	30.6
AAA	400	DD	1.0	BSPL50dx	0.0	0.0	.125	21.9
AAA	400	DD	0.9	BSPL50dx	0.0	0.0	.092	24.0
AAA	400	DD	0.7	BSPL50dx	0.0	0.0	.090	31.3
AAA	400	DD	1.0	BSPL50dx	0.3	0.3	.124	21.8
AAA	400	DD	0.9	BSPL50dx	0.3	0.3	.098	23.3
AAA	400	DD	0.7	BSPL50dx	0.3	0.3	.122	26.2
AAA	400	DD	0.7	BSPL50dx	0.3	0.0	.095	27.7
AAA	400	DD	0.9	BSPL50dx	0.3	0.0	.091	23.7
JM1	10^3	TRUE					.068	18.2

(Columns 4 & 5,6,7 blank indicates no recalibration of raw predictions;
Columns 5,6,7 blank indicates no smoothing involved in recalibrator.)

Continued...

Table 3: Results of Failure-Count Prediction (Continued)

Data source	Interval length	Raw PFS	Recalibrator				Pred. Perf.	
			r	Smoother	ϵ	δ	K-dist	χ^2
JM1	10^3	DJMAG					.288	28.2
JM1	10^3	DJMAG	1.0				.160	21.2
JM1	10^3	DJMAG	0.9				.093	18.7
JM1	10^3	DJMAG	0.7				.042	18.9
JM1	10^3	DJMAG	1.0	BSPL50dx	0.0	0.0	.160	21.3
JM1	10^3	DJMAG	0.9	BSPL50dx	0.0	0.0	.096	18.7
JM1	10^3	DJMAG	0.7	BSPL50dx	0.0	0.0	.046	18.8
JM1	10^3	DJMAG	1.0	BSPL50dx	0.3	0.3	.160	21.4
JM1	10^3	DJMAG	0.9	BSPL50dx	0.3	0.3	.095	19.1
JM1	10^3	DJMAG	0.7	BSPL50dx	0.3	0.3	.061	18.9
JM1	10^3	DJMAG	0.7	BSPL50dx	0.3	0.0	.044	18.7
JM1	10^3	DD					.302	28.5
JM1	10^3	DD	1.0				.131	20.3
JM1	10^3	DD	0.9				.078	18.5
JM1	10^3	DD	0.7				.044	18.9
JM1	10^3	DD	1.0	BSPL50dx	0.0	0.0	.129	20.3
JM1	10^3	DD	0.9	BSPL50dx	0.0	0.0	.078	18.6
JM1	10^3	DD	0.7	BSPL50dx	0.0	0.0	.041	18.8
JM1	10^3	DD	1.0	BSPL50dx	0.3	0.3	.129	20.6
JM1	10^3	DD	0.9	BSPL50dx	0.3	0.3	.077	19.0
JM1	10^3	DD	0.7	BSPL50dx	0.3	0.3	.058	18.9
JM1	10^3	DD	0.7	BSPL50dx	0.3	0.0	.041	18.7
JM1	10^3	DD	0.7	BSPL50dx	0.0	0.3	.054	18.9
L1	10^3	TRUE					.088	37.1
L1	10^3	DD					.161	38.0
L1	10^3	DD	1.0				.061	43.2
L1	10^3	DD	0.9				.050	47.3
L1	10^3	DD	1.0	BSPL50dx	0.3	0.3	.065	42.0
L1	10^3	DD	0.9	BSPL50dx	0.3	0.3	.051	45.9
L1	10^3	DJMNHPP					.064	46.3
L1	10^3	DJMNHPP	1.0				.050	49.2
L1	10^3	DJMNHPP	0.9				.044	51.2
L1	10^3	DJMNHPP	1.0	BSPL50dx	0.3	0.3	.048	48.4
L1	10^3	DJMNHPP	0.9	BSPL50dx	0.3	0.3	.038	50.7
L1	10^3	DJMAG					.098	39.3
LV1	500	DJMAG					.136	60.3
LV1	500	DJMAG	1.0				.143	∞
LV1	500	DJMAG	0.9				.078	∞
LV1	500	DJMAG	0.7				.082	∞
LV1	500	DJMAG	1.0	BSPL50dx	0.0	0.0	.119	46.4
LV1	500	DJMAG	0.9	BSPL50dx	0.0	0.0	.068	52.4
LV1	500	DJMAG	0.7	BSPL50dx	0.0	0.0	.077	50.9
LV1	500	DJMAG	1.0	BSPL50dx	0.3	0.3	.076	48.8

(Columns 4 & 5,6,7 blank indicates no recalibration of raw predictions;
Column 5,6,7 blank indicates no smoothing involved in recalibrator.)

Continued...

Table 3: Results of Failure-Count Prediction (Continued)

Data source	Interval length	Raw PFS	Recalibrator				Pred. Perf.	
			r	Smoother	ϵ	δ	K-dist	χ^2
LV1	500	DJMAG	0.9	BSPL50dx	0.3	0.3	.035	53.5
LV1	500	DJMAG	0.7	BSPL50dx	0.3	0.3	.060	50.2
LV1	500	DJMAG	0.9	BSPL50dx	0.3	0.0	.069	52.3
LV1	500	DJMAG	0.9	BSPL50dx	0.0	0.3	.037	53.7

(Columns 4 & 5,6,7 blank indicates no recalibration of raw predictions;
Column 5,6,7 blank indicates no smoothing involved in recalibrator.)

Of the real data sets, the SS3 data is the one for which the most obvious increase in predictive quality has been obtained by recalibration. Both K-dist and χ^2 performance indices can be greatly improved from their respective values of .207 and 85.6 obtained by, for example, the raw DD PFS. The best values of each from among the few recalibrators tried were .084 and 70.3—although these were not obtained simultaneously. The smoothed recalibrator with $r = .9$, $\epsilon = \delta = .3$ is a candidate for best performance of those tried, on the basis of the values, .154 and 70.3 of these two indices, and also the comparative stability of both the predictions themselves, (Fig. 14a), and the upward trend in logPLR, (Fig. 14b). The success of recalibration here appears to be related rather to the SS3 data set than to the particular raw PFS used for its prediction since similar results were obtained on applying other PFSs, Fig. 16b. Results from three cases using the DJMAG PFS are also presented. On examination of Fig. 12, certain sub-sequences of consecutive failure counts may be identified which are anomalous, (in comparison with the general trend of the whole preceding sequence), in some consistent manner, e.g. data points 15–21—reliability decay; or 33–38—instantaneous increase in reliability. There is arguably some correspondence between intervals of this kind and those over which is found the most marked improvement in PLR for recalibrated vs. raw predictions, (Figs. 13b,14b,15b). It is noted that the modified u-plot for raw prediction of the SS3 data has a pronounced, one-sided deviation from the 45°-line which is matched in magnitude only by the plots obtained from the JM1 simulated data, (Fig. 24c.) These data sets are also identified as two which lead to consistent evidence, from all three methods of assessment (K-dist, χ^2, and PLR), for the success of recalibration. This provides some confirmation of the result, which was to be expected by comparison with the studies on inter-failure time prediction, (see e.g. [6]), that a "bad" u-plot of this kind may be regarded as a positive indicator in favour of proceeding to recalibrate the PFS. (c.f. the poor results for e.g. DJMNHPP applied to the data set, L1, discussed below, in the light of the very different initial modified u-plot, Fig. 29c. But note also the results for AAA, Figs. 21abc

& 22abc.)

Although the results for SS3, along with some of the other data sets, do confirm the advantage of introducing "decaying" weights, (implemented here by taking $r < 1$), it is also evident from Fig. 13a and others that excessively small values of r cause the predictions to become over-sensitive to noise in the last few observations. This leads to a poorer predictive performance of the recalibrated PFS as judged by the χ^2 value. This decrease in quality is often *not* reflected by any increase in K-dist. In fact, some of the best K-dist values are obtained from such recalibrators. This illustrates a point mentioned elsewhere, [56], on the construction of a poor PFS which is nevertheless "well-calibrated". As in the case of continuous predictions assessed by standard u-plots, so here also, the property of being well-calibrated is not, alone, sufficient to demonstrate high predictive quality. It may also be noted that the noisiness of the predictions associated with smaller r can be reduced by smoothing using positive ϵ and δ. (Compare plot 4 in Fig. 14a with plot 4 in Fig. 13a, numbering the plots by reading the line-style keys below the graph from left to right, in rows.) However, it must also be born in mind that larger values of ϵ and δ will tend to inhibit *any* feature of a recalibrator—the extreme case being obtained by setting either ϵ or δ to 1, in which case the recalibrator will "disappear" altogether.

The test results for the recalibrators using SS3 data are followed by the less successful results on SYS1 and AAA data.

In the case of SYS1 data using the DD PFS, the χ^2 value is always damaged by recalibration, the least worse case of those tried being an increase of about 3. As usual improvements in K-dist *are* achieved. The best PLR performances are obtained by setting $r = .9$ or 1.0 and using positive bounds on the slope of the recalibrator, of which the lower bound appears most helpful. With these settings the logPLR plot is approximately level indicating that there is neither gain nor loss in PLR performance resulting from recalibration, (Fig. 19b). In fact the logPLR has a slight positive trend over the last thirty or so predictions. For these better recalibrator settings, the K-dist value favours $r = .9$ over $r = 1.0$, and so, marginally, does the χ^2 value.

The results for the AAA data are remarkably similar to those for SYS1. Again $r = 1.0$ or .9, smoothed with $\epsilon > 0$ gives the best logPLR plots, which have a small positive trend in the later stages of prediction. Again also, of these recalibrators, K-dist favours $r = .9$. But now χ^2 favours $r = 1$. One improvement over the SYS1 case is the actual decrease in χ^2 achieved by recalibration with $r = 1$.

In summary, for the results on SYS1 and AAA data, no decisive improvement in predictive quality is achieved by recalibration, other than an improvement in the modified u-plot, measured by

K-dist. It should be noted that the modified u-plot from raw prediction of the AAA data (Fig. 21c) is consistent with a significant pessimistic bias. This illustrates the need for caution in the interpretation of such a plot which does *not guarantee* that great improvements in prediction are achievable through recalibration. The same conclusion has been arrived at previously for continuously distributed predictions, [6, 59]. The most successful recalibrators for the failure-count sets SYS1 and AAA found here were obtained by smoothing with $\epsilon > 0$ and $r = 1.0$ or .9, these recalibrators being, at least, not significantly worse than the raw PFS—In some respects they are marginally better, particularly in the later stages of prediction.

As already mentioned, the JM1 data set is one on which recalibration proved a definite success. The fact of this data set having originated from simulation makes it is possible to gain an additional perspective on the success achieved. The improvement in performance of the "true PFS" over the performance of the raw PFS can be used as a yardstick against which to measure that achieved by recalibration. The effects of varying the recalibrator parameters are very similar on this data set whether applied to the DD or DJMAG PFS, and all of the following findings are more or less equally true in both cases. It was found that the better recalibrators gave an improvement over raw prediction of the order of one half of the amount by which the raw PFS was inferior to the true PFS, when assessed by logPLR. The K-dist and χ^2 values show an even greater improvement. It appears possible for a recalibrated PFS to do *better* than the true PFS in respect of K-dist, (achieved with $r = .7$). On consideration this is not such a suprising result since a recalibrated predictor is specifically designed to "look back", during each prediction stage, to see how the u-plot is developing and then to adjust the next prediction so as to tend to counteract any deviation from the 45°-line[1]. Another property specific to the K-dist scalar (and the u-plot, from which it is obtained) is that it is not much affected by smoothing or choice of bounds, ϵ and δ, during the recalibration phase. (Though larger values of ϵ and δ were not investigated.) This seems to apply generally with all the data sets. Of the various recalibrators applied to the JM1 data, the χ^2 measure favours $r = .9$ or .7 and seems little affected by smoothing or positive bounds on slope. In one respect this is consistent with the logPLR plot which again appears to favour $r = .9$ or $r = .7$ over $r = 1$. But for $r = .7$, smoothing and the use of $\epsilon > 0$ appears essential to a good logPLR plot. For larger r this appears to become less important until with $r = 1.0$ the logPLR plot is equally good almost irrespective of the use of a smoother. As usual, K-dist ranks the values of r in order with .7 preferred.

[1]The possibility of producing a PFS which is extremely well-calibrated, in the specific sense of u-plot behaviour, if one sets out to achieve *only* this, is easily demonstrated: In fact, taking this to the ultimate extreme, the next U_n value u say in $[0, 1]$ can be *chosen* as desired (i.e. to be the next term of a deterministic sequence $\langle u_n \rangle$) by the forecaster simply defining the next raw predictive c.d.f. to be a constant function of its argument x so that $F_n^X(x|x^{n-1}) = u$ over all conceivable values x of X_n.

The next data set, L1, is also simulated. The results from predicting this sequence of failure counts provide an example of how the choice of raw PFS can influence the success or otherwise of recalibration in the prediction of a single data set. Figs. 29abc compare the predictions from three distinct raw PFSs applied to the L1 data. The predictive mean plots in Fig 29a immediately suggest that, with this failure data, DD is a good candidate for recalibration due to the consistently pessimistic bias in prediction. However this precise information on predictor bias is only available in a simulation experiment. The modified u-plots in Fig. 29c do not suffer from this restriction (except of course the plot from the true PFS), and give the same indication that recalibration is most likely to be successful with the DD PFS. This forecast is proved correct by the logPLR plots in Fig. 30b and the modified u-plots in Fig. 30c. Both recalibrators in which $r = 1.0$ give logPLR values which are about half way between the plots from the raw DD PFS and the true PFS—i.e. a performance roughly level with that of the other two, initially superior raw PFSs. The unsmoothed recalibrator, in particular, has a plot whose trend is as good as that of the true PFS during the period after the initial eight recalibrated predictions. Note that the χ^2 scalar does not provide any support for this conclusion of the superiority of recalibrated predictions (with $r = 1.0$) over the raw PFS. This could be due to the fact that in the definition of χ^2, the denominators of the terms in (47) are decreased by about 25–30% as a result of recalibrating, here. (Compare plots 1 and 3 in Fig. 30a, reading the key at the bottom in rows, from left to right.) For the sake of comparison the results from recalibrating the raw DJMNHPP PFS are also presented (see Figs. 31abc), this being the PFS with the worst indication in favour of recalibrating from its modified u-plot (Fig. 30c). As expected no improvement in logPLR plot was obtained. Slight improvements in the already small value, .064, of K-dist are found, but these are the smallest obtained by recalibrating of any of the examples considered.

Applying the DJMAG PFS to simulated LV1 we find that recalibration achieves some improvement in all three indicators of predictive quality. The modified u-plot in Fig. 33c for the raw PFS indicates optimism of prediction, and this interpretation is corroborated by the tendency of the recalibrated predictive mean failure counts to exceed those of the raw PFS. Many of the previous conclusions can be repeated here. The K-dist values are very much improved by recalibration. However, note here that they seem to favour $r = .9$ over $r = 1.0$ or 0.7, even with smoothing and gradient constraints applied. The χ^2 evidence conflicts with this in preferring $r = 1$. Smoothing and the use of positive ϵ and δ produces the best logPLR plot, with some advantage here being obtained from $r < 1$ with $r = .9$ perhaps slightly outperforming $r = .7$ again, as it does for the K-dist measure. In the case of this data, the raw DJMAG predictor predicts zero failures with certainty (i.e. perfect

software) just once: after the first 5 failure counts only have been observed. This occurs because, at that point in time, the ML parameter fit has $\widehat{N} = 14$, equal to the number of failures that have been observed. That is, after observing the first 5 failure counts, the ML model fit is the "completed debugging" case mentioned at the end of §3.1.3 on p41. One consequence of this over-confident prediction is that when, in fact, two failures of this perfect software are encountered during the 6^{th} observation interval it becomes known with certainty that u_6=1, with the consequence that all unsmoothed u-plots thereafter concentrate some positive amount of mass at u=1 and we arrive at the situation, noted on p65 at the end of §3.6.2, that each recalibrated prediction $F_n^{*X}(X_n|x^{n-1})$ thereafter assigns positive mass to X_n=∞, with the consequence that the predictive means plotted in Fig. 33a are infinite in those cases in which they have been obtained by recalibration from an unsmoothed u-plot.

See also [94][2] for a similar analysis of software failure-counts from actual software, but in a context where inter-failure times genuinely had not been collected.

[2]A correction is needed to results in this PDCS2 project report: Table 1 on p349 contains an editing error in columns 3-8 of the bottom 6 rows, referring to the DJMNHPP predictor. These entries should read as follows (using semicolons to separate rows): blank, blank, blank, blank, .229, 434.7; 1.0, blank, blank, blank, .079, blank; 1.0, SPLINE, 0.0, 0.0, .085, 432.7; 1.0, SPLINE, 0.3, 0.3, .156, 438.0; 0.8, SPLINE, 0.0, 0.0, .103, blank; 0.8, SPLINE, 0.3, 0.3, .173, 409.9

Chapter 5

Prediction Using Additional Sources of Data

5.1 Statement of Problem

Much previous effort in mathematical modelling for software reliability prediction has focussed on the restricted problem of predicting ahead a single process of software failures in time. Thus the data on which each prediction is based consists only of the so far observed part of that process.

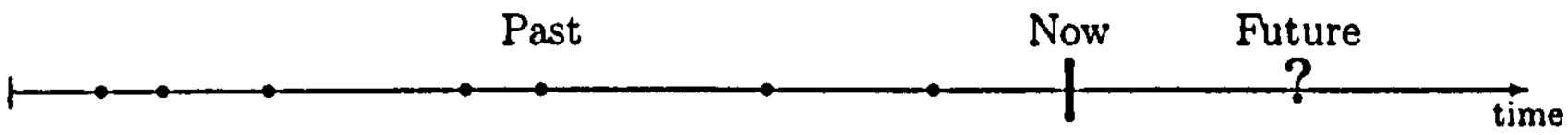


Figure 7: Most extensively studied software reliability prediction problem

Here and in what follows "time" refers to some measure or estimate of the amount of execution to which the software has been subjected. This one-dimensional scalar, which is fundamental to these software reliability models, is intended as a measure of the amount of opportunity which the software has been given to fail through its being employed to perform tasks. A number of alternative definitions may be appropriate depending on the nature of the software and application area as well as the constraints on the recording and extraction of software usage data [68, p170]. Thus observed reliability quantities (inter-failure times, or failure counts vs. time) have typically been used as the sole source of data for predicting corresponding quantities for the same software item at some future point in cumulative execution time, within a relatively stable environment.

There are sound reasons for restricting our formal models to handling this type of problem. It is difficult to relate the predicted reliability behaviour to other quantifiable characteristics of software

products, their development histories and execution environments because of the multitude of factors influencing software's propensity to fail, including for example the complexity and diversity of software and its applications, the variations in performance between different programmers, tools, and development methods, the high degree of design novelty often contained in software. There are simply too many sources of variability between software products, remembering that reliability may well depend on subtle and often poorly understood interactions between these different factors. This problem is compounded by differences between execution environments of a single software component (of which there may be several copies executing at different sites) and also by changes to these environments and their associated execution profiles over time. The term "execution profile" here is to be interpreted formally in terms of the frequency distribution with which a particular usage environment generates different "input points" in the software's "input space". The variability between different execution environments may also be expected to impact on the operational reliability which is observed, [35, 9, 25].

These difficulties have been circumvented by restricting attention in much formal modelling to comparing the reliability behaviour of software with *only its own corresponding behaviour at different points in time* under a *relatively stable execution profile*, and attempting to model the evolution of this behaviour of a *single software product.* For a classification of some such models and associated methods of analysis see e.g. [74, 78, 2] and [65, Ch. 3,4]. Such a restriction achieves a level of stability under which it becomes realistic to expect to discover or confirm the existence of systematic relationships between corresponding random quantities. However, constraining the scope of application of our formal modelling in this way appears to come at a price in terms of the limits of what can be concluded from limited evidence. One argument in particular for seeking to extend formal software reliability models so as to incorporate further sources of evidence concerns the very high reliability requirements of some software applications. Critical systems are coming to depend more and more upon the correct functioning of software to ensure their safe operation. At the same time, the size and complexity of these software subsystems is increasing as designers take advantage of the extensive functionality that software makes possible—functionality that sometimes enhances different aspects of safety.

There are important unresolved questions concerning how one might go about designing such systems so that they will be sufficiently safe in operation. For the purposes of this thesis, our concern is the difficult problem of *evaluation* that they pose. In particular, we are faced with the problem of how to measure the reliability of such a software system when that reliability is likely to be very high.

In several recent papers different authors have pointed out some of the basic difficulties here, [8, 60, 76]. They show that, if we are only going to use the evidence obtained from operational testing of the software, we shall only be able to make quite modest claims for its reliability. For example, Littlewood and Strigini show that even in the most favourable situation of all, that of a system that has not failed during τ hours of statistically representative operational testing, subject to reasonably plausible assumptions, we can draw only the weak conclusion that there is a 50:50 chance that it will survive failure-free for the same time τ in the future.

The limitations here seem intrinsic: they arise from the relative paucity of evidence (when compared with the stringency of the reliability level that needs to be demonstrated) and will not be ameliorated significantly by better statistical models. To make a very strong claim—that a particular system is ultra-reliable—needs a great deal of evidence. If that evidence comprises only observation of failure-free behaviour, then the length of time over which such behaviour is observed needs to be very great. To assure the reliability goals of certain proposed and existing systems, for example the 10^{-9} probability of failure per hour for the 'fly-by-wire' computer systems in civil aircraft [89, 86], would clearly require the systems to be observed *and show no failures* for lengths of time that are many orders of magnitude greater than is practicable.

Faced with these limitations to what can be claimed from merely observing the system in operation, it has been suggested that we should instead base our evaluations upon *all* the disparate kinds of evidence that are available. This seems to be the way in which some safety-critical software-based systems are currently assessed, although it must be said that the process of combining evidence here is somewhat informal and does not generally provide a *quantitative* assessment [34]. The different kinds of potential evidence include, in addition to the operational data discussed above, evidence of the efficacy of the development methods utilised, experience in building similar systems in the past, competence of the development team, architectural details of the design, etc. Most of these other sources of evidence about the dependability of a system will involve a certain amount of engineering judgement in the evaluator, which might itself introduce further uncertainty and potentiality for error. In addition, there are serious unresolved difficulties in *combining* such disparate evidence in order to make a single evaluation of the overall dependability and thus to make a judgement of acceptability.

In this chapter, the question is discussed of what more might be achieved, despite the difficulties, in terms of improving the accuracy of prediction of software failure behaviour by incorporating additional data in the predictions. That is, we will explore methods of supplementing the past failure vs. execution time behaviour data for the software in question executing in its current environment.

This chapter is purely theoretical in the sense that we have not obtained nor simulated any such data. We simply wish to discuss some modelling approaches.

We will consider two ways of making such an extension to the sources of data used for prediction. Firstly in §5.2 we will survey the general statistical models incorporating what we will term *explanatory variables* which we believe might be used in this context. We will attempt to compare the different ways in which such general models might be made to apply to the software reliability prediction context.

What is intended by the term explanatory variables is some collection of observable measures of any of the acknowledged or hypothesised factors of reliability, which could supplement the data on past observed failure behaviour vs. time. We will sometimes use the term *covariate* as synonymous to *explanatory variable.* These observable measures are thought of as deterministic for the purposes of the models discussed here. (Even if they are in some practical sense 'random', we require that their values are *observable* so that any question of how to model their distribution is passed over, as far as this model is concerned, by asserting that all probabilities mentioned in any application of this model are conditioned on one unique realisation of the explanatory variable values which we have been able to measure, observe, or perhaps even 'estimate' using other methods not within the scope of this discussion.)

Secondly we will explore in §5.3 a rather different idea for incorporating data on other software failure vs. time sequences, be they arising from different software products, or different execution environments than the one which we wish to predict (or perhaps even sequences that are different in both of these respects, or arise from different kinds of application, though it will be clear in what follows that we consider the difficulties here to be greater). We will refer to this second approach loosely as a *sequence of similar products*[1]. In this case we will attempt to use 'randomness' or 'statistical indifference' to interpret our notion of similar. The practical interpretation of such a mathematical approach would be that:-

- we have access to several failure vs. execution time data sequences, one of which we wish to predict forward into the future;
- we believe that there are differences between the 'true' reliabilities to which these sequences correspond but we have no idea how to capture the causes or correlates of these differences systematically in terms of any observation or metric that we are able to obtain; and

[1] We use *similar products* as a concise terminology for *similar (product, environment) pairs*, as explained further below.

- we nevertheless do have prior beliefs of a purely probabilistic nature about the possible *distribution* of reliabilities from which we may imagine these unknown, different true reliabilities to be independently drawn.

Thus the distinction in §5.3 from the *explanatory variables* regression approach of §5.2 is that while in §5.3 we still believe that the reliabilities differ and we have prior *distributional* beliefs about the likely size of this difference, we know of no way to systematically *explain, assess, or characterise* these differences—other than the obvious 'suck it and see' method of recording the unfolding empirical reliability of each sequence directly.

5.2 Explanatory Variables Regression Models

The variables contained in most existing software reliability models usually play the role of "model parameters" rather than "explanatory variables"—Their values are estimated from observable data (failures vs. time) during the statistical inference phase of the analysis, rather than being directly observed.

A principle which is often applied in modelling complex and poorly understood "random" phenomena is that of experimenting, at least initially, with models chosen for their mathematical tractability, including both ease of analysis and ease of expandability (flexibility as to number of model parameters). Compare the many applications of, for example, generalised linear models [66], or Box-Jenkins time series models [43]. Motivated in part by this principle, but also by their structural similarity to some of the parametric, single-(product, environment) software reliability models (which do not contain explanatory variables) referred to on the last two pages of §2.2, we consider two special classes of regression model. These have previously proved useful in a number of diverse applications in which an initially identified finite set of "individuals" may each experience or give rise to "events" along some time axis ("time-in-hazard"). As far as these specific models are concerned, the events here are distinguishable only by the individual to which they belong and the time of their occurrence. They can be pictured as points spaced along a horizontal time axis as in figure 7 on p95. However each point is now tagged with the identity of the associated "individual". It may also be necessary to take account of information on "censoring" in the form of known time intervals during which particular individuals have been for some reason withdrawn from observation so that it is not known whether or not they would otherwise have given rise to any observable events during these periods. The purpose in such applications is to obtain predictive expectations, higher moments, or distributions for the times of events not yet observed, based both on the number and time-spacings of earlier observed events *and also* on available measurements of *other characteristics of the individuals* in the population whose event times are being observed and predicted. In view of the kind of advantages mentioned at the beginning of this paragraph, it is not surprising that models from the two classes on which attention is focussed in this section have tended to be applied in situations where the causal mechanisms relating event times to individuals' characteristics are complex or poorly understood, perhaps involving a human element (e.g. insurance claims), or some problem in the 'softer' sciences such as medicine (e.g. the relationship between the treatment regime, age, history or lifestyle of each patient and the patient survival time or time to the development, reappearance or disappearance of some symptom, in medical research). In the case of software, the process of failures in execution time can be conceived as a result of the interaction of two other

processes, at least one of which can be argued to be of a similar (i.e. complex, poorly understood) nature. These two other processes are: (a) the process of software development and maintenance and in particular the human reasoning and other errors and communication difficulties during these activities, and (b) the process of the application domain which determines the sequence of inputs to the software. This observation does not guarantee that the same general regression models will be found useful here. However, the model fitting and validation techniques and the supporting statistical theory are now well developed for the two classes of statistical models under consideration, providing another argument to suggest that they may deserve empirical assessment in the software reliability application (exploiting the body of existing knowledge about these models gained through experience of their use in other applications).

The importance attached to obtaining the most trustworthy possible software reliability predictions is well justified in [78, pp21–29], [30, §11.2] for purposes such as estimating cost, effort and time scales in development and maintenance, or rigorously evaluating, where feasible, the various supposed factors in software development of final product reliability. There seems to be general agreement that many known characteristics of product, development, type of application, and execution profile have some kind of impact on failure rate vs. execution time. However there is a need, if possible, to develop a quantitative understanding of these effects. Some previous work along these lines is briefly discussed in §2.6. It may be that some improvement in quantitative understanding could be obtained using the kind of general purpose regression models outlined here, rather than, perhaps prematurely, attempting to model the exact causal mechanisms at a more detailed level—which attempt in any case suffers from the drawback that, because of the novelty and variety of software, it might lead in the direction of a different model for each development environment, each application domain, or even for each new piece of software. For example, one can imagine that a Markov assumption for the flow of control between sub-modules or sub-functions, e.g. [51], might be acceptably accurate for certain products and applications but not for others. Where it can be exploited successfully, the accuracy of such an assumption might require different notions of "state" and different partitioning between states depending on the nature of the software and application. This is not to deny the possibility of similar problems with the assumptions of the regression models and the choice of their explanatory variables; but rather to suggest that, when such problems are encountered, the flexibility of these model classes facilitates the task of exploring a variety of different hypotheses using established rigorous techniques of model validation. A brief discussion of methods of checking model assumptions, with references, is given in §2.7.3.

Mention should be made of the prior need for well defined, consistent, and preferably cheap

methods of *measuring* characteristics of software (including its development history) and its environments before searching for empirical relationships of these characteristics with reliability. One of the reasons that direct measurement alone is not sufficient, however, is that many of the attributes, like reliability, which are of greatest practical interest are actually most valuable if available in the form of predictions at an early stage, [30].

A schematic outline of the broad question being addressed in this section is therefore as follows: Can probability statements of the form

$$\mathrm{P}[\text{Future reliability behaviour of A ; Past reliability behaviour of A}]$$

be improved upon by probability statements of the form

$$\mathrm{P}[\text{Future reliability behaviour of A ; Past reliability behaviour of A, } x]\,?$$

Here, A is some software component and/or particular circumstances of its execution, "reliability behaviour" refers to some aspect of the resulting point process of failures in execution time, and the "explanatory variables" vector, x, comprises some set of *other data* supplementing past reliability behaviour of A. Discussion of exactly what characteristics and measures x may contain follows later. The items to the right of the semi-colon represent the data required to calculate the predictive probabilities of events describable in terms of the variables represented to the left of the semi-colon. In some implementations of this scheme, it may be possible to interpret these probabilities as conditional (replacing the semi-colon by "|", read as "given"[2]). However, it is preferred to describe the situation in more general terms. For example a model-fitting procedure may intervene between the observation of the data on the right and the calculation of a new prediction as a conditional probability. The precise calculations involved will be decided by a combination of (a) an informal understanding, however approximate or incomplete this may be, of the causal mechanisms leading to the occurrence of software failure, in which the variables, x, are believed to be either causal factors or merely indicators correlated with future failure behaviour; and (b) the outcomes of more formal investigations involving the tailoring of hypotheses or adjustment of models to fit empirical data. On the basis of (a), some kind of structural model may be employed which attempts to mimic mathematically the real-world connection, so far as this is understood, between the quantities, x, and the process of software failures vs. execution time. In (b), the term *model fitting* will now typically involve the fitting of *multiple* software failures vs. time data sets by a *single* enlarged model, where

[2]E.g., by a procedure analogous to that given in (16) on p47, it *may* be possible to define a process probability law (such as $\langle \Omega, \Sigma, \mathrm{P} \rangle$ on p9) with respect to which these predictive probabilities do indeed qualify as true *conditional probabilities*, satisfying the properties referred to on p10. However, in view of the difficulty [92] of understanding the relationship between prediction system and process probability distribution, it would be unnecessarily restrictive to exclude situations in which no such probability distribution for the failure process is available.

the different data sets each have an associated vector of explanatory variable values. Here we are thinking of the kind of analysis proposed in §§5.2.1.1 & 5.2.1.3 where we would expect to model *multiple* data sets in the sense that we can think of failures of each *individual* as comprising *one* data set of the kind modelled in isolation in Chapters 3 and 4 of this thesis and also in much previous work, such as [2, 56, 1, 7]. (Contrast this idea with the work in [2, 7] in which *a single data set* of this kind is *multiply fitted* using an array of *different* point-process models.) There are sufficient software reliability growth models and sufficient software failure-time data sets available now and which have been applied one to another in pairs in sufficient previous work [10, 2, 7] that these approaches to the extension of this work have seemed naturally to suggest themselves as worthy of some consideration. For both (a) and (b), we will require both a *faithful representation of some of the complexities of reality*, and *mathematical tractability* facilitating subsequent analysis and validation of the model. These are two important but frequently competing requirements. The requirement for *data for comparing* software items or their environments, to play the role of explanatory variables, is discussed briefly in §5.2.3.

5.2.1 The Role of Individual

The idea of "explanatory variables modelling of software reliability", as described above in §5.2, admits a number of distinct more precise interpretations. Here, three application categories are distinguished which enable a crude classification of some ideas and suggestions found in previous work. These three categories correspond to distinct notions of "individual", in the terminology of the statistical models discussed above. (Although neither this terminology nor a restriction to PHM/PIM models are essential to any of the three categories which follow.):–

5.2.1.1 Product

An application in which "individuals" are identified as *different software products* is perhaps the first to come to mind when thinking of extending the sources of data input to software reliability prediction systems beyond the single one of the so-far-observed portion of the point process of failure vs. time. Can a model be developed which helps to *explain reliability variation between software products* in terms of measurable characteristics of those products? There is general agreement for many attributes of software products that such attributes must have some kind of impact on, or at least correlation with, product reliability. These attributes include basic classifications of software, such as source language used, type of application, as well as internal structure metrics such as size and complexity metrics, and metrics or classifications relating to the development process of the

software (e.g. tools, methodology and personnel employed, or resources consumed).

One way of including such information might appear to be to use models for "prior belief" (Q on p16) about the values of the parameters of existing software reliability models. For example, a modeller might claim to be able to form a prior distribution for the two model parameters, N, ϕ, (see p37 and [36], [74, p13]) of the Jelinski-Moranda model in terms of measurable characteristics of the software product and its development, such as size in lines of source code, complexity as defined by the McCabe metric, rate of finding bugs per person hour of code inspection, etc. A Bayesian inference procedure could be applied to the Jelinski-Moranda model in which these explanatory variables would enter the reliability predictions via this Bayesian prior. However, doubt has been cast on the interpretation of actual fitted parameter values of some existing models as realistic estimates of internal software attributes (e.g. of N as "number of faults present in software" for the Jelinski-Moranda model). Such doubts are frequently confirmed by the instability (i.e. extreme variability) of the estimated parameter value in successive model fits as additional observed reliability data is accumulated from one single executing software product.

5.2.1.2 Fault

A PHM analysis of times to software failure with "individuals" identified as single faults in one software product could perhaps be seen as a limiting case of the first category, described in §5.2.1.1 above, applied to subcomponents of software where the size of these subcomponent-individuals has been reduced to the point where each is either fault-free or contains exactly one fault. But this analogy with §5.2.1.1 is actually very weak for at least two reasons: The first is the impossibility of associating certain software faults with any particular location or subcomponent of the executable software product, (e.g. defects in the higher level design). The second important difference between §5.2.1.1 and §5.2.1.2 relates to the observability of the population under study. This is discussed further below. Examples of characteristics of faults are: stage at which introduced, severity/cost of consequent failure, type of human error (e.g. typographic slip).

5.2.1.3 Operating Environment

A *single software product*, once having been developed, will usually be released into a number of *sites, or different usage environments.* Thus, the "individuals" of this model consist of different copies of one single piece of software, or more accurately the execution profiles associated with the distinct sites to which these copies have been shipped. An added opportunity with this kind of application arises from the time-variability of some measures of software operational environment.

This suggests the possibility of applying a version of one of the statistical models which is able to incorporate time-varying explanatory variables ($x_i = x_i(t)$), e.g. the PIM model [83, 81].

5.2.1.4 Discussion of Plausibility of Different Approaches

A serious reason to doubt the likely predictive capability of explanatory variables software reliability models of each of the above three categories concerns the regularity and stability of the population under study. Suppose that a "significant" statistical relationship between explanatory variable values and reliability behaviour is discovered to hold for a sample of individuals used to fit the model. (In fact such a relationship is very likely to be discovered if an approach is taken of persistently trying one hypothesis after another until "success" is achieved. This approach is assisted by the general purpose nature of linear regression models. It is clearly preferable that there should exist in advance some small number of well considered beliefs or hypotheses that specific sets of covariates will collectively have a particular kind of impact on reliability.) Then what are the chances of that relationship continuing to hold: (a) in the future for the same sample of individuals, and (b) for a new individual? Since there seems to be general agreement that many explanatory variables are not independent of reliability, this question must relate to the possibility of "unobserved heterogeneity" within the population: The sources of variability, amongst the population of individuals observed, which determine reliability variation may not all have been captured by the explanatory variables used—Or rather, since that is inevitably the case, a sufficient number of the more important sources may have been omitted, or inadequately captured, to the extent that their effect is too large to be successfully represented as noise. (By "noise" is meant the random variation allowed within the model, by the probability distributions which the model incorporates.) The other possible source of failure—that the model class, though flexible, is not flexible enough to represent the mathematical form of a true relationship to a sufficiently close degree of approximation—is difficult to eliminate when applying a general purpose model specifically because our understanding of the real relationships is poor. One can only point to the success of the model in previous applications. Returning to the possibility of a true relationship, adequately modelled for a restricted set of explanatory variables, but obscured by unobserved heterogeneity in the form of other individual characteristics which have been missed, there seem to be three positive responses to this problem before abandoning the model altogether: (a) try harder to define and measure more explanatory variables which capture the remaining variation, (b) fit the model to a larger number of individuals and hence detect and verify weaker relationships than would otherwise be possible, or (c) restrict the field of application of the models to populations which are less heterogeneous. (The extreme case of the latter, where

most effort has been concentrated up to the present time, is to model only a single software item within a single, stable environment.)

The above mentioned problem of an overwhelming multiplicity of sources of heterogeneity appears particularly true for the application category of §5.2.1.1. Apart from anything else, if no restriction whatsoever is imposed on the population under study, then the variability between *application tasks* alone—never mind the software systems developed to address the task—is very great. Perhaps some success could be achieved by restricting to sets of software products built for a single application or by a single developer.

Model applications of the category described in §5.2.1.2 involve a different sampling rule which is reflected in a crucial difference in the probabilistic modelling and statistical inference problems. In most software which is of practical interest, if faults are the "individuals" then there will remain an unknown number of individuals which have been "in-hazard" for the period of observation and which have given rise to no event during that period, (i.e. a number of faults which remain undiscovered, not having caused a detected software failure during the observation period). The sub-population of faults which *do* cause a detected failure, during the period of software usage, is far from being either a random sample, or a sample whose members can be identified by the observer in advance. It is in fact a self-selecting sample (i.e. the chance of an arbitrary member of the entire population of faults being included in this sample depends on its covariate values[3]). Some further probabilistic model of the entire fault population, including some assumptions about the number of hidden faults and their unobserved covariate values, would be needed in order to develop a likelihood function which properly takes account of the existence (and non-occurrence during the period of observation) of these faults. The non-occurrence of events for an individual can clearly be, as far as inference about the population is concerned, one of two quite different things depending on whether or not the observer has knowledge of the existence of that individual, and of the fact that the individual has been 'in-hazard' during the period of his or her observations. In the terms of the medical analogy, we are dealing, in §5.2.1.2, with a situation in which a researcher remains ignorant as to the number of patients involved in his study, and only discovers the fact that a patient exists (let alone the details relevant to that particular patient, such as treatment administered) if and when that patient dies (or experiences a medical 'event' however alternatively defined) during the period of the study. For prediction of future reliability it is precisely the population of hidden faults, of unknown number and covariate values, which is of interest. Bearing in mind the non-observability both of the population size and of individuals' characteristics for this population, it is nevertheless

[3] the values of its 'explanatory variables', as we would observe them them, were the fault's existence to be made known to us.

the case that many existing software reliability models which are used for reliability prediction without explanatory variables do rely on a model of the entire fault population of the software, including as yet undiscovered faults. This model typically represents the size of this population and the distribution of its fault rates, [53, 74]. Perhaps a study of the manifestation rates of software fault populations regressed onto other individual fault characteristics could ultimately help to validate or improve upon such models, or to learn more about the effect on reliability of different phases of software development. However, with realistically sized software, such a study would always have to contend with the difficulty of working with a self-selecting sample from a concealed population. Software reliability prediction theory might perhaps benefit from examining techniques used in other application areas for inference from analogous data, if such statistical applications can be identified.

Models for the category of applications described in §5.2.1.3 include structural models such as Littlewood [51] in which the operational environment of the software is characterised directly in terms of the software's response to that environment. Specifically, statistics on the flow of control through the software are utilised, via code instrumentation, as observable characterisations of the software's current environment. For example the proportion of time spent in each module, or a matrix of module transition probabilities can be regarded as measures of the environment. The reasoning behind this kind of environment measure is that the failure rate due to a bug will probably (though not certainly) increase with the exercise frequency of the code containing it. Regarding PIM as a crude general purpose model, it would be interesting to examine detailed structural models: (a) to find out how mathematically distant are their theoretical conclusions from a PIM mathematical form, and also, (b) to compare their predictive performance, using real data, against that of PIM models incorporating the same metrics, i.e. to use PIM models as a standard of comparison for predictive performance of such more detailed structural models.

Of course, there are many other means of characterising the environment of a software product, without working in terms of the structure of the software itself, such as classifications like commercial/industrial/academic, type of hardware, number of simultaneous users [35, 73], type of user applications, or some measure of the diversity of the user population. The use of a binary test/usage explanatory variable would be equivalent to fitting a 'testing compression factor' as in [78].

In the environments-as-individuals category it may be possible to carry out inference using fewer (or even one alone, [35]) individual environments. The variability of explanatory variables ($x_i = x_i(t)$) required to achieve a fit might be present within a single environment. There is a question about the granularity of the logging of time-varying $x_i(t)$ in practice. e.g. there may be a diurnal cyclic effect—Ought this to be ignored by logging only totals per day, or should the daily variability

be used to assist with the model fitting? How should time-varying measures be smoothed before use in the model?

There is also a question about how the global time metric, required by a PIM model, should be defined for software of which several copies (the "individuals") are running in distinct environments. The obvious solution is to assume that each copy of the software will produce failures mapped onto a local time line in the form of its particular host-machine's cumulative execution time of the specific software concerned. The single global time required by the model for each failure event can then simply be defined as the local time at which that fault occurred on the host which experienced it. However, this raises questions if any evolution in reliability is occurring following maintenance and debugging actions, since, for example, a fault could be removed in one environment "before" (model global time) its time of discovery (by manifestation) in another environment. Thus using global time, defined in this way, for the single horizontal axis of the statistical model, may result in switching the chronological order of two events from their calendar-time chronological order, if these events belong to distinct individuals (environments). Clearly this may have implications for the kind of baseline model which will be appropriate. It appears also that the relative appropriateness of PIM or AFT models for reliability variation between environments would be effected by (a) the definition of failure time as discussed here, and also (b) the degree of coordination of debugging and other maintenance activity between different environments. PIM is able to model variation of failure rate between individuals, whereas AFT will model a related variation between individuals of both failure rate and reliability evolution rate. (Compare equations (4) and (5) in §2.7.2.)

Intuitively it seems likely that there will generally be greater uniformity for a single software product operating in several distinct environments, than for a number of totally distinct products. (The precise assertion suggested here would be that there is 'more variability' between software products than between software environments of a single software product.) Consequently it is to be expected that practical success in fitting a general model using explanatory variables in order to predict reliability is more likely for the category of applications discussed in §5.2.1.3 than for models of the variation between products discussed in §5.2.1.1. One hope is that a model of this third category would enable reliability observations during testing to be used as a basis for the prediction of subsequent operational reliability in various usage environments.

5.2.2 Use of Recalibration

Concerning recalibration we remark here that—although in Chapters 3 & 4 as well as in previous work [6, 10, 11, 20, 59, 7, 56], recalibration has tended to be applied in the context where a sequence

of scalar values is predicted, using a PFS as defined in §2.4.2, one-step-ahead using past observation of that process as the *sole* source of data—there is no reason why the incorporation of further data as discussed in the current chapter should preclude the use of a similar recalibration technique to achieve improvements in predictive quality. Indeed all that is essential for the construction of some technique of recalibration—applied to prediction incorporating explanatory variables used to model the differences between the failure-vs-time behaviour within a family of individuals—would be that we should remain within a setup where:–

- we can accumulate a stock of a reasonable number of *observed values*[4] each of which had previously been predicted in the form of a predictive distribution for that value; and
- we continue to believe that an analogy or similarity holds between the relation of each of these successive predictions to the reality of what it predicts[5], yielding a systematic aspect to the prediction error (distribution of the u) produced at each prediction step.

For example, one could imagine using a PIM model to track the reliabilities of several separate installations of one software product. Then we could still observe inter-failure times (or failure-counts) at each installation and hence build up a sample u-plot from the pool of u-residuals so obtained. Whether to construct a separate u-plot from the residuals for each individual, or to pool the entire set and use a common plot for the recalibration of all, or even to construct some more complicated analysis of the joint distribution of a family of u-processes would be an interesting question. An appropriate decision about this would probably depend on details of the distinction between individuals and on the quantities of data available. One can imagine, for example, that imperfections in the regression part of the model could result in systematic differences in the shape of u- distribution emanating from the different individuals. For a PHM rather than PIM model we would be faced with the problem of having only one u from each individual, so that the us would inevitably have to be pooled. But it should be clear from the previous parts of this chapter that we feel that some kind of PIM model would be more likely to be of use in the software reliability application.

5.2.3 Acquiring Data

A problem of equal if not greater difficulty to the theoretical problems of whether in principle and, if so, how explanatory variables could be included in software reliability predictions appears to be the

[4] We suggested during the discussion beginning on p78 that even a posterior distribution of some kind—in place of exact observation of the predicted RV—might be used for a kind of recalibration.

[5] See list on p76

more practical problem of getting some data to test such models. See e.g. [4, p61]. Perhaps a next step would be to produce some more specific requirements of (a) what kinds of data are most likely to provide positive results, and (b) approximately how much data from how many individuals would be required in order to objectively verify any relationship with reliability which exists. Clearly the number of distinct individuals on which data can be obtained imposes some kind of limitation on the number of parameters which can be included in any model.

One possible source of data is experimentation, which suffers from the drawbacks of expense and perhaps unrepresentativeness due to the smallness of scale of the problems which can be tackled in an experiment. It seems probable that many factors of software reliability will undergo shifts in relative importance as applications of different scale and complexity are considered. On the positive side, with an experiment it becomes possible to attach greater importance to accurate measurement, this being the main purpose of the exercise. There would be less concern about the cost of degraded system performance resulting from the additional burden of measurement and collection of data. Also, particular care could be taken to reduce or randomise over sources of heterogeneity not represented in the model, and to apply statistical principles of experimental design.

As a final resort, perhaps there may even be limited possibilities in *simulation* of software failure processes and explanatory variables.

5.3 'Similar Products' Model

In this section we shall consider a different approach to the situation where we wish to augment the evidence that can be gained from the operational testing of a particular product within a particular environment by also taking into account the data on success/failure sequences either of other products or of the current product executing under operational conditions which differ from the present conditions. Thus these other failure data sets may be records of the success (or not) in building and operating 'similar' products in the past. Alternatively, they may originate as records of execution of the current product in different environments. An important special case, of course, is the one where there is unreserved good news from these previous data sets—i.e. where there have been no failures in any of the data sets up till the present time.

As before the goal is to obtain a *quantification* of the reliability of a product within an operating environment. The model that is developed in this section, therefore, requires us to make certain assumptions about the failure process, and about how we represent our beliefs about certain model parameters. Essentially we have replaced the assumption of the last section that meaningful 'explanatory variables' vectors are available by assumptions about the availability of probabilistic prior beliefs of a particular mathematical form concerning the distribution of reliability variation between our 'individuals'. We acknowledge that these assumptions can be questioned, and are certainly very difficult to validate. However, we believe that they are reasonably plausible. More importantly, our main aim is to demonstrate that this kind of evidence can only improve our confidence in the reliability of a product quite modestly. Thus, we would regard a critique of our results on the grounds that they are not sufficiently conservative as being in the spirit of our own aims; suggestions, on the other hand, that the assumptions here can be modified in order to arrive at much higher confidence in product reliability we would regard with suspicion and we believe would demand careful analysis of extensive supporting data. It seems to us that, particularly in the case of safety-critical applications, it is safest to adopt a conservative view of the informativeness of evidence unless there are scientifically valid reasons to believe the contrary.

The model contained here may be applicable either to the data sets arising from a number of different software products, or from a single software product executing in a number of different operational environments. (Compare subsections 5.2.1.1 and 5.2.1.3 of §5.2 on *explanatory variables regression models.*) The 'indifference' assumption discussed below is all that is required in either case. Thus we can think of our 'experimental unit' or 'individual' as a particular product operating in a particular environment. From each such (product, environment) unit, we observe operational data. For mathematical simplicity we have chosen to work now in terms of the 'discrete time' of

§2.1—a sequence of discrete demands on the software, each resulting in success or failure—so that the operational data arising from each ⟨product, environment⟩ unit is a binary sequence. We see no obstacle in principle to using the approach presented in this section with failure times or failure count data of the kind used in previous sections of this thesis. We are interested in using data from a family of such units to improve our ability to forecast a particular one of them, i.e. a particular one of the binary success-failure sequences. Here, we make what we see as the simplest assumption which allows this kind of learning from one sequence to another: An assumption of prior indifference between the members of our family of operational success-failure sequences. For example, we assume that we are without prior[6] beliefs of any kind which would cause us to identify some particular pair $\mathcal{A}$ and $\mathcal{B}$ of testing data sequences about which we could say 'I expect that sequence $\mathcal{A}$ will show greater reliability than sequence $\mathcal{B}$'. Our model now explicitly says that such beliefs about comparisons of reliabilities between different sequences will emerge only after we begin to examine the numerical values of the failure-counts which those sequences contain. In particular, we use no other observable characteristic or measure in terms of which to differentiate between the expected reliabilities of these sequences: There are no 'explanatory variables' now. With this understanding of our meaning, we refer in what follows to a set of success-failure sequences about which we feel this indifference as a family of 'similar' sequences (emanating from a family of 'similar' ⟨product, environment⟩ pairs). So in our usage here the 'similarity' of the success-failure sequences within a family is nothing more than a prior statistical indifference between these sequences. Of course, this idea might elsewhere be extended by means of an ordering of the distinct sequences and some kind of process model for, say, increasing reliability expectations from one ⟨product, environment⟩ to the next. But our indifference assumption is simpler, whilst allowing us to explore mathematically the learning which might take place from one sequence to the next and, we believe, being a plausible model in some circumstances. In particular, even this simple model well illustrates the importance of prior belief—about the statistical relationship between these failure-success sequences—for any conclusions we might wish to draw about one such sequence from data on other products or operating environments.

In the next section a doubly stochastic Bayesian model of the failures (if any) of a family of 'similar' software success-failure sequences is constructed. The intention is to augment the relatively meagre evidence that can realistically be gained from testing of a particular product in a single environment. We can now take account also of the success (or not) in conducting similar operational trials in the past. The analytical results which follow in §5.3.2 lead to an examination of an important special case in §5.3.3. §5.3.3 explores the conclusions which can legitimately be drawn

[6] prior to observing the success-failure data itself

from observation of a number of sequences *all* of which contain *no* failure up till the present time. We examine this no-past-failure case in some detail, and, after a brief enumeration in §5.3.4 of some practical questions whose answers our model might be used to explore, we proceed in §5.3.5, to obtain some analytic and some numerical results for a few example cases of our general model. In discussing these special cases which arise from introducing specific parametric distributional assumptions, we concentrate mainly on the no-failures case introduced in §5.3.3. Some considerations about the difficulties of choosing a measure of reliability are mentioned in §5.3.6. Our main conclusions are summarised in Chapter 6. The appendices contain some of the mathematical details required by this modelling approach, including, in Appendix B.5 our procedure for calculating very high order non-central moments of the Beta distribution which we used for the numerical work of §5.3.5.2.

5.3.1 Modelling Approach

We wish to use evidence we have obtained from building and operating previous products, or from previous operational use (in different environments) of our current product, in order to try to improve the accuracy of the predictions that we can make about the reliability either of an entirely novel product or a previously used product which now operates in a novel environment. To do this we must take account of two kinds of uncertainty. In the first place, there will be uncertainty concerning the actual reliabilities that were achieved by these earlier ⟨product, environment⟩ applications. Even in those cases where there is extensive operating experience, we shall never know the true reliability of a given product in a given environment and will have to use an estimate based upon the finite amount of operational data collected during its use within the environment concerned. In those situations where we are dealing with products that are likely to be very reliable in their intended environments, we shall probably only see a small number (or even none at all) of failures even in quite extensive periods of operation. The second source of uncertainty will concern the statistical 'similarity' to one another of the success-failure sequences that have been observed in the past and the 'similarity' of the one under study to these past sequences. Clearly it will be misleading (and give optimistic results) if we simply assume these earlier sequences, and the present one, are 'exactly similar' in the sense that they all arise from exactly the same true reliability [47].

In what follows, we shall assume that the true per-demand probabilities of failure of the different sequences, past and present, can be assumed to be realisations of independent and identically distributed random variables.

This assumption, although an idealisation, captures the essentials of what we mean by 'similarity'. Thus, it means that the actual reliabilities of the different sequences will be different, as is clearly the case in reality. We would not expect the reliabilities of, say, two versions of a software-based telephone switch to be identical, even though we might be prepared to agree that the problems posed, and the quality of the processes deployed in their solution, and the operational environments in which they are situated were similar. The notion of 'similarity' in the eye of an observer here seems to be equivalent to a kind of 'indifference'. You might agree that two different success-failure sequences were similar for the purposes of the current exercise if you were indifferent between them in reliability terms: if you were asked to predict which of two ⟨product, environment⟩ pairs $\mathcal{A}$ and $\mathcal{B}$ would show the best reliability, you would have no preference. This is represented by their probabilities of failure per single demand being identically distributed random variables: any probability statements you would make about the reliabilities of demand sequences $\mathcal{A}$ and $\mathcal{B}$ would be identical. The important point here is that this interpretation of 'similarity' in terms of indifference does not mean that you

believe that the two sequences will show identical reliabilities - indeed you will know that the actual reliabilities of the sequences will differ. The two sources of uncertainty here are both important. However, it is the nature of the uncertainty concerning 'how similar' the sequences actually are that will be most difficult to estimate in practice, since this requires us to see as many different sequences as possible. That is, we would require operational data on a large number of (product, environment) pairs between any two of which, prior to inspection of actual failure data, we felt indifferent. But in practice, it is far more likely that we have large quantities of testing information about a few (product, environment) combinations than it is that we have information on many such testing sequences.

Consider first the failure process of a *single* software success-failure sequence $\mathcal{A}$. Assume a Bernoulli trials process model of the failures of this (product, environment) in a sequence of 'demands' with neither debugging, maintenance, nor significant variation in the 'stressfulness' of the software's operational environment. An example might be the installed software protection system of a nuclear reactor, where demands could be assumed to be sufficiently separated in time as to be treated as independent. For flexibility of expression we will use $\mathcal{A}$ to refer to both the (product, environment) pair and the 'sequence' of successes and failures on successive demands on this software in this environment. Then strictly, 'sequence' means the exact *probability law*[7] *governing the sequence*, rather than the realised boolean values of the sequence. With this understanding, we can refer to $\mathcal{A}$ sometimes as a 'pair', at other times as a sequence, and even as a single number p which we have an interest in estimating as accurately as possible. Thus, in the first n trials of sequence $\mathcal{A}$, let R be the random number of failures occurring and p be the probability of failure on demand. Then the distribution of R for fixed n and p is binomial:

$$R|n,p \sim \binom{n}{r} p^r (1-p)^{n-r} \tag{50}$$

Now think of p as unknown and construct a Bayesian model by assuming that p is a realisation of a random variable P having a parametric distribution

$$P|\theta \sim f_p(p|\theta)$$

with parameter θ, possibly a vector. Here we can think of the shape of this distribution $f_p(p|\theta)$ for P as a representation of the general reliabilities of sequences in a particular *family of (product, environment) pairs*, perhaps representing the different failure histories of a single product executing in multiple environments. Alternatively this family might consist of the failure histories of a number of 'similar' products produced by a single development team, using a common development

[7] by assumption in our model a Bernoulli trials process completely specified by a single numerical parameter p

method, and for similar applications. For example, a family of data sequences known to have highly variable reliability levels would correspond to a distribution $f_p(p|\theta)$ with a large variance, whereas, for another family of sequences, an expected high 'average' reliability figure would correspond to a small mean for $f_p(p|\theta)$. If we fully understood the true variation in reliabilities of the sequences in each of these two success-failure sequence families then we could describe the two families by specifying two different P-distributions having the required characteristics, and index these P-distributions with two different θ-values, θ_1 and θ_2, say. More generally, our parameter space Θ, say, for θ, could be said to represent a set of different conceivable reliability characteristics each of which potentially characterises a different *family* of ⟨product, environment⟩ pairs. That is, given sufficient data on the reliability variation amongst the sequences of a particular family, a value of θ (and hence a particular distribution $f_p(p|\theta)$) could in principle be assigned as descriptive of that variation. In this way, we have defined a model in which θ can be thought of as a parameter characterising a family of ⟨product, environment⟩ pairs. For a ⟨product, environment⟩ chosen at random from those of a particular family (i.e. particular θ) and observed for the first n demands, it follows that (R, P) has joint distribution[8]

$$(R, P)\,|\,n, \theta \sim \binom{n}{r} p^r\,(1-p)^{n-r}\,f_p(p|\theta), \tag{51}$$

given n and θ. Integrating (51) over p gives the conditional distribution of R given n and θ as

$$R|n, \theta \sim \binom{n}{r} \int_0^1 p^r\,(1-p)^{n-r}\,f_p(p|\theta)\,dp \tag{52}$$

or, expressed in terms of moments of $f_p(\cdot|\theta)$ (a form which will repeatedly be found in results later),

$$R\,|\,n, \theta \sim \binom{n}{r} \mathbf{E}\left[P^r(1-P)^{n-r}|\theta\right]\,. \tag{53}$$

Note from (52) that mixing over p using this fixed θ, not surprisingly, has the effect that the distribution for the number of failures which will be seen during a given sequence of demands is now *more dispersed* than a corresponding binomial distribution. We can quantify this effect precisely by verifying that from the distribution (52) we have mean

$$\mathbf{E}[R|n, \theta] = n\mathbf{E}[P|\theta]$$

where

$$\mathbf{E}[P|\theta] = \int_0^1 p f_p(p|\theta)\,dp\,,$$

and

$$\mathbf{Var}(R|n, \theta) = n\mathbf{E}[P|\theta]\,(1 - \mathbf{E}[P|\theta]) + n(n-1)\mathbf{Var}(P|\theta)\,.$$

[8]Notice that we keep to the usual notational convention of upper case for a random variable and lower case for a numerical value obtained as a particular realisation.

In this sum, the left-hand term is the variance of a binomial distribution with the same maximum n and mean $n\mathbf{E}[P|\theta]$. As one might expect, the right-hand 'excess' term depends on the variance

$$\mathbf{Var}(P|\theta) = \int_0^1 (p - \mathbf{E}[P|\theta])^2 f_p(p|\theta)\, dp$$

of the mixing distribution $f_p(\cdot|\theta)$.

If we *observe* that $R = r$ failures actually occur during n demands, then we can condition on this data by normalising (51) to give the updated distribution

$$P|r, n, \theta \sim \frac{p^r (1-p)^{n-r} f_p(p|\theta)}{\int_0^1 p^r (1-p)^{n-r} f_p(p|\theta)\, dp} \tag{54}$$

of the probability of failure on demand for this program, given θ, n and the observation r.

The last eight equations describe properties of a general mixture of Bernoulli trials processes [23, pp213-4,217], where $f_p(\cdot|\theta)$ is the mixing distribution. Note that although exchangeability[9] of the original Bernoulli trials process has not been lost by mixing the processes, the property that non-intersecting sections of the process are independently distributed does not hold in general for the resulting mixed process. In fact the number R' of failures in a subsequent set of n' demands from the same sequence now has an updated distribution obtainable from (54) as

$$\begin{aligned} R'|r, n, n', \theta &\sim \binom{n'}{r'} \frac{\int_0^1 p^{r+r'} (1-p)^{n+n'-r-r'} f_p(p|\theta)\, dp}{\int_0^1 p^r (1-p)^{n-r} f_p(p|\theta)\, dp}, \\ &= \binom{n'}{r'} \frac{\mathbf{E}\left[P^{r+r'}(1-P)^{n+n'-r-r'}|\theta\right]}{\mathbf{E}[P^r(1-P)^{n-r}|\theta]} \end{aligned} \tag{55}$$

given n, r.

The distributions which we have considered up till this point are parameterised by θ. Under our chosen model, (55) is not a predictive distribution of future failures given past failure behaviour since it depends on the unknown value of the parameter θ. This deliberate θ-dependence is intended to take account of the practical fact that we are unable with any confidence to accurately state the distribution $f_p(\cdot|\theta)$ of failure probabilities of the ⟨product, environment⟩ pair within our family. This inability is captured in the model as our uncertainty about the ⟨product, environment⟩-family parameter θ. This parameter uncertainty has yet to be expressed and incorporated into the picture. We now adopt a Bayesian approach to handling this dimension of the problem by supposing a prior distribution

$$\Theta \sim \text{Prior}_\theta(\theta)\,,$$

[9]The property that any permutation of a portion of the boolean (success-failure) sequence has the same probability as the unpermuted sequence. Equivalently, we can say that the probability of a precise sequence of successes and failures during a specified interval of discrete time (say from the 10^{th} to the 20^{th} demand, inclusive) can be expressed as a function of the *number*, only, of successes during that interval.

with support set $\theta \in \Theta$. If we plan to observe and predict reliability only of a single software ⟨product, environment⟩, this extension actually adds very little, if anything, useful to the model as so far described, since, by integrating over θ, the model is reduced to a degenerate ($|\Theta| = 1$) case of the assumptions described earlier. (Simply replace $f_p(p|\theta)$ by $\int_{\theta\in\Theta} f_p(p|\theta)\text{Prior}_\theta(\theta)\,d\theta$ in the distributions above.) However, the idea of a prior distribution for θ becomes a useful concept if we wish to address the problem of *learning* about a *distribution* of software reliabilities by observing *multiple* sequences of software failure behaviour from a single family $\langle \mathcal{A}_i \rangle$, say, of ⟨product, environment⟩ pairs. We can then represent a conservative[10] version of a *process* concept for their reliabilities, from one ⟨product, environment⟩ to the next, by modelling these sequences as individual failure processes of the Bernoulli-trials kind discussed above but with *different* p_i, and an assumption that each of these p_i arises *independently given* θ for some *unknown, common* parameter value θ characterising the entire family of ⟨product, environment⟩ pairs. We are then able to learn from observation of the early data sequences about the likely behaviour of another sequence through the medium of our improving knowledge of their common parameter θ.

Thus θ and p now play distinct roles in terms of the model concepts: Whereas each p_i still captures a property of a single software testing sequence, θ now represents a common unknown characteristic of the whole family of such sequences. To obtain the value of θ would be to capture the reliability-relevant characteristic which these software pairs ⟨product, environment⟩ all have in common. For this *multi*-sequence model, there is now a real purpose behind including separate distributional assumptions for firstly θ, and secondly p_i given θ. Below, we do not in fact assume that θ can ever be known[11]. However, we assume that we hold *probabilistic prior beliefs about* θ (i.e. beliefs about the possible distributions $f_p(\cdot|\theta)$ of reliabilities of sequences belonging to the family $\langle \mathcal{A}_i \rangle$). Then, any observation of failure behaviour of any subset of the sequence $\langle \mathcal{A}_i \rangle$ can be regarded as information about θ which we will use in order to learn about θ by the usual Bayesian learning mechanisms. Thus the second stage of our doubly stochastic model is to represent our prior beliefs about a subjective random variable Θ of which the true value θ for our particular family of sequences is a single unknown realisation. Figure 8 depicts these conditional dependence relationships diagramatically. This popular DAG (directed acyclic graph) representation of conditional independence assumptions is equivalent to the assertion that the joint distribution of all the nodes is equal to the product of

[10] in the sense that we refrain from making any stronger assumption of any kind of systematic development of reliability from one sequence to the next. For example, we do not assume an increasing trend in reliabilities of different sequences in the family

[11] Loosely, we can say that in order to *know* the value of θ characterising a family $\langle \mathcal{A}_i \rangle$ of executing software products $\langle \text{product}_i, \text{environment}_i \rangle$, we would require a very large amount of operational failure data on *each* of a very large number of sequences belonging to that family. We could then accurately describe from empirical data the shape of the distribution $f_p(\cdot|\theta)$

the conditional distributions of each node conditioned on the values of its parents. Actually, we have tended to condition on the values $\langle n_i \rangle_{i=1}^k$ throughout our probabilistic analysis so that the n_i-nodes can be thought of as degenerate, constant random variables. Note that we have used a notation for

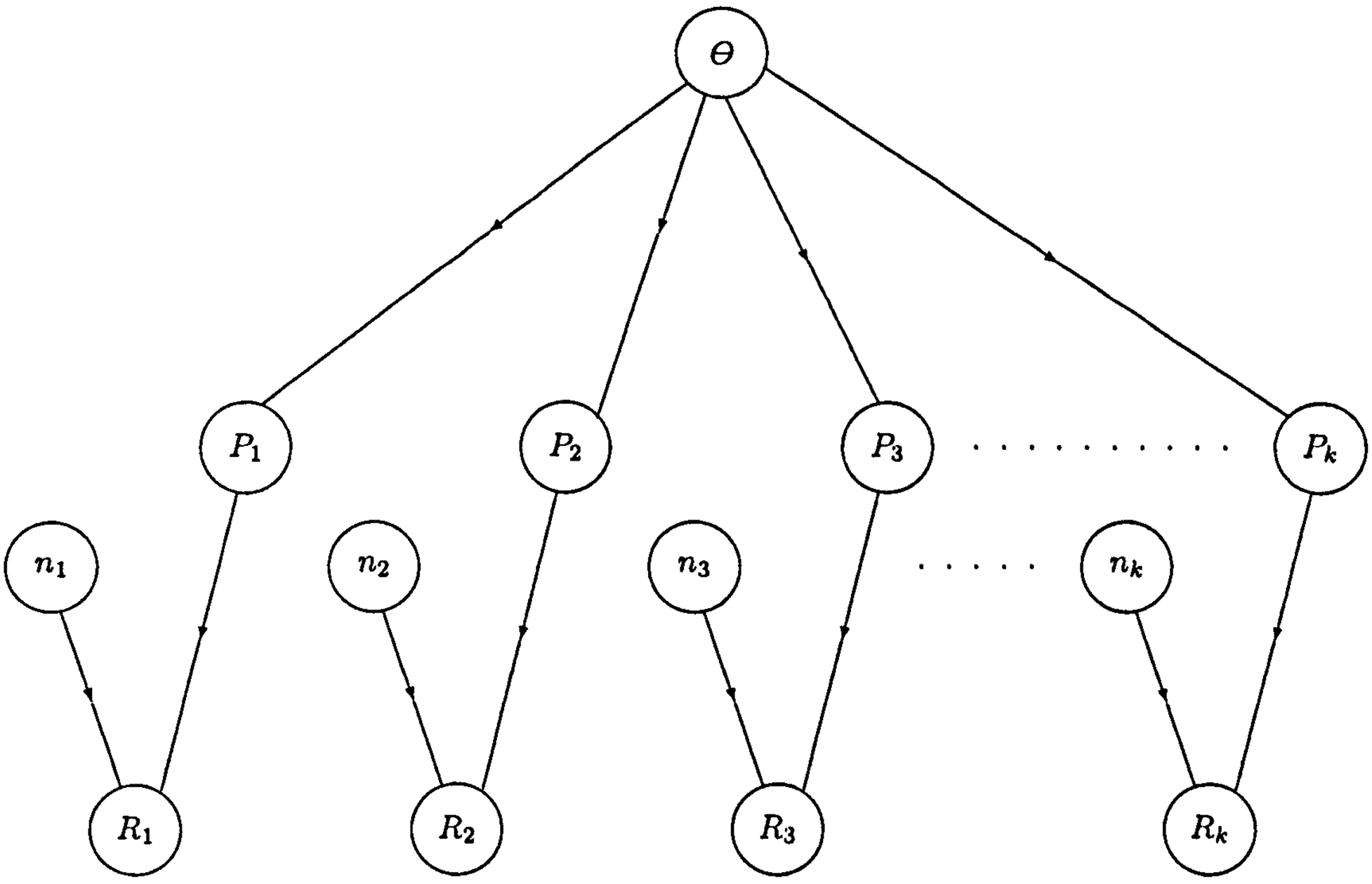


Figure 8: Diagram of the dependencies of the model

our mixtures and marginal distributions which assumes that both the distributions f_p, for P given θ, and the prior distribution, Prior_θ for θ, are continuous. The cases where either or both of these distributions are discrete are also of interest and correspond to the replacement of integrals by sums, or, alternatively, to the use of the Dirac delta function in specifying definitions for our densities f_p and Prior_θ.

Before proceeding to consider in §5.3.5 specific distributional assumptions appropriate for the i.i.d. P_i given Θ, and for the ⟨product, environment⟩ family parameter Θ itself, we obtain, in the following sections, a few consequences of these model assumptions in the general case. Observe firstly that, conditionally given θ and $\langle n_i \rangle_{i=1}^k$, our independence assumption for the $\langle P_i \rangle$ tells us that the first k terms of our $\langle R_i \rangle$ sequence are jointly distributed

$$\langle R_i \rangle_{i=1}^k \big| (\langle n_i \rangle_{i=1}^k, \theta) \sim \prod_{i=1}^k \binom{n_i}{r_i} \int_0^1 p^{r_i} (1-p)^{n_i - r_i} f_p(p|\theta)\, dp\,. \tag{56}$$

Once we are in possession of data in the form of observed failure behaviour of these k software products executing in their k environments (i.e., r_i failures out of n_i trials for each sequence $\mathcal{A}_i$)

then we can regard (56) as the likelihood function $L(\theta; \langle n_i, r_i \rangle_{i=1}^k)$ of the parameter θ given this failure data. $L(\theta; \langle n_i, r_i \rangle_{i=1}^k)$ is a product involving constant[12] combinatorial terms together with moments of the parametric distribution $f_p(\cdot|\theta)$

$$\langle R_i \rangle_{i=1}^k \big| (\langle n_i \rangle_{i=1}^k, \theta) \sim \prod_{i=1}^k \binom{n_i}{r_i} \mathbf{E}[P^{r_i}(1-P)^{n_i - r_i} | \theta] \ . \tag{57}$$

We find, not surprisingly, in the following sections that, using this Bayesian model, our reliability predictions turn out to depend heavily on our *prior beliefs*, and not only on the empirical reliability data $\langle r_i \rangle_{i=1}^k$ which is later observed. We have expressed the shape of these beliefs formally by our selection of the distributions $\langle \{ f_p(\cdot|\theta) \, ; \, \theta \in \Theta \}, \text{Prior}_\theta \rangle$ comprising our model for the failure probabilities $\langle P_i \rangle$ of our family $\langle \mathcal{A}_i \rangle$ of sequences of Bernoulli trials. There are several ways of understanding the entity $\langle \{ f_p(\cdot|\theta) \, ; \, \theta \in \Theta \}, \text{Prior}_\theta \rangle$ less formally, which may help with the selection of appropriate distributions in the case of a particular family $\langle \mathcal{A}_i \rangle$. To begin with, $\langle \{ f_p(\cdot|\theta) \, ; \, \theta \in \Theta \}, \text{Prior}_\theta \rangle$ should contain at least

1. Our best guess, prior to observation, of the *family average reliability level* of the $\langle \mathcal{A}_i \rangle$,

i.e. the average reliability towards which our beliefs would hypothetically converge if we could acquire arbitrarily large amounts of data from each of arbitrarily many distinct $\langle \text{product}, \text{environment} \rangle$ pairs which were representative of this family. But $\langle \{ f_p(\cdot|\theta) \, ; \, \theta \in \Theta \}, \text{Prior}_\theta \rangle$ is much more than a complicated way of expressing a guess at the family average reliability. We emphasise that it contains at least two other dimensions of expressed prior belief, each of which should be verified against intuition, and against any available objective prior knowledge, if this model is to be applied. $\langle \{ f_p(\cdot|\theta) \, ; \, \theta \in \Theta \}, \text{Prior}_\theta \rangle$ also contains

2. Our prior beliefs about the *shape of the distribution of true reliabilities* (as distributed around the average value assessed in 1.) of this $\langle \mathcal{A}_i \rangle$ family. How *consistent* will the reliabilities governing success and failure in the testing sequences $\mathcal{A}_i$ of our family eventually be found to be?

Lastly, but of no lesser significance in terms of the reliability predictions emanating from our approach, we recall that the entity $\langle \{ f_p(\cdot|\theta) \, ; \, \theta \in \Theta \}, \text{Prior}_\theta \rangle$ incorporates a Bayesian subjective parameter distribution Prior_θ. Through this, the construction $\langle \{ f_p(\cdot|\theta) \, ; \, \theta \in \Theta \}, \text{Prior}_\theta \rangle$ pays due regard to our stated measure of

3. Our *confidence* (or lack of it) in own our *ability to produce accurate a priori guesses* at 1. and 2.

How confident are we that both of these initial assessments are close to the truths that would

[12] i.e. not depending on θ

ultimately be discovered given unlimited data from an unlimited number of representative boolean valued sequences $\mathcal{A}_i$?

This third component of prior belief is a classic Bayesian subjective prior distribution describing our uncertainty about a model feature which in this case is effectively an entire continuous probability distribution on the unit interval, and whose unknown true value characterises our whole family of ⟨product, environment⟩ pairs. This is perhaps also the component whose effects on the subsequent analysis are the most easy to overlook—or at least whose effects in our analysis can be the most difficult to follow intuitively. To simplify for the sake of illustration, suppose we make θ a one-dimensional real quantity (so that $\Theta \subseteq \mathbb{R}$), and suppose that we happen to have used a parameterisation of our family of $f_p(\cdot|\theta)$ distributions that orders these distributions according to their means. Then, holding this parameterisation $\{f_p(\cdot|\theta)\,;\,\theta\in\Theta\}$ fixed, the act of choosing a relatively more dispersed distribution Prior_θ will correspond to a statement of relatively lower confidence in our ability to accurately guess the value of the family average of reliabilities (item 1. above). Similarly, if we adopt, say, the coefficient of variation of the unknown true distribution from which the P_i are drawn as a numerical representation of an important attribute of the shape (item 2. above) of this distribution of failure probabilities P_i around this mean value, and if we assume, again for simplicity, that our chosen θ-parameterisation now orders the distributions $f_p(\cdot|\theta)$ instead according to their coefficients of variation rather than their means, then the choice of a relatively more dispersed Prior_θ will represent our relatively lower confidence in our ability to assess, a priori, the true amount of consistency amongst the different reliabilities of the members of our family $\langle\mathcal{A}_i\rangle$. So in the case of such a parameterisation we could ask ourselves whether we already possess a thorough understanding of reliability variability within this kind of ⟨product, environment⟩ family. If so, then a highly concentrated Prior_θ distribution would be an appropriate choice. If instead we considered within-family reliability variation between the $\mathcal{A}_i$ to be rather difficult to assess, without spending some time accumulating an operational history of a number of sequences from the family concerned, then we should choose a larger spread for Prior_θ : and by so doing admit a greater variety of distributions $f_p(\cdot|\theta)$ on the unit interval which could each plausibly represent the true nature of the variation of failure probabilities between the sequences of our family.

Of course, we do not have to use the exact 3 items defined above in order to informally decompose the structure $\langle\{f_p(\cdot|\theta)\,;\,\theta\in\Theta\}, \text{Prior}_\theta\rangle$ of our prior belief into a number of salient features whose effects will transmit themselves through the mathematical analysis of this model. The important point is that we must be aware of the profound implications—for the reliability predictions obtained in the following sections—that each one of these components of our prior belief model has. To

summarise the last paragraph, our 'similar products' model proposes a formal representation of prior beliefs about a family of ⟨product, environment⟩ pairs between which we are initially indifferent. This representation $\langle \{f_p(\cdot|\theta)\,;\,\theta\in\Theta\}, \text{Prior}_\theta \rangle$ is expressive enough to allow us—in fact it requires us—to state with precision *how much we know* (and often the extent to which we are in fact ignorant) about the average *level* and *the distribution* of the achieved reliabilities of the members of this family. Given an available amount $\langle k, n_1, n_2, \ldots, n_k \rangle$ of testing, we go on in the following sections to show how this model of prior belief, combines with empirical testing data $\langle r_1, r_2, \ldots, r_k \rangle$ to yield predictions of future reliability of an individual success-failure sequence within the family. An important question running through the analysis of this model is the amount of improvement in our ability to assess high reliabilities that is achieved by incorporating data on other sequences within the family. It is of interest to examine formally the dependence of the answer to this question on the strength of our prior beliefs—particularly our prior beliefs about reliability consistency—concerning the family $\langle A_i \rangle$.

5.3.2 Bayesian Updating of Distributions and Moments in the General Case

To implement the Bayesian learning about Θ given observation of $\langle r_i \rangle_{i=1}^k$ we need to calculate the posterior distribution of Θ

$$\Theta \big| \langle n_i, r_i \rangle_{i=1}^k \sim cL\big(\theta; \langle n_i, r_i \rangle\big)\, \text{Prior}_\theta(\theta)$$

where c is a function of $\langle r_i, n_i \rangle_{i=1}^k$ not involving θ, i.e.

$$\Theta \big| \langle n_i, r_i \rangle_{i=1}^k \sim \frac{\left[\prod_{i=1}^k \int_0^1 p^{r_i}\,(1-p)^{n_i-r_i}\, f_p(p|\theta)\,dp\right] \text{Prior}_\theta(\theta)}{\int_{\theta\in\Theta} \left[\prod_{i=1}^k \int_0^1 p^{r_i}\,(1-p)^{n_i-r_i}\, f_p(p|\theta)\,dp\right] \text{Prior}_\theta(\theta)\,d\theta} \tag{58}$$

Equation (58) draws the focus of attention away from failure probabilities P_i of sequences $\mathcal{A}_i$ by the integrations over p. But it is now of great practical interest to know an up-to-date distribution for P given what has been observed (in order to make predictions about a particular new (product, environment), for example). Then our learning could be expressed directly in terms of the changing nature of the current uncertainty about a failure probability of some particular sequence. At this stage it is instructive to distinguish three different circumstances under which we will have learned, in different ways, about one of the failure probabilities, say P_k. These three different circumstances will each result in an up-to-date Bayesian posterior distribution for this failure probability,

which may be compared with the prior marginal distribution of P_k

$$P_k \sim \int_{\theta\in\Theta} f_p(p_k|\theta)\mathrm{Prior}_\theta(\theta)\,d\theta\,, \tag{59}$$

which represents our initial state of uncertainty concerning the reliability of any given sequence, $\mathcal{A}_k$, prior to any observation either of that or of any other (product, environment) pair's behaviour. At this point of no observation, (59) is the mixing distribution associated with our mixture-of-Bernoulli-trials model for future failure of $\mathcal{A}_k$. This comparison of (59) with subsequent updated P_k-distributions determines the nature and limits of what we can learn from observed failure behaviour alone, be it of a single sequence or of a number of sequences from a particular (product, environment) family.

Firstly the most trivial case—observing only the past failure behaviour of the specific (product, environment) pair of interest—has effectively already been covered by (54). Substituting $\int_{\theta\in\Theta} f_p(p|\theta)\mathrm{Prior}_\theta(\theta)\,d\theta$ for $f_p(p|\theta)$ in (54) gives a conditional distribution

$$P_k\,|\,n_k, r_k \sim \frac{p_k^{r_k}(1-p_k)^{n_k-r_k}\int_{\theta\in\Theta} f_p(p_k|\theta)\mathrm{Prior}_\theta(\theta)\,d\theta}{\int_{\theta\in\Theta}\int_0^1 p^{r_k}(1-p)^{n_k-r_k}\,f_p(p|\theta)\,dp\,\mathrm{Prior}_\theta(\theta)\,d\theta} \tag{60}$$

for P_k given n_k and r_k. Note that we will continually assume, as we have done in the denominator here, that the families of densities chosen are such that changes of the order of integration are legitimate.

Secondly replacing k by $k-1$ in (58) and then substituting this distribution in place of $\mathrm{Prior}_\theta(\theta)$ in (59) (or, alternatively, directly substituting $n_k = r_k = 0$ in (62) below) gives the distribution

$$P_k\,\big|\,\langle n_i, r_i\rangle_{i=1}^{k-1} \sim \frac{\int_{\theta\in\Theta} f_p(p_k|\theta)\left[\prod_{i=1}^{k-1}\int_0^1 p^{r_i}(1-p)^{n_i-r_i}\,f_p(p|\theta)\,dp\right]\mathrm{Prior}_\theta(\theta)\,d\theta}{\int_{\theta\in\Theta}\left[\prod_{i=1}^{k-1}\int_0^1 p^{r_i}(1-p)^{n_i-r_i}\,f_p(p|\theta)\,dp\right]\mathrm{Prior}_\theta(\theta)\,d\theta} \tag{61}$$

of P_k given observation of the failure behaviour $\langle n_i, r_i\rangle_{i=1}^{k-1}$ *only* of *other* sequences $\langle\mathcal{A}_i\rangle_{i=1}^{k-1}$.

Thirdly, replacing k by $k-1$ in (58) and then substituting this distribution in place of $\mathrm{Prior}_\theta(\theta)$ in (60) gives the distribution

$$P_k\,\big|\,\langle n_i, r_i\rangle_{i=1}^{k} \sim \frac{p_k^{r_k}(1-p_k)^{n_k-r_k}\int_{\theta\in\Theta} f_p(p_k|\theta)\left[\prod_{i=1}^{k-1}\int_0^1 p^{r_i}(1-p)^{n_i-r_i}\,f_p(p|\theta)\,dp\right]\mathrm{Prior}_\theta(\theta)\,d\theta}{\int_{\theta\in\Theta}\left[\prod_{i=1}^{k}\int_0^1 p^{r_i}(1-p)^{n_i-r_i}\,f_p(p|\theta)\,dp\right]\mathrm{Prior}_\theta(\theta)\,d\theta} \tag{62}$$

for P_k given observation both of the failure behaviour $\langle n_k, r_k \rangle$ of the sequence $\mathcal{A}_k$ itself and *also* of the failures $\langle n_i, r_i \rangle_{i=1}^{k-1}$ of other sequences $\langle \mathcal{A}_i \rangle_{i=1}^{k-1}$.

We remark here that the effects of observation respectively of past and of the present software $\langle$product, environment$\rangle$ pair's failure behaviour on our beliefs about the present pair's per-demand failure probability P_k appear to obey a simple multiplicative property. Comparing the *numerators* of the four different probability densities of P_k given by equations (59–62) we see that these are in common proportions to each other[13]. The denominators appear to spoil these relationships, but the denominators are only normalising constants, i.e. they do not depend on P_k. We can use this fact to express the property more concisely in terms of the effect of the different observations on the extent to which we favour one value, say p'_k, of P_k over another, say p''_k. If, in this way, we compare the values of the densities at this arbitrary pair of P_k values, we see that

$$\frac{\mathrm{pdf}\left(p'_k \middle| \langle n_i, r_i \rangle_{i=1}^{k}\right)}{\mathrm{pdf}\left(p''_k \middle| \langle n_i, r_i \rangle_{i=1}^{k}\right)} \cdot \frac{\mathrm{pdf}\,(p'_k)}{\mathrm{pdf}\,(p''_k)} = \frac{\mathrm{pdf}\,(p'_k | n_k, r_k)}{\mathrm{pdf}\,(p''_k | n_k, r_k)} \cdot \frac{\mathrm{pdf}\left(p'_k \middle| \langle n_i, r_i \rangle_{i=1}^{k-1}\right)}{\mathrm{pdf}\left(p''_k \middle| \langle n_i, r_i \rangle_{i=1}^{k-1}\right)} \tag{63}$$

provided, of course, that p''_k is not a point of zero density for any of these four densities. Equation (63) perhaps becomes more intuitively meaningful if converted to the form

$$\frac{\mathrm{pdf}\left(p'_k \middle| \langle n_i, r_i \rangle_{i=1}^{k}\right)}{\mathrm{pdf}\left(p''_k \middle| \langle n_i, r_i \rangle_{i=1}^{k}\right)} \Big/ \frac{\mathrm{pdf}\,(p'_k)}{\mathrm{pdf}\,(p''_k)} = \left\{\frac{\mathrm{pdf}\,(p'_k | n_k, r_k)}{\mathrm{pdf}\,(p''_k | n_k, r_k)} \Big/ \frac{\mathrm{pdf}\,(p'_k)}{\mathrm{pdf}\,(p''_k)}\right\} \cdot \left\{\frac{\mathrm{pdf}\left(p'_k \middle| \langle n_i, r_i \rangle_{i=1}^{k-1}\right)}{\mathrm{pdf}\left(p''_k \middle| \langle n_i, r_i \rangle_{i=1}^{k-1}\right)} \Big/ \frac{\mathrm{pdf}\,(p'_k)}{\mathrm{pdf}\,(p''_k)}\right\}. \tag{64}$$

In practical terms this says that observation of *both the present and previous sequences* changes the 'odds of $P_k = p'_k$ vs. $P_k = p''_k$' by a factor which is the product of the corresponding changes in odds resulting from observing, respectively, *only the present* sequence, or *only previous* sequences[14]. The same property is alternatively captured by the formula

$$P_k \left| \langle n_i, r_i \rangle_{i=1}^{k} \sim \frac{\mathrm{pdf}\,(p_k | n_k, r_k) \cdot \mathrm{pdf}\left(p_k \middle| \langle n_i, r_i \rangle_{i=1}^{k-1}\right) / \mathrm{pdf}\,(p_k)}{\int_0^1 \mathrm{pdf}\,(p_k | n_k, r_k) \cdot \mathrm{pdf}\left(p_k \middle| \langle n_i, r_i \rangle_{i=1}^{k-1}\right) / \mathrm{pdf}\,(p_k)\; dp_k} \right. \tag{65}$$

(defining this density to be zero wherever $\mathrm{pdf}\,(p_k)$ is zero).

On closer examination the model property captured by (63) and (64) is found to be merely an instance of a quite general result in Bayesian statistical modelling which applies wherever the construction of a probability model makes two observables, Y_1 and Y_2, *conditionally* independent *given* the value of some model parameter ξ. Then if we ask the question: 'How do the Bayesian

[13] The algebraic product, as functions of P_k, of the 'most informed' (62) and the 'least informed' (59) is equal to the product of the other two (60) and (61) arising from the intermediate levels of information

[14] Of course, it is likely that for some pairs $\langle p'_k, p''_k \rangle$ the two terms in curly braces may not be on the same side of unity, so that, for such pairs, when both sources of data are observed, a kind of cancellation will occur between the tendency of each kind of data separately to cause us to prefer p'_k over p''_k, or vice versa.

updated distributions of ξ for the three possible cases relative to observation or non-observation of Y_1 and Y_2 compare with the prior distribution of ξ?', we obtain an answer of the above form. Our model is clearly of this kind for $\xi = p_k$, $Y_1 = \langle R_i \rangle_{i=1}^{k-1}$, and $Y_2 = R_k$. Note that no similar proportionate relationship holds when we consider updated *reliability predictions*, rather than updated distributions of the per demand failure probability P_k of the current sequence $\mathcal{A}_k$. Below, on p128, we compare the effects of these same four 'states of observation' on explicit *reliability predictions* (equations (78–81)). Nor is it possible to further factorise the right-most term in equations (63,64) since under our model we do not have the required conditional independence, given the value of P_k for the *current* observation sequence $\mathcal{A}_k$, of observations on *distinct previous* $\mathcal{A}_i$, $i < k$.

We remark that the approach used to obtain (62) is not limited to providing us with the updated univariate distribution of a single sequence's failure probability. The updated joint distribution of say $\langle P_{k-1}, P_k \rangle$ can be obtained from a bivariate form of the arguments. Since (60) and (61) are actually just special cases of (62) for certain of the n_i set equal to zero, we will not go through the extension to the bivariate case separately for each of the three observation cases distinguished in (60-62). In the general case where the observations take the form $\langle n_i, r_i \rangle_{i=1}^{k}$, which includes the three cases previously distinguished, we obtain an updated joint distribution

$$(P_{k-1}, P_k) \,\big|\, \langle n_i, r_i \rangle_{i=1}^{k} \sim \frac{p_{k-1}^{r_{k-1}}(1-p_{k-1})^{n_{k-1}-r_{k-1}} p_k^{r_k}(1-p_k)^{n_k-r_k} \int_{\theta\in\Theta} f_p(p_{k-1}|\theta) f_p(p_k|\theta) \left[\prod_{i=1}^{k-2} \int_0^1 p^{r_i}(1-p)^{n_i-r_i} f_p(p|\theta)\, dp\right] \mathrm{Prior}_\theta(\theta)\, d\theta}{\int_{\theta\in\Theta} \left[\prod_{i=1}^{k} \int_0^1 p^{r_i}(1-p)^{n_i-r_i} f_p(p|\theta)\, dp\right] \mathrm{Prior}_\theta(\theta)\, d\theta} \tag{66}$$

from which we can, if we wish, investigate the sign and magnitude of any correlation between P_{k-1} and P_k (or powers of these) conditional on the observed data. For higher dimensional joint posterior distributions of the $\langle P_i \rangle$, (66) extends in the way you would expect, to give for the k-dimensional joint distribution

$$\langle P_i \rangle_{i=1}^{k} \,\big|\, \langle n_i, r_i \rangle_{i=1}^{k} \sim \frac{\left\{\prod_{i=1}^{k} p_i^{r_i}(1-p_i)^{n_i-r_i}\right\} \int_{\theta\in\Theta} \left[\prod_{i=1}^{k} f_p(p_i|\theta)\right] \mathrm{Prior}_\theta(\theta)\, d\theta}{\int_{\theta\in\Theta} \left[\prod_{i=1}^{k} \int_0^1 p^{r_i}(1-p)^{n_i-r_i} f_p(p|\theta)\, dp\right] \mathrm{Prior}_\theta(\theta)\, d\theta}$$

We will not pursue this investigation further, concentrating instead on the updated distribution of the univariate P_k, and its consequences for reliability predictions of the single sequence $\mathcal{A}_k$ given the various combinations of observations discussed above.

Depending on the choice of the distribution family $\{f_p(\cdot|\theta)\,;\, \theta\in\Theta\}$ and of the distribution Prior_θ, we may anticipate some analytic and computational difficulties in obtaining these updated distributions for P_k. However, we can perhaps more easily obtain expressions for the effect of the learning

on the *moments* of the distribution of P_k. In fact the moments of these three alternative updated P_k-distributions (which will play the role of mixing distribution in Bayesian reliability prediction for $\mathcal{A}_k$) are important since any probability prediction of future failures of $\mathcal{A}_k$ is equivalent to the expectation, with respect to one of these updated P_k-distributions, of the equivalent prediction conditioned on P_k; and the latter conditional probability will generally involve *positive integer powers of* P_k. (See e.g. (50).) This follows because our model assumptions tell us that the three quantities *past failure behaviour of* $\mathcal{A}_k$, *future failure behaviour of* $\mathcal{A}_k$, *failure behaviour of other sequences* are conditionally independent given P_k. For example, the predictive probability of $R{=}r$ failures in n further demands on $\mathcal{A}_k$ is obtained by substituting the appropriate one of (60), (61), or (62) for $f_p(p|\theta)$ in (52).

More generally if:–

1. the term *observations* refers to some partial or complete joint observation of past failure behaviours of $\mathcal{A}_k$ and of other sequences $\mathcal{A}_i$; and

2. the term *future failure behaviour of* $\mathcal{A}_k$ refers to some pre-specified *event* concerning the pattern of future failure of sequence $\mathcal{A}_k$;

then we have

$$\mathbf{P}[\textit{future failure behaviour of } \mathcal{A}_k \,|\, \textit{observations}] = \mathbf{E}\Big[\, \mathbf{P}[\textit{future failure behaviour of } \mathcal{A}_k \,|\, P_k] \;\Big|\; \textit{observations} \,\Big] \qquad (67)$$

where, on the right-hand side, the value of the inner probability will be a function of P_k (calculated as for an ordinary Bernoulli trials process) and where the outer expectation is *calculated with respect to the updated distribution of* P_k given '*observations*', which distribution will be one of equations (60), (61), and (62) when '*observations*' is of one of the three specific kinds we have discussed explicitly.

For an alternative perspective on the same predictions we remark that we are not *obliged* to think of them in terms of the updated distributions (60–62) of P_k. We can instead use the doubly stochastic structure of our model and its two layers of conditional independence[15] assumptions to show that a prediction of the form (67) will in fact assume a ratio form which can be understood directly in terms of two layers of nesting of probabilities and expectations with respect to our initial model distributions. In fact, our independence assumptions tell us that whenever '*observations*' is of such a 'product' form that we can decompose it into $\bigwedge_{i=1}^{k}(\textit{past behaviour of } \mathcal{A}_i)$ (i.e. if it is

[15] of $\langle P_i \rangle$ given θ for the sequence family, and of success/failure on separate demands given P_i for a particular sequence $\mathcal{A}_i$

actually a conjunction of separate events concerning each $\mathcal{A}_i$ in isolation) then (67) can be shown to be equivalent to the formula

$$\begin{aligned}
&\mathbf{P}[\textit{future failure behaviour of } \mathcal{A}_k \,|\, \textit{observations}] \\
&= \frac{\mathbf{E}\Big[\mathbf{P}[\textit{future} \text{ and } \textit{past failure behaviour of} \mathcal{A}_k \,|\, \theta] \prod_{i=1}^{k-1} \mathbf{P}[\textit{past failure behaviour of} \mathcal{A}_i \,|\, \theta]\Big]}{\mathbf{E}\Big[\prod_{i=1}^{k} \mathbf{P}[\textit{past failure behaviour of} \mathcal{A}_i \,|\, \theta]\Big]} \\
&= \frac{\mathbf{E}\Big[\,\mathbf{E}\Big(\mathbf{P}[\textit{future f. b. of } \mathcal{A}_k \,|\, P_k]\,\mathbf{P}[\textit{past f. b. of } \mathcal{A}_k \,|\, P_k] \,\Big|\, \theta\Big) \prod_{i=1}^{k-1} \mathbf{E}\Big(\mathbf{P}[\textit{past f. b. of } \mathcal{A}_i \,|\, P_i] \,\Big|\, \theta\Big)\Big]}{\mathbf{E}\Big[\prod_{i=1}^{k} \mathbf{E}\Big(\mathbf{P}[\textit{past f. b. of } \mathcal{A}_i \,|\, P_i] \,\Big|\, \theta\Big)\Big]}
\end{aligned} \tag{68}$$

where this last form is a prediction expressed as a ratio directly in terms of the distributions used to construct the model. In both numerator and denominator the inner probabilities are calculated as for a Bernoulli trials process, the inner expectations are obtained using the distribution $f_p(\,\cdot\,|\theta)$, and the outer expectations are taken with respect to our prior distribution Prior_θ.

Before making any observations, P_k has a marginal distribution whose m^{th} non-central moment is given by

$$\mathbf{E}[P_k^m] = \int_{\theta\in\Theta} \int_0^1 p^m f_p(p|\theta)\, dp\, \text{Prior}_\theta(\theta)\, d\theta = \int_{\theta\in\Theta} \mathbf{E}[P_k^m \,|\, \theta]\, \text{Prior}_\theta(\theta)\, d\theta \tag{69}$$

This moment of P_k is updated, by our three distinguished observation assumptions, to give expressions for the moments of the distributions (60), (61), and (62) which take the general form of ratios of expectations with respect to Prior_θ of multinomials in the moments of f_p (which moments are of course functions of Θ). This is a consequence of the fact that (60), (61), and (62) are simple linear transforms of this θ-parameterised p.d.f. of our assumed conditional distribution for P_k given Θ. (Or it can also be explained as a particular case of (68).) Specifically, taking the three observation cases in the same order as earlier, the m^{th} *updated* non-central moment of P_k is

$$\mathbf{E}[P_k^m \,|\, n_k, r_k] = \frac{\int_{\theta\in\Theta} \mathbf{E}\Big[P^{m+r_k}(1-P)^{n_k-r_k} \,\Big|\, \theta\Big]\, \text{Prior}_\theta(\theta)\, d\theta}{\int_{\theta\in\Theta} \mathbf{E}\Big[P^{r_k}(1-P)^{n_k-r_k} \,\Big|\, \theta\Big]\, \text{Prior}_\theta(\theta)\, d\theta}, \tag{70}$$

or

$$\mathbf{E}\Big[P_k^m \,\Big|\, \langle n_i, r_i\rangle_{i=1}^{k-1}\Big] = \frac{\int_{\theta\in\Theta} \mathbf{E}[P^m|\theta] \Big[\prod_{i=1}^{k-1} \mathbf{E}\Big[P^{r_i}(1-P)^{n_i-r_i} \,\Big|\, \theta\Big]\Big]\, \text{Prior}_\theta(\theta)\, d\theta}{\int_{\theta\in\Theta} \Big[\prod_{i=1}^{k-1} \mathbf{E}\Big[P^{r_i}(1-P)^{n_i-r_i} \,\Big|\, \theta\Big]\Big]\, \text{Prior}_\theta(\theta)\, d\theta}, \tag{71}$$

or

$$\mathbf{E}\Big[P_k^m \,\Big|\, \langle n_i, r_i\rangle_{i=1}^{k}\Big] = \frac{\displaystyle\int_{\theta\in\Theta} \mathbf{E}\Big[P^{m+r_k}(1-P)^{n_k-r_k}\,\Big|\,\theta\Big]\left[\prod_{i=1}^{k-1}\mathbf{E}\Big[P^{r_i}(1-P)^{n_i-r_i}\,\Big|\,\theta\Big]\right]\mathrm{Prior}_\theta(\theta)\,d\theta}{\displaystyle\int_{\theta\in\Theta}\left[\prod_{i=1}^{k}\mathbf{E}\Big[P^{r_i}(1-P)^{n_i-r_i}\,\Big|\,\theta\Big]\right]\mathrm{Prior}_\theta(\theta)\,d\theta}, \tag{72}$$

respectively, under the three different assumptions: observation of $\mathcal{A}_k$ only; observation only of other sequences $\langle\mathcal{A}_i\rangle_{i=1}^{k-1}$; or observation of all of $\langle\mathcal{A}_i\rangle_{i=1}^{k}$. Note that here the left hand sides are updated expectations conditioned on observed data: The right-hand sides are *ratios* of *unconditional* expectations taken with respect to the original, prior Θ-distribution Prior_θ. The random variables whose unconditional expectations form these ratios are 'binomial-like' expressions in the moments of the distribution $f_p(\cdot|\theta)$, which, being deterministic functions of Θ, inherit their distributions from our chosen Prior_θ distribution. To emphasise this role played by these moments of P_k given θ, and at the same time to shorten equations (69–72) slightly, if we define

$$\mu_{r,s}(\theta) = \int_0^1 p^r(1-p)^s f_p(p|\theta)\,dp, \tag{73}$$

then we can write

$$\mathbf{E}[P_k^m] = \mathbf{E}[\mu_{m,0}] \tag{74}$$

$$\mathbf{E}[P_k^m\,|\,n_k, r_k] = \frac{\mathbf{E}[\mu_{m+r_k,n_k-r_k}]}{\mathbf{E}[\mu_{r_k,n_k-r_k}]}, \tag{75}$$

$$\mathbf{E}\Big[P_k^m\,\Big|\,\langle n_i, r_i\rangle_{i=1}^{k-1}\Big] = \frac{\mathbf{E}\left[\mu_{m,0}\displaystyle\prod_{i=1}^{k-1}\mu_{r_i,n_i-r_i}\right]}{\mathbf{E}\left[\displaystyle\prod_{i=1}^{k-1}\mu_{r_i,n_i-r_i}\right]}, \tag{76}$$

$$\mathbf{E}\Big[P_k^m\,\Big|\,\langle n_i, r_i\rangle_{i=1}^{k}\Big] = \frac{\mathbf{E}\left[\mu_{m+r_k,n_k-r_k}\displaystyle\prod_{i=1}^{k-1}\mu_{r_i,n_i-r_i}\right]}{\mathbf{E}\left[\displaystyle\prod_{i=1}^{k}\mu_{r_i,n_i-r_i}\right]}. \tag{77}$$

Equations (67) and (68) tell us that up-to-date reliability predictions may similarly be expressed as ratios of expectations of moments of $f_p(\cdot|\theta)$. Firstly, given no observation data at all, we have

$$\mathbf{P}[r_k|n_k] = \binom{n_k}{r_k}\mathbf{E}[\mu_{r_k,n_k-r_k}] \tag{78}$$

and, once having observed (only) that $R_k = r_k$, if r'_k is the number of failures predicted in a further n'_k demands on sequence $\mathcal{A}_k$,

$$\mathbf{P}[r'_k\,|\,n'_k, n_k, r_k] = \binom{n'_k}{r'_k}\frac{\mathbf{E}\left[\mu_{r_k+r'_k,n_k+n'_k-r_k-r'_k}\right]}{\mathbf{E}[\mu_{r_k,n_k-r_k}]}, \tag{79}$$

For our other two observation assumptions we can write

$$\mathbf{P}\left[r_k \middle| n_k, \langle n_i, r_i\rangle_{i=1}^{k-1}\right] = \binom{n_k}{r_k} \frac{\mathbf{E}\left[\prod_{i=1}^{k} \mu_{r_i, n_i - r_i}\right]}{\mathbf{E}\left[\prod_{i=1}^{k-1} \mu_{r_i, n_i - r_i}\right]}, \tag{80}$$

$$\mathbf{P}\left[r'_k \middle| n'_k, \langle n_i, r_i\rangle_{i=1}^{k}\right] = \binom{n'_k}{r'_k} \frac{\mathbf{E}\left[\mu_{r_k + r'_k, n_k + n'_k - r_k - r'_k} \prod_{i=1}^{k-1} \mu_{r_i, n_i - r_i}\right]}{\mathbf{E}\left[\prod_{i=1}^{k} \mu_{r_i, n_i - r_i}\right]}. \tag{81}$$

(Note that the updated P_k-moments (74–77) are merely a special case of this prediction : the probability that the next m demands result in a string of m successive system failures.) In equation (78–81), the expectations occurring within the right-hand sides are taken with respect to the prior Prior_θ. So the conditioning observations are present in the right-hand expressions only through the specification of *which* moment-terms $\mu_{r,s}(\theta)$ comprise the products whose prior expectation is to be taken. Indeed, it may be useful to think of the distribution $f_p(\cdot|\theta)$, given a θ value, as represented by an infinite, 2-dimensional matrix of its moments $\mu_{r,s}(\theta)$. Then our choice of Prior_θ can be viewed as a distribution over these matrices. Our future reliability predictions will be expressed as product-expectations (over Θ) of certain elements from these matrices, where these elements are selected from the matrix at positions determined by the values of the failure counts we have observed in the past and by the precise future failure-count value whose predictive probability we wish to obtain.

5.3.3 An Upper Bound on Reliability Prediction : The Case of No Observed Failures

Consider the special case in which no failures at all have been observed—neither failures of the ⟨product, environment⟩ pair for which we specifically wish to predict reliability, nor failures of other pairs ⟨product, environment⟩ within the same family. This case may have importance as an upper limit for the reliability levels which can be objectively measured in a given amount of observation time purely from observation of failure behaviour of sequences within the family. Specialising the equations above to this case is simply a matter of substituting the observation $\langle r_i\rangle = \langle 0\rangle$. If we similarly specialise the form of our *predictions* by considering the Bayesian predictive probability of a *further* period of failure-free operation, we find that these predictions can be expressed in rather a simple form as ratios of expectations of products of the non-central moments[16] of $1-P$, with

[16] i.e. moments of the probability of successful completion of an individual demand

P coming from the distribution $f_p(\cdot|\theta)$. So, conclusions about the best reliability levels potentially measurable using this model can be thought of as dependent exclusively[17] on our decision about what may be considered realistic assumptions for our subjective prior distribution of the moment-vector

$$\langle \mu_{0,1}, \mu_{0,2}, \mu_{0,3}, \ldots \rangle = \langle \mathbf{E}[1-P|\theta], \mathbf{E}[(1-P)^2|\theta], \mathbf{E}[(1-P)^3|\theta], \ldots \rangle \tag{82}$$

of the ⟨product, environment⟩ family.

Assuming that we do begin by believing that our family is highly reliable (to be more exact, that any individual ⟨product, environment⟩ pair within the family is highly likely to be highly reliable), then the conditional distribution of P given θ will be concentrated very close to 0 (for all except, perhaps, some values of the family parameter θ which we consider to be very unlikely, i.e. that are assigned small probability (density) values $\text{Prior}_\theta(\theta)$ by our prior for θ). Suppose a particular $f_p(\cdot|\theta)$, i.e. a particular value of the parameter θ, were highly reliable. This θ might correspond to say a particularly good design process, or perhaps a single product which is successful in achieving high operating reliability in a number of different operating environments. Then the first few at least of these moments $\mu_{0,i}$ ought to be very close to 1. But it now appears that it is the relative amounts by which we, at the outset, stochastically believe the higher moments are less than 1, and certain kinds of *correlations* in our beliefs about these moments (as functions of Θ) which determines how much our confidence in failure-free operation for $\mathcal{A}_k$ should grow when we observe failure-free operation of other sequences $\langle \mathcal{A}_i \rangle_{i=1}^{k-1}$. To understand precisely the sense in which the last statement is true substitute $\langle r_i \rangle = \langle 0 \rangle$ and $r'_k = 0$ in equations (78-81). This yields three expressions for the reliability function, i.e. the Bayesian predictive probability that the next m demands on $\mathcal{A}_k$ will be failure-free, given previous observation of failure-free execution of respectively: $\mathcal{A}_k$ only; $\langle \mathcal{A}_i \rangle_{i=1}^{k-1}$; or, lastly, all of $\langle \mathcal{A}_i \rangle_{i=1}^{k}$. These three alternative predictive probabilities of future consecutive successful demands on $\mathcal{A}_k$ should be compared with the unconditional

$$\mathbf{E}[(1-P)^m] = \mathbf{E}[\mu_{0,m}] \,, \tag{83}$$

the probability that the next m demands on $\mathcal{A}_k$ will be failure-free given *no* conditioning observation of either $\mathcal{A}_k$ or any other sequences. Indeed it is the comparison of (78) with (80), and the comparison of (79) with (81) which indicate the impact of evidence from other sequences on our beliefs about the probability of failure-free operation, or *reliability function*, of sequence $\mathcal{A}_k$. In each case, in our 'no-observed-failures' situation, the admission of evidence from sequences $\langle \mathcal{A}_i \rangle_{i=1}^{k-1}$ introduces a common

[17] As far as reliability prediction is concerned, the significance of our specification and parameterisation $\{f_p(\cdot|\theta)\,;\, \theta \in \Theta\}$ of a collection of possible P-distributions, and the significance of our choice of prior Prior_θ over this collection, is contained entirely in the resulting distribution of the moment-vector (82).

factor $\prod_{i=1}^{k-1} \mu_{0,n_i}$ into the arguments of the unconditional **E**-operators in both the numerator and the denominator.

5.3.4 Some Questions About Model Implications

Some particular questions of interest are:–

- How does our confidence in $\mathcal{A}_k$ behave as a function of the *number* of previous sequences, when these have all been observed to contain no failures for an equal number $n_i = n$ of demands?

- For a fixed number $k-1$ of previous sequences, observed for fixed periods, how much does one failure in one sequence spoil things as far as our confidence in sequence $\mathcal{A}_k$ is concerned? Then, how much does one failure in each of two of these sequences affect our conclusions? And so on, for $1, 2, 3, \ldots$ out of the $k-1$ previous sequences exhibiting one failure each, and the rest no failures?

- Is it best, given a fixed number, in total, of demands on previous ⟨product, environment⟩ pairs, to know that fewer ⟨product, environment⟩ pairs have shown failure-free operation over a larger number of demands each, or that a larger number of such pairs have each worked perfectly over a relatively small number of demands each? How important is this distinction, in terms of its effect on the size of the amount by which our confidence in $\mathcal{A}_k$ is improved by observation of the previous sequences?

- Where there have been some previous failures, and again keeping the total number of previous demands constant, do we prefer to hear that those failures have been concentrated amongst a small number, or even a single, previous sequence, or is it less depressing news for the current ⟨product, environment⟩ pair if we find that the previous sequences all showed a similar level of unreliability? (It seems obvious that, if we are especially interested in the reliability of the current sequence $\mathcal{A}_k$, then, given the choice, we should in general prefer observed failures to have been found in previous sequences, i.e. to be failures associated with software products or environments other than the current one.)

- Which, if any, of the answers to the above four groups of questions holds quite generally for all possible parametric distribution families $\{f_p(\cdot|\theta)\,;\,\theta \in \Theta\}$, and for all possible prior beliefs Prior_θ? Does a preference for, say, all failures to have occurred in a single previous sequence, rather than for the same total number of failures to have been distributed between several previous sequences, depend on specific characteristics of our assumed prior distributions?

- Extending further such consideration of the influence of our choice of Prior_θ, we might even ask, the family $\{f_p(\cdot|\theta)\,;\,\theta\in\Theta\}$ being assumed specified, about variation in the quantities of interest over *the space of all priors* Prior_θ, and, in particular, ask various, probably mathematically non-trivial, questions about extrema here. In practice Prior_θ ought ideally to capture genuine prior belief. However, given that the conclusions from this model are likely to be highly dependent on the shape of our prior belief, it is important to try to gain a general understanding of more precisely *how*, and to what extent, various different distributions Prior_θ will effect our conclusions. What are the extremes, in both the sense of extreme favourability and extreme unfavourability to high current reliability predictions, of the prior beliefs we might hold? Are the mathematical extremes here at all plausible in practice? Can we introduce geometric constraints on the shape of the prior distribution, such as unimodality, or continuous density function, or upper and lower limits on the values of P_i admitted as having positive probability, and how do such constraints effect the answers to our questions about extrema?

There are several other similar questions that can be asked, given this general model structure. The next section contains some tentative results relating to some of the above questions in the context of some simple instantiations of this 'similar products' model. In terms of this basic model structure, there is of course an added complication to these questions: It may well turn out that the questions as we have just listed them are insufficiently precise. What precisely does 'high current reliability' mean in the last bullet point above? It might transpire that the answers will depend on specifically how we choose to quantify the reliability of the present ⟨product, environment⟩ pair. For example, in terms of the updated distribution (62) of P_k, or in terms of the associated[18] reliability function. And in each of these two cases, how do we compare two *functions*? Two alternative $\mathcal{A}_k$-reliability functions resulting from different observed behaviours of previous sequences might cross at future demand $m=10^4$, for example. That is to say, the previous-sequence observations which give the greater confidence in the current sequence's long-term reliability may give lower confidence in its short-term reliability. In such a case, the set of previous-sequence observations which we would prefer to see would depend on factors such as our predicted operational lifetime of the ⟨product, environment⟩ pair $\mathcal{A}_k$.

[18] using equation (67)

5.3.5 Examples of Particular Choices of Prior Distributions for P given Θ, and for Θ

We shall retain throughout what follows our original assumptions that each sequence $\mathcal{A}_i$ constitutes a Bernoulli trials failure process with unknown parameter P_i, and that the (P_i) sequence is i.i.d. conditionally given an unknown sequence-family characterising parameter θ. To generate particular cases of our model we are then left with the tasks of choosing the distribution family $\{f_p(\cdot|\theta)\,;\,\theta\in\Theta\}$ and the single prior distribution Prior_θ over this family. To begin with, we will investigate a simple two-point distribution $f_p(\cdot|\theta)$ in §5.3.5.1. Though clearly a simplification, this model instantiation can be argued to have some practical relevance to attempts to certify 'ultra-high reliability (product, environment)s' as well as illustrating in a simple way the structure of our general model.

5.3.5.1 Two-point f_p, with θ interpreted as mass at fixed points of support, one of which is p=0

Suppose $P|\theta$ has a two-point distribution with θ equal to the probability[19] assigned to $p = 0$. So we assume

$$\mathbf{P}[P_i = 0|\theta] = \theta\,; \qquad \mathbf{P}[P_i = \pi|\theta] = 1 - \theta\,.$$

Thus we assume that, for each sequence $\mathcal{A}_i$ in the family, θ is the probability that $P_i = 0$. For example, we could imagine a formal verification technique is applied to each software product and that this technique fails—to deliver a perfect ($p = 0$) (product, environment)—with an unknown probability $1 - \theta$. When this happens, we assume that the resulting program failure probability is known; for example $\pi = 10^{-5}$ might be used, if these are high-integrity products. This assumption of a single known value for p whenever $p \neq 0$ would perhaps better be relaxed by allowing a distribution for p, but it simplifies the application of our general model, retains sufficient flexibility to provide a useful illustration of the model, and could perhaps be justified on the grounds of conservatism by assuming a worst case value π for the non-zero p. We can now apply our previous results to the analysis of this model. Though now containing a discrete distribution component, the model can if desired be obtained directly from the results of sections 5.3.1–5.3.2 by defining the common density of each of the P_i in terms of Dirac delta functions[20]

$$P|\theta \sim f_p(p|\theta) = \theta\,\delta(p) + (1 - \theta)\,\delta(p - \pi), \tag{84}$$

[19] Contrast with the also interesting case where θ defines the *position* of the points of support of f_p—Or perhaps further generalisations where the positions and masses of two points of support for P are represented by a two or three dimensional θ.

[20] Provided we agree either to slightly extend our usual range $0 \leq p \leq 1$ of integration with respect to p, or to modify the usual definition of the δ-functions so that $\int_0^1 \delta(p)\,dp$ and $\int_0^1 \delta(p-1)\,dp$ should evaluate to 1 rather than $\frac{1}{2}$

say, where $0 < \pi < 1$ is fixed.

The likelihood of θ given periods $\langle n_i \rangle_{i=1}^{k}$ of observation of k sequences (c.f. equation (56)) is then

$$L(\theta\,;\,\langle n_i, r_i\rangle_{i=1}^{k}) = \prod_{1\le i\le k} \binom{n_i}{r_i} \cdot \prod_{\substack{1\le i\le k,\\ r_i>0}} \pi^{r_i}(1-\pi)^{n_i-r_i}(1-\theta) \cdot \prod_{\substack{1\le i\le k,\\ r_i=0}} \{(1-\pi)^{n_i}(1-\theta)+\theta\}$$

To within a factor which does not depend on θ we can write this as

$$L(\theta\,;\,\langle n_i, r_i\rangle_{i=1}^{k}) \propto L_k(\theta) = \prod_{\substack{1\le i\le k,\\ r_i>0}} (1-\theta) \cdot \prod_{\substack{1\le i\le k,\\ r_i=0}} \{(1-\pi)^{n_i}(1-\theta)+\theta\}$$

1st Case: General Prior$_\theta$ It follows that the posterior distribution of Θ given this data is now

$$\Theta \big| \langle n_i, r_i\rangle_{i=1}^{k} \sim \frac{L_k(\theta)\mathrm{Prior}_\theta(\theta)}{\int_{\theta\in\Theta} L_k(\theta)\mathrm{Prior}_\theta(\theta)\,d\theta}$$

In fact, since the parameter θ has a direct interpretation here as a probability, we must have $\Theta \subseteq [0,1]$, and we can assume without loss of generality that Prior$_\theta$ is extended in such a way that $\Theta = [0,1]$. We shall assume this has been done for the remainder of this section. If there has been a failure in the observed part of the current sequence (i.e. if $r_k > 0$), then the updated posterior distribution of P_k given our observation is trivially just $P_k = \pi$ with certainty. In this case, future reliability prediction is simply that of a Bernoulli trials failure process with parameter π. In the interesting case where the current sequence $\mathcal{A}_k$ has so far exhibited no failure we have[21]

$$\mathbf{P}\Big[P_k = 0 \Big| \langle n_i, r_i\rangle_{i=1}^{k-1}, n_k, r_k{=}0\Big] = \frac{\int_0^1 \theta L_{k-1}(\theta)\mathrm{Prior}_\theta(\theta)\,d\theta}{\int_0^1 \{(1-\pi)^{n_k}(1-\theta)+\theta\}\,L_{k-1}(\theta)\mathrm{Prior}_\theta(\theta)\,d\theta};$$

$$\mathbf{P}\Big[P_k = \pi \Big| \langle n_i, r_i\rangle_{i=1}^{k-1}, n_k, r_k{=}0\Big] = \frac{\int_0^1 (1-\pi)^{n_k}(1-\theta)L_{k-1}(\theta)\mathrm{Prior}_\theta(\theta)\,d\theta}{\int_0^1 \{(1-\pi)^{n_k}(1-\theta)+\theta\}\,L_{k-1}(\theta)\mathrm{Prior}_\theta(\theta)\,d\theta}.$$

Equation (67) tells us that this pair of probabilities may now be substituted in the following equation to obtain a reliability prediction for sequence $\mathcal{A}_k$

$$\mathbf{P}\Big[\text{No failure in next } m \text{ demands} \Big| \langle n_i, r_i\rangle_{i=1}^{k-1}, n_k, r_k{=}0\Big] = \mathbf{P}\Big[P_k = 0 \Big| \langle n_i, r_i\rangle_{i=1}^{k-1}, n_k, r_k{=}0\Big] + \mathbf{P}\Big[P_k = \pi \Big| \langle n_i, r_i\rangle_{i=1}^{k-1}, n_k, r_k{=}0\Big](1-\pi)^m \qquad (85)$$

This is equivalent to the $r_k = r'_k = 0$, $n'_k = m$ case of equation (81), where for this model structure we have

$$\mu_{r,s} = \begin{cases} (1-\pi)^s(1-\theta)+\theta, & \text{if } r = 0, \\ \pi^r(1-\pi)^s(1-\theta), & \text{if } r > 0. \end{cases} \qquad (86)$$

Consider now the case in which our periods of observation have given rise to no observed failure of any of the family $\langle \mathcal{A}_i \rangle_{i=1}^{k}$. Expressing the up-to-date distribution of P_k slightly more concisely using

[21] This formula holds also for $k = 1$ if we put $L_0(\theta) = 1$

the 'odds' form mentioned in equations (63) and (64), we can compare it with the odds obtained from observation of only the present or only previous sequences. Cancelling some constants from the likelihood term $L_k(\theta)$ which occurs in both the numerator and the denominator, we find that

$$\frac{\mathbf{P}\left[P_k = 0 \middle| \langle n_i\rangle_{i=1}^{k}, \langle r_i\rangle_{i=1}^{k} = \langle 0\rangle\right]}{\mathbf{P}\left[P_k = \pi \middle| \langle n_i\rangle_{i=1}^{k}, \langle r_i\rangle_{i=1}^{k} = \langle 0\rangle\right]} = \frac{(1+y_k)}{y_k} \frac{\int_0^1 \prod_{i=1}^{k-1}(\theta + y_i)\,\theta \mathrm{Prior}_\theta(\theta)\,d\theta}{\int_0^1 \prod_{i=1}^{k-1}(\theta + y_i)\,(1-\theta)\mathrm{Prior}_\theta(\theta)\,d\theta} \tag{87}$$

$$\frac{\mathbf{P}\left[P_k = 0 \middle| \langle n_i\rangle_{i=1}^{k-1}, \langle r_i\rangle_{i=1}^{k-1} = \langle 0\rangle\right]}{\mathbf{P}\left[P_k = \pi \middle| \langle n_i\rangle_{i=1}^{k-1}, \langle r_i\rangle_{i=1}^{k-1} = \langle 0\rangle\right]} = \frac{\int_0^1 \prod_{i=1}^{k-1}(\theta + y_i)\,\theta \mathrm{Prior}_\theta(\theta)\,d\theta}{\int_0^1 \prod_{i=1}^{k-1}(\theta + y_i)\,(1-\theta)\mathrm{Prior}_\theta(\theta)\,d\theta} \tag{88}$$

$$\frac{\mathbf{P}[P_k = 0 | n_k, r_k{=}0]}{\mathbf{P}[P_k = \pi | n_k, r_k{=}0]} = \frac{(1+y_k)}{y_k} \frac{\mathbf{P}[P_k = 0]}{\mathbf{P}[P_k = \pi]}, \tag{89}$$

where the second term on the right-hand side of (89) is just the prior odds that $P_k = 0$ (before any portion of any sequence has been observed) and where we introduce the notation

$$y_i = \frac{1}{(1-\pi)^{-n_i} - 1}, \quad i = 1 \ldots k. \tag{90}$$

Note that for this two-point model with π assumed known, improving reliability estimates of the current sequence translate directly into an improving up-to-date probability of current ⟨product, environment⟩ perfection. This can be simply expressed in terms of the odds values given by equations (87)–(89). If the prior odds of $\mathcal{A}_k$-perfection is denoted o, and if equations (88) and (89) represent, respectively, improvements on this by factors of $\mathscr{R}$, achieved by means of previous $\mathcal{A}_i$-observation, and $\mathscr{R}'$, by means of direct observation of (product, environment) pair $\mathcal{A}_k$, then we will have a prior $\mathcal{A}_k$-perfection probability of $\left(1 + \frac{1}{o}\right)^{-1} = \int_0^1 \theta \mathrm{Prior}_\theta(\theta)\,d\theta$, improving to posterior $\mathcal{A}_k$-perfection probabilities of $\left(1 + \frac{1}{\mathscr{R}'o}\right)^{-1}$, $\left(1 + \frac{1}{\mathscr{R}o}\right)^{-1}$, and, $\left(1 + \frac{1}{\mathscr{R}\mathscr{R}'o}\right)^{-1}$, respectively, under the three different observation scenarios of equations (60–62). Clearly, it is the factor $\mathscr{R}$ which is of particular interest since it represents the advantage to be gained from incorporating data on the *previous* $\mathcal{A}_i$. There is a need to understand the way that this factor is determined by the combined effect of observations and of our prior distribution Prior_θ.

If failures in some *previous* demand sequences $\mathcal{A}_1 \ldots, \mathcal{A}_{k-1}$ *have* been observed then this is handled (irrespective of how many of these failures there were for each $\mathcal{A}_i$ which was seen to fail at least once) by replacing the corresponding factors $(\theta + y_i)$ by $(1 - \theta)$ in both the numerator and denominator of equations (87) and (88). The expression in terms of y_i here brings to light an interesting limiting case

$$y_i \to \frac{1}{e^{n_i\pi} - 1} \tag{91}$$

which is likely to be a good approximation to reality for pairs (product, environment) which are very reliable, and which is obtained by letting $\pi \to 0$ and $n_i \to \infty$ (for those $\mathcal{A}_i$ which have been

observed not to fail) whilst holding $n_i\pi$ constant for each of these sequences. In this limiting case, the updated distribution of P_k comes to depend only on the products $n_i\pi$, and not, other than via these products, on the values of n_i and π. Some idea of closeness to this limiting case can be obtained, in terms of the value of π, from the crude bounds

$$\frac{1}{e^{\frac{n_i\pi}{1-\pi}}-1} < y_i < \frac{1}{e^{n_i\pi}-1} \tag{92}$$

which are respectively obtained by applying the two well known inequalities

$$\left(1+\frac{x}{n}\right)^n < e^x, \quad n>0,\ x>0 \qquad \text{and} \qquad \left(1-\frac{x}{n}\right)^{-n} > e^x, \quad 0<x<n. \tag{93}$$

For an interpretation of y_i we can say that y_i is a kind of inverse measure of the informativeness to us of our no-failures observation on sequence $\mathcal{A}_i$. Precisely, y_i is the odds of observing what has been observed (i.e. no failures) of sequence $\mathcal{A}_i$, under the assumption that its true failure probability is $P_i=\pi$. So $r_i=0$ with a y_i which is close to zero means that we have observed something about sequence $\mathcal{A}_i$ which would be extremely unlikely under the assumption $P_i=\pi$. Conversely, $r_i=0$ with a very large y_i means that, even if we somehow knew for certain[22] that $P_i=\pi$, we would still be virtually certain not to observe any failures of $\mathcal{A}_i$ in the n_i trials we have carried out. Equations (87–89) confirm that there is virtually no effect on our beliefs about P_k, arising from the observation that $r_i=0$, if the value of y_i is very large : In terms of inferences about P_k, large y_i makes the the observation $r_i=0$ almost equivalent to *no* observation of sequence $\mathcal{A}_i$ (i.e. almost equivalent to $n_i=0$)[23].

We developed our model of §§5.3.1–5.3.2 in a very general setting from which these two-point f_p results are a very special case. However, even for this simple model instantiation, several interesting and non-trivial questions can be asked about how much extra confidence in a current sequence $\mathcal{A}_k$ can be gained from the observation that previous pairs ⟨product, environment⟩ have performed well. The information that previous products have been observed to perform perfectly in their assigned environments over finite observation periods $(n_i)_{i=1}^{k-1}$ is a special case of obvious interest for reasons stated earlier.

By concentrating first on this simple two-point case of our general model, we can avoid immediately having to grapple with many of the complications concerning alternative quantifications of reliability. For this two point model, it is effectively true to say that high reliability of the present ⟨product, environment⟩ $\mathcal{A}_k$ is unarguably equivalent to a large value of the updated probability $\mathbf{P}\left[P_k=0 \middle| (n_i,r_i)_{i=1}^k\right]$. So in the case of this 2-point model there does exist a *single number* which

[22] the worst possible belief we can hold about $\mathcal{A}_i$, however much observing we do, under this two-point model

[23] For a logically consistent definition of y_i in the vacuous $n_i=0$ case, we might use $y_i=\infty$ on the understanding that we can then simply cancel from equations (87–89) all the infinite factors involving y_i

can be said to represent the current $\mathcal{A}_k$ reliability prediction. Thus we can unambiguously order the reliability predictions which would result from two different sets of past failure observations. For instance, (85) shows us that, with a fixed numerical value for π specified by the model, we will not experience the complication of two reliability functions, produced by different past-sequence behaviours, which cross at some number m of demands into the future. We proceed to examine some of the questions of §5.3.4 for this simple model.

Firstly, we can easily see from (89) that observation of failure-free operation in sequence $\mathcal{A}_k$, itself, will improve the odds that this sequence has P_k=0. The odds in favour of P_k=0 increase by a factor $1 + 1/y_k$, or $(1-\pi)^{-n_k}$. (We note that this factor is not influenced by our prior beliefs about Θ.) This remains true, and by an identical proportion (see comments on p124), irrespective of whether observation of previous sequences has occurred, and irrespective of what failure behaviour was observed for those sequences. Intuitively we might expect that observing perfect failure-free behaviour in *previous* sequences should improve the odds that P_k=0 in a similar sort of consistent way, if not to the same extent. To confirm such an effect, and to investigate its magnitude we need to look at the ratio

$$\mathscr{R} = \frac{\mathbf{P}\left[P_k{=}0 \,\middle|\, \langle n_i\rangle_{i=1}^{k}, \langle r_i\rangle_{i=1}^{k} = \langle 0\rangle\right]}{\mathbf{P}\left[P_k{=}\pi \,\middle|\, \langle n_i\rangle_{i=1}^{k}, \langle r_i\rangle_{i=1}^{k} = \langle 0\rangle\right]} \Bigg/ \frac{\mathbf{P}[P_k{=}0 \,|\, n_k, r_k{=}0]}{\mathbf{P}[P_k{=}\pi \,|\, n_k, r_k{=}0]} =$$

$$\frac{\displaystyle\int_0^1 \prod_{i=1}^{k-1}(\theta + y_i)\,\theta \mathrm{Prior}_\theta(\theta)\,d\theta}{\displaystyle\int_0^1 \prod_{i=1}^{k-1}(\theta + y_i)\,(1-\theta)\mathrm{Prior}_\theta(\theta)\,d\theta} \Bigg/ \frac{\displaystyle\int_0^1 \theta \mathrm{Prior}_\theta(\theta)\,d\theta}{\displaystyle\int_0^1 (1-\theta)\mathrm{Prior}_\theta(\theta)\,d\theta}$$

To do this we will first introduce some slightly more concise versions of our existing notation and make some definitions. Define

$$l(\theta) = \prod_{i=1}^{k-1}(\theta + y_i)\,, \quad \theta_1 = \mathbf{P}[P_k{=}0] = \int_0^1 \theta \mathrm{Prior}_\theta(\theta)\,d\theta\,, \quad \text{and} \quad \theta_2 = l^{-1}\left(\int_0^1 l(\theta)\mathrm{Prior}_\theta(\theta)\,d\theta\right).$$

Note that we know from the convexity of l, using Jensen's inequality, and from l's monotonicity, that, for any Prior_θ, we must have $0 \le \theta_1 \le \theta_2 \le 1$. In fact we will have strict inequalities here except in some degenerate cases such as y_i=∞, or where Prior_θ is a single point mass. With this notation it can be shown that the ratio representing our improvement of odds simplifies as follows

$$\mathscr{R} = \frac{\displaystyle\int_0^1 l(\theta)\theta \mathrm{Prior}_\theta(\theta)\,d\theta}{\displaystyle\int_0^1 l(\theta)(1-\theta)\mathrm{Prior}_\theta(\theta)\,d\theta}\,\frac{1-\theta_1}{\theta_1} = 1 + \frac{\displaystyle\int_0^1 l(\theta)\theta \mathrm{Prior}_\theta(\theta)\,d\theta - l(\theta_2)\theta_1}{\theta_1 \displaystyle\int_0^1 l(\theta)(1-\theta)\mathrm{Prior}_\theta(\theta)\,d\theta} \tag{94}$$

where here the numerator and denominator of the ratio on the right-hand side are both positive and, together through this term, express the 'amount of benefit' obtained from observing the non-failure in the previous sequences. We know that the numerator of this ratio (i.e. of the amount by which the ratio $\mathscr{R}$ of the odds exceeds 1) on the right-hand side of (94) is non-negative because it expresses the *covariance*[24] of the random variables Θ and $l(\Theta)$: A random variable cannot be negatively correlated with any variable obtained by applying to it a non-decreasing function. In fact we can express this numerator as an integral of a non-negative function, in either of two slightly different ways

$$\begin{aligned}\int_0^1 l(\theta)\theta \mathrm{Prior}_\theta(\theta)\,d\theta - l(\theta_2)\theta_1 &= \int_0^1 (\theta-\theta_1)\big(l(\theta)-l(\theta_1)\big)\mathrm{Prior}_\theta(\theta)\,d\theta \\ &= \int_0^1 (\theta-\theta_2)\big(l(\theta)-l(\theta_2)\big)\mathrm{Prior}_\theta(\theta)\,d\theta\,. \qquad (95)\end{aligned}$$

It is clear from (94) that the improvement in the odds that P_k=0 which results from the previous sequence observations can be thought of as the result of three interacting influences: the *original odds* (prior to observation of either this or any other sequence) captured in terms of the value of θ_1; the actual detailed description of the *observation of previous sequences* (both their number, and what is observed of each), which we can think of as being summarised by the function $l(\theta)$[25]; and, going beyond the simple *expectation* θ_1, the exact *shape* Prior_θ *of our prior beliefs about* θ. In terms firstly of the previous sequence observations, we can see that $l(\theta)$=constant, corresponding to a large y_i value for each sequence observed, is equivalent to a lack of useful information observed from the previous sequences. At the opposite extreme, the function $l(\theta) = \theta^{k-1}$ is the upper bound on the proportionate variability of l over the unit interval. This represents an upper bound on the improvement of our beliefs about a k^{th} sequence that can arise from observation of periods of perfect operation of the $k-1$ previous sequences.

Now, looking instead at the influence on $\mathscr{R}$ of the form of Prior_θ, we see that, for *whatever* set of past-sequence observations, (94) will have approximately the value 1 in the case where Prior_θ approaches the distribution of a degenerate, constant, random variable. I.e. if we are already more or less certain before observation commences, that $\Theta \approx \theta_1$, then one sequence will have little to tell us about another. At the other extreme of the form of Prior_θ for the same fixed mean θ_1, it seems that high levels of *variation* or *spread* in our prior subjective Θ-distribution will have the opposite effect, magnifying to its limits the significance for sequence $\mathcal{A}_k$ of what we have observed from previous sequences $(\mathcal{A}_i)_{i=1}^{k-1}$. For a particular function $l(\theta)$, these limits are finite, and so we might

[24] deriving from our assumed prior distribution Prior_θ

[25] although, as we have already mentioned, it is the vector of products $n_i\pi$, or to be more exact, in the case where π is not very small, of y_i values defined by (90), which contains the significant part of the previous (product, environment) pairs' influence on our beliefs here, i.e. π as well as the n_i determine l

investigate them further. But we do this as a way of obtaining a slack upper bound on *how much* previous sequences could ever tell us (within the two-point f_p model of this section) rather than because we believe the extreme of Prior$_\theta$-variance is likely to be a realistic model of a person's true prior beliefs about reliability variation within the family of software (product, environment) pairs. For fixed mean θ_1 the most extreme spread in prior beliefs about Θ is given by the distribution Prior$_\theta$ which consists of two point masses: θ_1 at Θ=1 and $1-\theta_1$ at Θ=0. It is easy to see that this Prior$_\theta$[26], when substituted in the left-hand side of (94) gives the P_k=0 odds-increase of

$$\mathscr{R} = \frac{l(1)}{l(0)} = \prod_{i=1}^{k-1} \frac{1+y_i}{y_i} = (1-\pi)^{-\{\sum_{i=1}^{k-1} n_i\}} . \tag{96}$$

In considering the influence of the shape of Prior$_\theta$ on the usefulness of previous sequence observations, a point worth making about the form of the obtainable odds-improvement (94) is the following. If we consider the set of probability distributions Prior$_\theta$ on the unit interval *having some common fixed mean* θ_1, and if we hold fixed the previous-sequence observations (i.e. specify some fixed function $l(\theta)$), then, as we vary Prior$_\theta$ within this set (which, mathematically, is a convex set in a suitable vector space of real measures), we will find that the extrema of (94) must be attained somewhere on the boundary of this set of distributions. This is because $\mathscr{R}$, regarded as a function of the distribution Prior$_\theta$, has a monotonicity property along 'straight lines' in the set of candidate Prior$_\theta$ distributions : When Prior$_\theta$ is a mixture[27], say,

$$\text{Prior}_\theta(\theta) = \lambda p_1(\theta) + (1-\lambda) p_2(\theta) , \quad 0 < \lambda < 1 , \tag{97}$$

of two probability distributions on the unit interval, having a common mean θ_1, but different values, say $\mathscr{R}(p_1) < \mathscr{R}(p_2)$, of the ratio (94), then the value $\mathscr{R}(\text{Prior}_\theta)$ of (94) corresponding to the mixture will satisfy $\mathscr{R}(p_1) < \mathscr{R}(\text{Prior}_\theta) < \mathscr{R}(p_2)$. This in turn follows from the fact that the numerator[28] and denominator of (94) are both non-negative-valued linear functionals of the probability distribution Prior$_\theta$[29]. This kind of reasoning can be used to confirm that (96) is indeed the maximum possible value of $\mathscr{R}$ for a fixed observation function l and mean θ_1 (and in fact, as we see in (96), that the value of this maximum is actually the same for all $0 < \theta_1 < 1$). The value in (96) tends (using the same reasoning as for (92)) to a limit $\exp\left(\sum_{i=1}^{k-1} n_i \pi\right)$ as $\pi \to 0$ keeping each of the the product terms $n_i\pi$ in the exponent constant. In addition to this limiting value, we also have, for finite n_i, an upper bound $\exp\left(\sum_{i=1}^{k-1} n_i \frac{\pi}{1-\pi}\right)$ by the same reasoning as that used to produced (92).

---

[26] strictly speaking a weighted sum of two Dirac delta functions, but note footnote 20 on p133

[27] the argument extends easily to more general mixtures than the discrete mixture of just two distributions used here

[28] use either the left-hand side of (94), or the right-hand side of (94) with the first form of the right-hand side of (95) used as the numerator

[29] essentially we are using the identity $\frac{\lambda a+(1-\lambda)A}{\lambda b+(1-\lambda)B} = \mu\frac{a}{b} + (1-\mu)\frac{A}{B}$ (a convex combination) for any pair of ratios $\frac{a}{b}$ and $\frac{A}{B}$ of positive numbers, where $\mu = \frac{\lambda b}{\lambda b+(1-\lambda)B}$.

It is difficult to imagine how a two-point Prior_θ, such as that required above to attain the maximum (96) effect of past sequences, could possibly arise in practice as a realistic model of subjective prior belief. Some further, more realistic restriction on the set of admissible shapes of the Prior_θ distributions which have some particular mean θ_1 is probably worth exploring. For example the question of how big (94) can become (for some fixed θ_1 and l) when we require that the prior distribution Prior_θ should be *unimodal* looks a more interesting one from a practical point of view, if, unfortunately, more difficult mathematically to solve.

2nd Case: Parametric Restriction of Prior_θ to Beta Family Consider a further model specialisation in the form of the assumption that Prior_θ is a Beta distribution[30]. This assumption is convenient for numerical reasons since it allows us to expand out the θ-polynomials in (87) and integrate analytically, term by term, using

$$\int_0^1 \theta^m (1-\theta)^{m'} \text{Prior}_\theta(\theta)\, d\theta = \frac{\beta(a+m, b+m')}{\beta(a,b)} \tag{98}$$
$$= \frac{a}{(a+b)} \frac{a+1}{(a+b+1)} \cdots \frac{a+m-1}{(a+b+m-1)} \frac{b}{(a+b+m)} \frac{b+1}{(a+b+m+1)} \cdots \frac{b+m'-1}{(a+b+m+m'-1)}$$

Hence problems of optimisation (of say the ratio (94)) within this parametric restriction of the choice of prior beliefs Prior_θ become scalar optimisation problems with respect to the two independent variables a and b, rather than mathematically more difficult optimisations in which the independent variable is a 'point' lying in a convex set of general probability measures (contained within a larger vector space of general measures on the unit interval).

Note also that the Beta Prior_θ assumption contains, as the limiting cases $a, b \to 0$ with a/b constant, the largest-variance[31], 2-point Prior_θ distribution of a given mean, as mentioned above, and also contains, as the cases $a, b \to \infty$ with a/b constant, the degenerate Prior_θ under which there is no possibility to learn about P_k from observation of other $\mathcal{A}_i$, $i = 1 \ldots k-1$.

It is of interest[32] to fix the prior probability $\mathbf{P}[P_k{=}0]$ and to examine how variation of the parameters a, b of the Beta distribution for Θ, subject to such a constraint, affects the amount that can be learned from previous sequences. In fact, fixing this prior probability is equivalent simply to fixing the ratio a/b, i.e. the prior odds, say o, that $P_k{=}0$. If we reparameterise the Beta distribution in terms of this odds o and the parameter b, then, as we have just said, for fixed o, the

[30] Not to be confused with the assumption, in the example of the next sub-section, that θ is $\langle a, b\rangle$ the parameter-pair of a Beta distribution, where that Beta distribution is our f_p-distribution for P_i given θ.

[31] the coefficient of variation of the Beta(a,b) distribution is $\sqrt{b/\{a(a+b+1)\}}$

[32] because this will provide an upper bound on 'how much' a given amount of previous-sequence data can tell us, under this model : not because we wish to suggest that such an optimisation is a valid method of 'eliciting' the shape of genuine Bayesian prior beliefs, nor even that this bound will be closely approachable in a genuine analysis of real systems

greatest (or least) possibility, as measured by the ratio $\mathscr{R}$ of equation (94), for learning about the current ⟨product, environment⟩ pair from perfect behaviour of all observed previous $\mathcal{A}_i$, $1 \leq i < k$, corresponds to the extreme limiting case $b \to 0$ (or $b \to \infty$). In fact, for this model, we can verify that, as we might expect, in this situation where all previous sequences have shown perfect behaviour over their fixed observation periods, $\mathscr{R}$ is a monotonic non-increasing function of b. See Appendix B.3 for the details of this proof. This $\mathscr{R}(b)$-monotonicity provides us some information, at least in the Beta case, about what happens as we vary the shape of the distribution Prior_θ, for fixed mean θ_1 (i.e. fixed o given by $o = \theta_1/(1-\theta_1)$), between the two extreme cases of the constant (zero variance) prior distribution corresponding to $\mathscr{R} = 1$ at one extreme, and the other extreme of the maximal-variance, 2-point distribution with mass only at Θ=0, 1 corresponding to the largest possible value of $\mathscr{R}$, given by (96). Fixing $o = a/b$ and varying b for our Beta Prior_θ is equivalent to moving along the line $b = oa$ in the $\langle a, b\rangle$-plane. In this plane of Beta distribution parameters, it is precisely the points *inside* the unit square (i.e. the pairs $\langle a, b\rangle$ with $\max(a, b) < 1$) which correspond to bimodal Beta distributions. The Beta distributions corresponding to the outside or boundary of this square (those for which $\max(a, b) \geq 1$) are unimodal Beta distributions. Hence, in the case of a Beta Prior_θ, we can conclude that, moving back from infinity (the degenerate constant $\Theta = \frac{o}{o+1}$ prior) along the line $b = a/o$ towards the origin gives a steadily increasing variance of Prior_θ, and simultaneously an increasing value of $\mathscr{R}$, with both the maximal variance, and the maximal $\mathscr{R}$, which can arise from a unimodal Beta prior, being attained on the boundary of the square at $\langle a, b\rangle = \langle 1, \frac{1}{o}\rangle$, if $o \geq 1$, or at $\langle a, b\rangle = \langle o, 1\rangle$, if $o \leq 1$. Expressions for the accompanying $\mathscr{R}$ values are obtained by substituting $\text{Prior}_\theta(\theta) \propto (1-\theta)^{o^{-1}-1}$ and $\text{Prior}_\theta(\theta) \propto \theta^{o-1}$ in the formula (94). If this movement towards the origin of the $\langle a, b\rangle$-plane is continued *inside* the unit square, then, as the origin is approached, increasingly extreme forms of bimodality in the prior for Θ result in the variance of Prior_θ approaching a maximum value of $o/(o+1)^2$, and $\mathscr{R}$ approaching the extreme limiting case of (96). This latter extreme case has the rather absurd interpretation of all sequences being known to have the same P_i value in advance of observation, but with uncertainty somehow persisting (despite such a strong belief in uniformity of failure rates) as to whether the actual value of this universal failure probability is 0 or π. It seems that this smaller $\mathscr{R}_{\max}$ arising from the restriction to unimodal priors might be a more realistic upper bound on the attainable size of the improvement $\mathscr{R}$. However, we have not answered the general question under a unimodal Prior_θ of how large $\mathscr{R}$ can be for given fixed observations periods throughout which the $k-1$ previous sequences have all been failure-free. We do not know how much greater we might be able to make $\mathscr{R}$ if we experiment with unimodal Prior_θ outside the Beta family. Mathematically this appears to

be a difficult constrained optimisation problem.

We can also use the analytic tractability gained by this Beta restriction on Prior_θ to investigate the question[33] of how the shape of our prior beliefs affect the preferred allocation of a fixed number of demands between a number of past ⟨product, environment⟩ pairs $\mathcal{A}_i$, i=1,2,...,k-1. The algebra is a bit awkward, but even with this Beta family assumption for Prior_θ, and while limiting ourselves to the more tractable cases of small k, we are still able to establish the following result: Our prior beliefs about the ⟨product, environment⟩ *perfection probability* $\theta = \mathbf{P}[P_i{=}0|\theta]$ are of sufficient importance that the answer to the question posed in the third bullet point at the beginning of §5.3.4 may be 'No', 'Yes', or something more complicated, depending on the *combined effects* of the shape Prior_θ of our prior beliefs and the total amount N (or Z, see below) of past product data we have available. This establishes a principle that there are qualitative, as well as quantitative, questions concerning what our model says about the influence of observations of past ⟨product, environment⟩ pairs which cannot be answered until we have described the shape of our prior uncertainty about reliability variation between the ⟨product, environment⟩ pairs of our family $\langle \mathcal{A}_i \rangle$. Suppose we have a total number N of demands to distribute between $k-1$ previous ⟨product, environment⟩ pairs $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_{k-1}$ and that our objective is to increase our confidence in the reliability of $\mathcal{A}_k$ as much as possible. To simplify the notation slightly, we work in terms of $Z = N(-\log(1-\pi))$, which we might choose to think of as a quantification of the total amount of past-sequence observation 'adjusted' for the difficulty of our task of discriminating between $P_k{=}0$ and $P_k{=}\pi$. (Clearly, the closer π is to zero, the more difficult it becomes to discriminate, by means of data, between the two possibilities $P_k{=}0$ and $P_k{=}\pi$.) We can describe our allocation of this past data between the $k-1$ previous sequences by means of a vector $\langle \nu_1, \nu_2, \ldots, \nu_{k-1} \rangle$, with $0 \leq \nu_i \leq 1$, $\nu_1 + \nu_2 + \ldots + \nu_{k-1} = 1$ where $\nu_i = n_i/N$.

Taking first the simplest case of just two previous sequences, i.e. $k = 3$, equation (94) (using (88) and (90)) can be written

$$\mathscr{R} = \frac{b}{a} \cdot \frac{\int_0^1 \left[e^{\nu_1 Z}\theta + (1-\theta)\right]\left[e^{\nu_2 Z}\theta + (1-\theta)\right]\theta^a(1-\theta)^{b-1}\,d\theta}{\int_0^1 \left[e^{\nu_1 Z}\theta + (1-\theta)\right]\left[e^{\nu_2 Z}\theta + (1-\theta)\right]\theta^{a-1}(1-\theta)^{b}\,d\theta} \tag{99}$$

Expanding the products of square-bracketed terms and using (98), this reduces to

$$\mathscr{R} = \frac{(a+2)(a+1)e^{(\nu_1+\nu_2)Z} + (a+1)b\left(e^{\nu_1 Z} + e^{\nu_2 Z}\right) + (b+1)b}{(a+1)ae^{(\nu_1+\nu_2)Z} + a(b+1)\left(e^{\nu_1 Z} + e^{\nu_2 Z}\right) + (b+2)(b+1)} \tag{100}$$

Remembering that $\nu_1 + \nu_2 = 1$ and taking $\left|\nu_1 - \frac{1}{2}\right|$ as our measure of unevenness of allocation of

[33] The third bullet point at the beginning of §5.3.4

the N (or Z) past observations between $\mathcal{A}_1$ and $\mathcal{A}_2$, we find that

$$\begin{aligned}\mathscr{R} &= \frac{(a+2)(a+1)e^Z + 2(a+1)be^{\frac{Z}{2}}\cosh\left[(\nu_1-\frac{1}{2})Z\right] + (b+1)b}{(a+1)ae^Z + 2a(b+1)e^{\frac{Z}{2}}\cosh\left[(\nu_1-\frac{1}{2})Z\right] + (b+2)(b+1)} \\ &= \frac{a+1}{b+1}\left[\frac{b}{a} + \frac{(a+b+2)\left(e^Z - \frac{b(b+1)}{a(a+1)}\right)}{(a+1)ae^Z + 2a(b+1)e^{\frac{Z}{2}}\cosh\left[(\nu_1-\frac{1}{2})Z\right] + (b+2)(b+1)}\right] \qquad (101)\end{aligned}$$

and it is apparent that $\mathscr{R}$ is a monotonic function of $\left|\nu_1 - \frac{1}{2}\right|$ with bounds

$$\mathscr{R}_1 = \frac{(a+1)e^Z + b}{ae^Z + b + 1}, \qquad \text{achieved at } \nu_1 = 0, 1 \qquad (102)$$

$$\mathscr{R}_2 = \frac{(a+2)(a+1)e^Z + 2(a+1)be^{\frac{Z}{2}} + (b+1)b}{(a+1)ae^Z + 2a(b+1)e^{\frac{Z}{2}} + (b+2)(b+1)}, \qquad \text{achieved at } \nu_1 = \tfrac{1}{2} \qquad (103)$$

We showed earlier that, irrespective of how much direct observation of $\mathcal{A}_3$ has been done, $\mathscr{R}$ is the factor by which the odds that $\mathcal{A}_3$ is perfect are improved by the past sequence (in this case $\mathcal{A}_1$ and $\mathcal{A}_2$) observation. Thus we conclude, for this simple $k = 3$ case, that

- If our prior beliefs for θ are Beta(a, b) with $\frac{b(b+1)}{a(a+1)} = e^Z$, then our posterior probability that P_3=0 is unaffected by changes in allocation of a fixed total amount Z of past sequence observation between $\mathcal{A}_1$ and $\mathcal{A}_2$. In this case, the observation of $\mathcal{A}_1$ and $\mathcal{A}_2$ improves our odds that $\mathcal{A}_3$ is perfect by a fixed factor $\mathscr{R} = \frac{(a+1)b}{a(b+1)}$.

- If our prior beliefs are Beta(a, b) with $\frac{b(b+1)}{a(a+1)} > e^Z$ then our posterior probability that P_3=0 is a strictly increasing function of $\left|\nu_1 - \frac{1}{2}\right|$ (i.e. we prefer our previous observations to have been allocated as unevenly as possible between the two previous sequences $\mathcal{A}_1$ and $\mathcal{A}_2$). In this case we have $\mathscr{R}_2 \leq \mathscr{R} \leq \mathscr{R}_1$ as we vary ν_i. If all of these previous observations are concentrated on only one past $\mathcal{A}_i$, then the maximum possible improvement $\mathscr{R}$ of odds that P_3=0 is attained as $\mathscr{R}$=$\mathscr{R}_1$.

- If our prior beliefs are Beta(a, b) with $\frac{b(b+1)}{a(a+1)} < e^Z$ then our posterior probability that P_3=0 is a strictly decreasing function of $\left|\nu_1 - \frac{1}{2}\right|$ (i.e. we prefer our previous observations to have been allocated as evenly as possible between the two previous sequences). Here we have $\mathscr{R}_1 \leq \mathscr{R} \leq \mathscr{R}_2$ as we vary ν_i. If these previous observations are exactly evenly allocated between $\mathcal{A}_1$ and $\mathcal{A}_2$, then the maximum possible improvement $\mathscr{R}$ of odds that P_3=0 is attained as $\mathscr{R}$=$\mathscr{R}_2$. (Of course this is only possible to do *exactly* when N is even.)

So, in general, we have shown that, supposing N and π to be given (so that Z is fixed) then we cannot answer the question about whether a person prefers the observation of previous ⟨product, environment⟩ pairs to be allocated evenly between those previous $\mathcal{A}_i$ without first clarifying the shape of that person's prior beliefs about the unknown perfection probability parameter θ.

However we can draw a few conclusions of a more general nature for this k=3 case. Suppose that the person's prior probability that a randomly selected $\mathcal{A}_i$ is perfect $\mathbf{E}[\theta]$ has been stated, and we know that their Prior$_\theta$ is in the Beta family. Then the ratio a/b is determined, and so, if $\mathbf{E}[\theta] > \frac{1}{2}$, i.e. if $a > b$, it must be true for any value of Z that

$$\frac{b(b+1)}{a(a+1)} < 1 < e^Z$$

giving a preference for even allocation of observations between $\mathcal{A}_1$ and $\mathcal{A}_2$, whatever the exact values of a and b. Similarly it can be shown that if $\frac{1}{2} \geq \mathbf{E}[\theta] > (1+e^{\frac{Z}{2}})^{-1}$ then the same preference will be found; whereas if $\mathbf{E}[\theta] < (1+e^Z)^{-1}$ then the converse must apply and we will prefer the past observations to be concentrated as much as possible on a single $\mathcal{A}_i$. In terms of the stated prior expectation for θ, the remaining possibility $(1+e^Z)^{-1} < \mathbf{E}[\theta] < (1+e^{\frac{Z}{2}})^{-1}$ corresponds to the situation in which the preference may be for or against even distribution of past observations between $\mathcal{A}_1$ and $\mathcal{A}_2$, depending on the exact values of the parameters a, b and the value of Z. These conclusions follow easily, with the restriction to a Beta Prior$_\theta$, from the facts that $\frac{b(b+1)}{a(a+1)}$ will lie between $\frac{a}{b}$ and $\frac{a^2}{b^2}$ for all $a > 0$, $b > 0$, and that the prior expectation is defined in terms of the Beta parameters by $\mathbf{E}[\theta] = (1+\frac{b}{a})^{-1}$.

We can, without too much difficulty, gain some understanding of what happens when we are considering the effects of the allocation of a fixed amount of observations between *three* previous sequences $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, i.e. in the case k=4. We take the Euclidean distance (which is proportional to the sample standard deviation of $\{\nu_1, \nu_2, \nu_3\}$)

$$r = \sqrt{(\nu_1 - \tfrac{1}{3})^2 + (\nu_2 - \tfrac{1}{3})^2 + (\nu_3 - \tfrac{1}{3})^2} = \sqrt{\nu_1^2 + \nu_2^2 + \nu_3^2 - \tfrac{1}{3}}$$

between the points $\langle \nu_1, \nu_2, \nu_3 \rangle$ and $\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$ as a measure of *how unevenly the past observations are distributed* between the three available previous (product, environment) pairs $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$. The maximum allowable value of r is clearly $\sqrt{(1-\frac{1}{3})^2 + (\frac{1}{3})^2 + (\frac{1}{3})^2} = \sqrt{\frac{2}{3}}$. Equation (94), with the Beta(a, b) Prior$_\theta$, now expands to

$$\mathscr{R} = \frac{(a+3)(a+2)(a+1)e^Z + (a+2)(a+1)be^Z\left(e^{-\nu_1 Z} + e^{-\nu_2 Z} + e^{-\nu_3 Z}\right) + (a+1)(b+1)b\left(e^{\nu_1 Z} + e^{\nu_2 Z} + e^{\nu_3 Z}\right) + (b+2)(b+1)b}{(a+2)(a+1)ae^Z + (a+1)a(b+1)e^Z\left(e^{-\nu_1 Z} + e^{-\nu_2 Z} + e^{-\nu_3 Z}\right) + a(b+2)(b+1)\left(e^{\nu_1 Z} + e^{\nu_2 Z} + e^{\nu_3 Z}\right) + (b+3)(b+2)(b+1)} =$$

$$\frac{(a+3)(a+2)(a+1)e^Z + (a+2)(a+1)be^{\frac{2Z}{3}}\left(e^{(\frac{1}{3}-\nu_1)Z} + e^{(\frac{1}{3}-\nu_2)Z} + e^{(\frac{1}{3}-\nu_3)Z}\right) + (a+1)(b+1)be^{\frac{Z}{3}}\left(e^{(\nu_1-\frac{1}{3})Z} + e^{(\nu_2-\frac{1}{3})Z} + e^{(\nu_3-\frac{1}{3})Z}\right) + (b+2)(b+1)b}{(a+2)(a+1)ae^Z + (a+1)a(b+1)e^{\frac{2Z}{3}}\left(e^{(\frac{1}{3}-\nu_1)Z} + e^{(\frac{1}{3}-\nu_2)Z} + e^{(\frac{1}{3}-\nu_3)Z}\right) + a(b+2)(b+1)e^{\frac{Z}{3}}\left(e^{(\nu_1-\frac{1}{3})Z} + e^{(\nu_2-\frac{1}{3})Z} + e^{(\nu_3-\frac{1}{3})Z}\right) + (b+3)(b+2)(b+1)} \tag{104}$$

Note, for a, b fixed, and fixed $\langle \nu_i \rangle$, we have a limiting case, representing an upper bound on $\mathscr{R}$, of $\lim_{Z\to\infty} \mathscr{R} = (a+j)/a$, where j is the number of the ν_i that are non-zero. This limiting case corresponds to conclusive information that j of $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ are perfect, accompanied by a complete

absence of operational observation on the other $3-j$ previous ⟨product, environment⟩ pairs. The expression (104) is more difficult to analyse as a function of $\langle \nu_1, \nu_2, \nu_3 \rangle$ (with $\nu_1 + \nu_2 + \nu_3 = 1$) than its one-dimensional counterpart (101) because of $\mathscr{R}$'s dependence on the *direction* as well as the modulus r of the 3-vector $\langle \nu_i - \frac{1}{3} \rangle$ of differences from the uniform allocation of observations between the previous three sequences $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$. However, as a first approach to understanding something of the behaviour of this expression we can try replacing both numerator and denominator[34] by low order Taylor expansions in the ν_i. For some argument values, such as Z sufficiently small[35], and the Beta parameters a and b lying within certain ranges, this results in an approximate form for $\mathscr{R}$ in a case where both the numerator and denominator depend much more on the *modulus* of $\langle \nu_1 - \frac{1}{3}, \nu_2 - \frac{1}{3}, \nu_2 - \frac{1}{3} \rangle$ than they do on its *direction*. In fact, we are able in each case (i.e. for both numerator and denominator) to contain the influence of the direction of this vector entirely within a remainder term which is negligible under these conditions on the parameters. Furthermore, in each case, this dependence on r can be approximated by a non-negative quadratic whose minimum is at r=0. This approximation, though somewhat simplistic and consequently restricted in terms of the range of parameter values for which it is accurate, is sufficient to guide us in the identification of some examples, analogous to the alternatives found in the k=3 case above, where the uniform allocation of observations to previous sequences, is either a global minimum, or a global maximum of $\mathscr{R}$, as desired. However, for this k=4 case, we can also show that for certain values a, b, Z, the two remainder terms which, only, are the terms influenced by the direction as well as the modulus r of the deviation $\langle \nu_1 - \frac{1}{3}, \nu_2 - \frac{1}{3}, \nu_2 - \frac{1}{3} \rangle$ from uniformity may become larger and acquire a significant role in determining $\mathscr{R}$'s behaviour. In some of these cases, in contrast to what we found above for the k=3 situation, the uniform allocation $\nu_i = \frac{1}{3}$, $i = 1, 2, 3$, may turn out to be *neither* the global minimum nor the global maximum point of $\mathscr{R}$.

Taking the two ν_i-dependent summands in the numerator obtained above, it is shown in Appendix B.4 that if we define

$$\alpha = \frac{1}{2} \log \left(\frac{b+1}{a+2} \right) - \frac{Z}{6} \tag{105}$$

then a Taylor expansion of degree 2 at the point $\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$ in ν_i, $i = 1, 2, 3$ (still assumed confined to the plane $\nu_1 + \nu_2 + \nu_3 = 1$), with remainder term, gives us

$$(a+2)(a+1)be^{\frac{2Z}{3}} \sum_{i=1}^{3} e^{(\frac{1}{3} - \nu_i)Z} + (a+1)(b+1)be^{\frac{Z}{3}} \sum_{i=1}^{3} e^{(\nu_i - \frac{1}{3})Z} =$$

[34] We note that the denominator is obtained from the numerator by replacing a, b by $a-1, b+1$ so the reasoning only has to be done once.

[35] Z will tend of necessity to be small for small π (which we hope π should be for highly reliable systems) since it becomes infeasible to carry out the very large number of demands then necessary to make Z larger

$$e^{\frac{Z}{2}}(a+1)b\sqrt{(a+2)(b+1)}\Big\{6\cosh(\alpha)+\cosh(\alpha)Z^2r^2+\frac{Z^3}{3}\sum_{i=1}^{3}(\nu_i-\tfrac{1}{3})^3\sinh\big[\alpha+u(\nu_i-\tfrac{1}{3})Z\big]\Big\}$$

for some value $0 < u < 1$ (u probably varies with a, b, Z and the ν_i but is confined to the unit interval). Of the terms inside the curly brackets, the final, remainder term is the only term that depends on the allocation proportions ν_i in a way not confined purely to a dependence on the modulus r. Hence, this term, and a corresponding term (with different u) in the analogous Taylor expansion of the denominator of $\mathscr{R}$, are the two which when small enough to ignore, lead to a much simplified behaviour of $\mathscr{R}$. Note that this approximation (obtained by disregarding the remainder term) is best thought of as a quadratic in r—not rZ—because the coefficients are functions of Z, but not of r. We will not look in detail[36] at the general conditions affecting the size of the two remainder terms as proportions respectively of the numerator and denominator of $\mathscr{R}$. We merely clarify the situation a little by remarking that we can expand

$$\frac{Z^3}{3}\sum_{i=1}^{3}(\nu_i-\tfrac{1}{3})^3\sinh\left[\alpha+u(\nu_i-\tfrac{1}{3})Z\right]=$$
$$\frac{Z^3}{3}\Big\{\sinh(\alpha)\sum_{i=1}^{3}(\nu_i-\tfrac{1}{3})^3\cosh\big[u(\nu_i-\tfrac{1}{3})Z\big]+\cosh(\alpha)\sum_{i=1}^{3}(\nu_i-\tfrac{1}{3})^3\sinh\big[u(\nu_i-\tfrac{1}{3})Z\big]\Big\},$$

that the three different arguments of the hyperbolic function in each sum all fall in an interval of length less than Z, and that, simply by virtue of the constraints $\nu_i \geq 0$, $\sum_{i=1}^{3}\nu_i = 1$ we can obtain by elementary calculus the constraints

$$-\frac{1}{36} \leq \max\left(-\frac{r^3}{\sqrt{6}}, -\frac{1}{9}+\frac{r^2}{2}\right) \leq \sum_{i=1}^{3}(\nu_i-\tfrac{1}{3})^3 \leq \frac{r^3}{\sqrt{6}} \leq \frac{2}{9}.$$

Thus, we have reduced $\mathscr{R}$ to an expression of the form

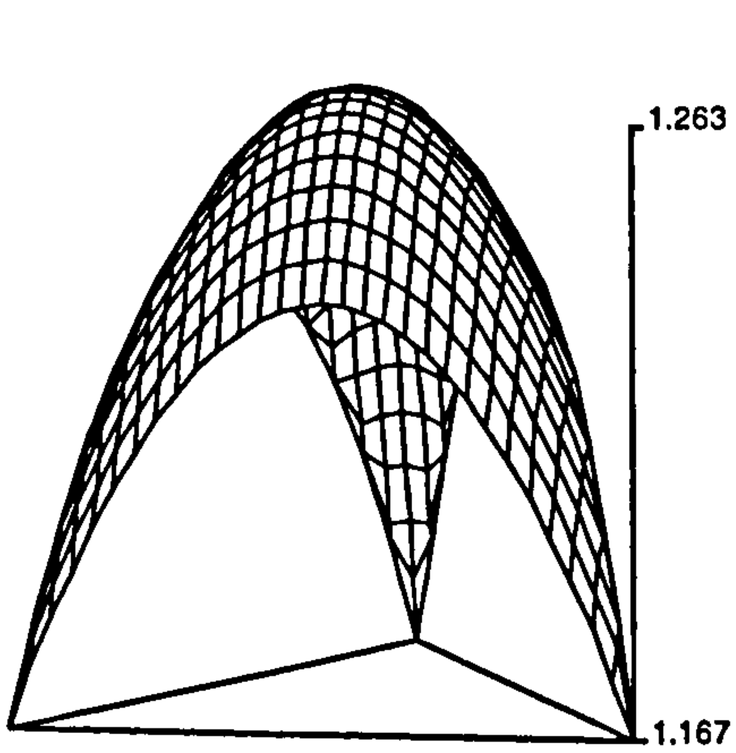
$$\mathscr{R} \approx K\cdot\frac{1+C_1r^2}{1+C_2r^2}, \qquad \text{with } K, C_1, C_2 > 0, \text{ and } 0 \leq r \leq \sqrt{\tfrac{2}{3}} \tag{106}$$

where $\langle K, C_1, C_2\rangle$ can be thought of as a transform of the parameters $\langle a, b, Z\rangle$ of our model, but with the caveat that this approximation (106) is accurate (enough to serve as a useful model of the behaviour of the more complicated function $\mathscr{R}$ of (104)) only within some subdomain—defined by the requirement that the two neglected remainder terms should be sufficiently small—of the set of all possible values $a, b, Z > 0$. Then it is straightforward to conclude that if we hold a, b, Z fixed at some point within this subdomain, and vary the allocation vector $\langle\nu_i\rangle$, we will find $\mathscr{R}$ to be a monotonic function of r with r=0 being the global maximum, or minimum, respectively, as $C_1 < C_2$ or $C_1 > C_2$.

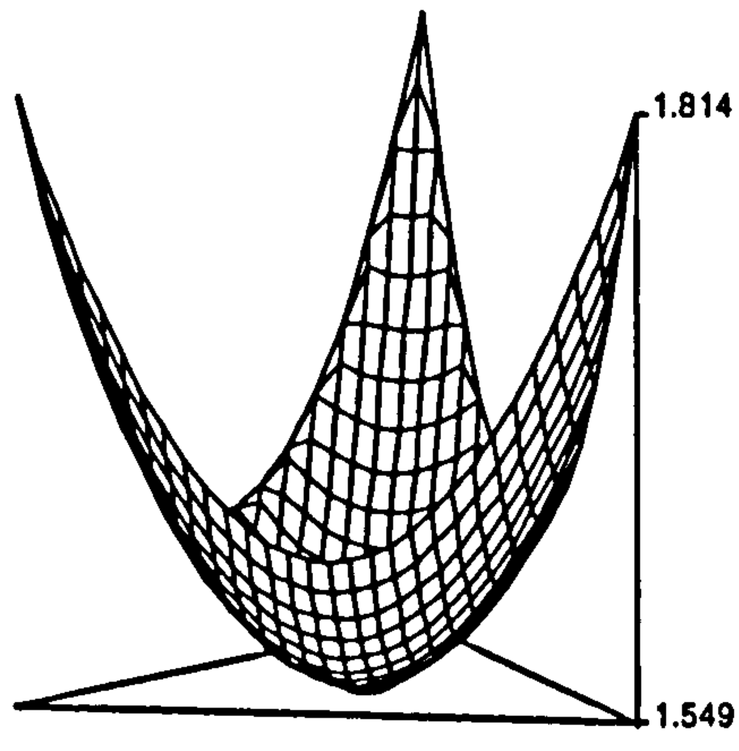
[36] Lengthier analyses are possible, such a transformation of $\langle\nu_i-\frac{1}{3}\rangle$ to polar coordinates (r,ϕ) allowing a further expansion of both the numerator and the denominator of $\mathscr{R}$ into double series, each of whose terms is of the form $a_{ij}r^i\{\substack{\cos\\\sin}\}(j\phi)$ where ϕ is an angle describing the direction of the vector $\langle\nu_i-\frac{1}{3}\rangle$ in the plane $\sum_{i=1}^{3}\nu_i = 1$ (i.e. using a Fourier series expansion in terms of ϕ).

We have plotted $\mathscr{R}$ as a function of $\langle \nu_1, \nu_2, \nu_3 \rangle$ below for four different example values of $\langle a, b, Z \rangle$. Note that in each case it is the exact value (104) which is plotted. The approximation (106) was merely used as a guide to obtaining values of a, b, Z which achieve the three alternative general forms which $\mathscr{R}$ seems to display in this k=4 case, classified here as: decreasing from max at r=0 with approximate rotational symmetry[37]; increasing from min at r=0 with approximate rotational symmetry; and a third, catch-all category of 'other' more complex general behaviour.

If r is small and the parameters $\langle a, b, Z \rangle$ are within the right range, e.g. Z, $\alpha(a, b, Z)$ and $\alpha(a-1, b+1, Z)$ are not too large, then we have seen from (106) that $\mathscr{R}$ will be approximately an increasing function of r if $C_1 > C_2$, and a decreasing function if $C_1 < C_2$. Figure 9 shows two graphs illustrating this case. In these graphs, the base is the equilateral triangular surface: $\nu_1, \nu_2, \nu_3 > 0$,



Left plot : $\langle a, b, Z \rangle = \langle 5, 0.25, 2 \rangle$;



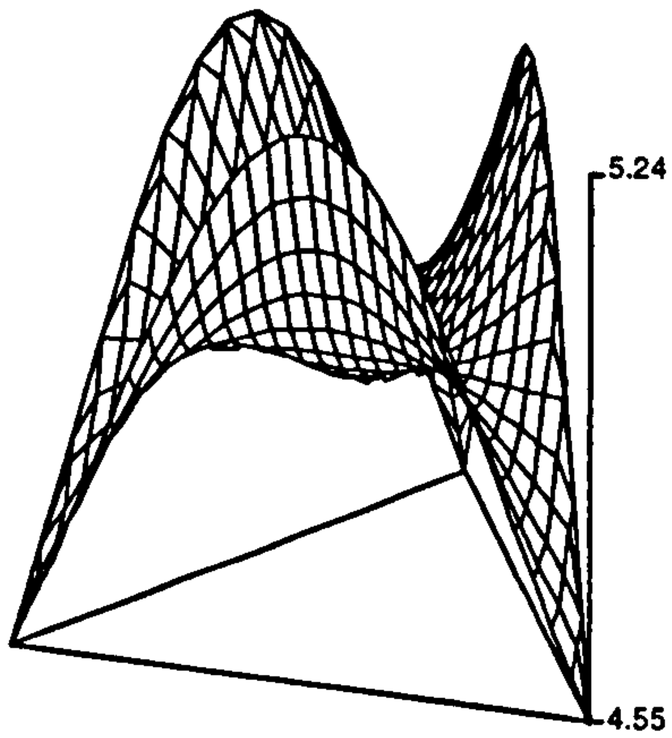
Right plot : $\langle a, b, Z \rangle = \langle 0.25, 5, 2 \rangle$

Figure 9: Plots of $\mathscr{R}$ vs. $\nu_1+\nu_2+\nu_3$=1

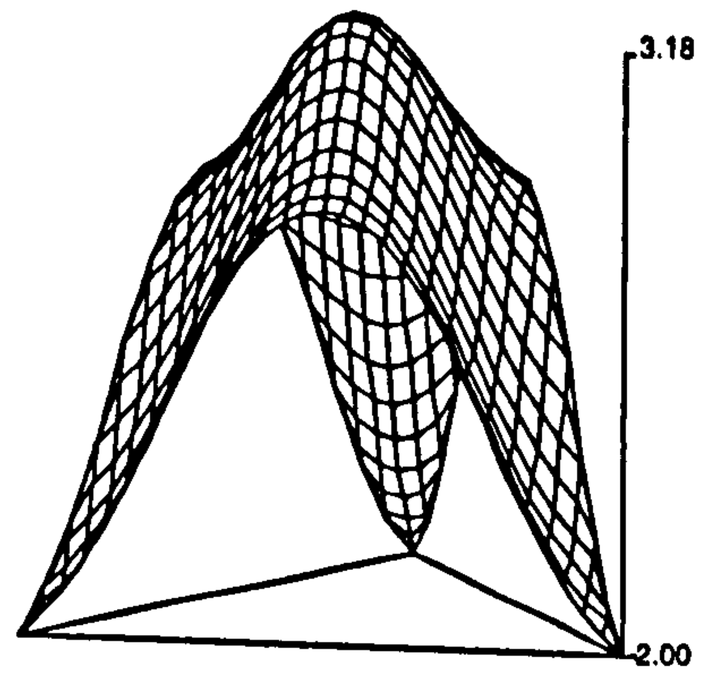
$\nu_1+\nu_2+\nu_3 = 1$. The vertical axis is the gain $\mathscr{R}$ of (104) obtained from observing that the three previous $\langle$product, environment$\rangle$ pairs have not failed. In the left plot we have $\langle C_1, C_2 \rangle = \langle .03, .16 \rangle$. In the right $\langle C_1, C_2 \rangle = \langle .38, .13 \rangle$.

Figure 10 shows examples where the situation is more complex with the remainder terms beginning to play a significant role so that we lose our approximate rotational symmetry of the plot. The left hand plot illustrates a case where the situation of even allocation among three previous $\langle$product, environment$\rangle$ pairs is intermediate (in terms of how much extra confidence it buys us in the current $\mathcal{A}_k$) to the cases of the same number of previous demands being either concentrated on a single previous $\langle$product, environment$\rangle$ pair, or being evenly allocated between two previous pairs. The right hand plot in Figure 10 is included to make the point that we are not suggesting that such odd behaviour of $\mathscr{R}$ will *necessarily* occur *everywhere* outside the domain of accuracy

[37] i.e. approximately circular contours



Left plot : $\langle a, b, Z \rangle = \langle 0.25, 20, 5.3 \rangle$;



Right plot : $\langle a, b, Z \rangle = \langle 1, 20, 10 \rangle$

Figure 10: Plots of $\mathscr{R}$ vs. $\nu_1+\nu_2+\nu_3=1$

of our approximation (106). Here, the remainder terms are large but we still do have relatively uncomplicated behaviour in the sense that there is a global maximum at $\langle \nu_i \rangle = \langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$, with $\mathscr{R}$ a decreasing function of r.

It is worth underlining that it is often the *interaction* between prior beliefs and amount of past $\langle$product, environment$\rangle$ data that determines which of the cases illustrated in these plots applies; rather than either one of these things alone. These findings about the effects of the shape of a Beta Prior_θ on the preference for an even or an uneven allocation of previous sequence observations between 2 or 3 previous sequences raise two interesting questions which in this paper we have not explored:–

- What, if anything, can we say generally about the preferred allocation among larger numbers $k-1$ of previous sequences?

- To what extent are the precise results we found here arbitrary, accidental consequences of the fact that we happen to have restricted our priors to the *Beta* family? Perhaps some of these results are in fact particular cases of effects that could be stated in a framework of more general geometric constraints on the Prior_θ distribution without the need to constrain Prior_θ to a particular parametric family?

5.3.5.2 Use of a Beta family for f_p

The Beta-family of distributions

$$f_p(p|\theta) = \frac{p^{a-1}(1-p)^{b-1}}{\beta(a,b)}, \qquad \theta = \langle a, b \rangle, \quad a, b > 0$$

is conjugate to both the binomial and the negative binomial (including geometric) distributions and also provides a unique representation[38] of each possible[39] (mean,standard deviation) pair for a random variable P confined to the interval $[0,1]$. If we use this as our f_p distribution family, we obtain a mixed process for the failures in each single sequence for which the probability of r failures in n demands is given by equation (53) to be

$$R|n,a,b \sim \frac{\binom{n}{r}\beta(r+a,n-r+b)}{\beta(a,b)},$$

obtained by integrating over p the joint distribution of equation (51) which would be

$$(R,P)|n,a,b \sim \frac{\binom{n}{r}p^{r+a-1}(1-p)^{n-r+b-1}}{\beta(a,b)}$$

in this case.

The likelihood (57) resulting from observation of k products operating in k allocated environments is

$$\langle R_i\rangle_{i=1}^{k}\left|(\langle n_i\rangle_{i=1}^{k},a,b)\right. \sim \prod_{i=1}^{k}\binom{n_i}{r_i}\frac{\beta(a+r_i,b+n_i-r_i)}{\beta(a,b)}$$

with

$$L_k(a,b) = \prod_{i=1}^{k}\frac{\beta(a+r_i,b+n_i-r_i)}{\beta(a,b)}$$

as an expression proportional to the likelihood of $\langle a,b\rangle$.

Having decided to investigate the Beta f_p, the choice of Prior_θ over Θ, the positive quadrant[40], remains problematic. In real life there would be an 'expert' from whom we would wish to elicit the distribution that truly reflects his/her a priori belief. This is not an easy task in such a complex model, and the expert may find it difficult to represent his/her beliefs in a distribution for $\langle a,b\rangle$. A way out of this difficulty is to assume that the expert is 'ignorant', and use that prior distribution which represents ignorance. Even this is a non-trivial task. As an example we consider the simple case of distributions uniform on some finite rectangle with sides parallel to the a and b axes,

$$\text{Prior}_\theta(a,b) = \begin{cases} \frac{1}{(a_2-a_1)(b_2-b_1)}, & \text{if } a_1 < a < a_2,\ b_1 < b < b_2 \\ 0, & \text{elsewhere.} \end{cases}$$

Firstly we can examine characteristics of the prior distribution (59) for P_k implied by these model assumptions,

$$P_k \sim \int_{a_1}^{a_2}\int_{b_1}^{b_2}\frac{p^{a-1}(1-p)^{b-1}}{\beta(a,b)}\frac{db\,da}{(a_2-a_1)(b_2-b_1)}.$$

---

[38] provided that limiting cases of the Beta parameters a,b are included

[39] i.e., all pairs in the closed half disk $\{(\mu,\sigma)\ ;\ \mu,\sigma \geq 0 \wedge (\mu-\frac{1}{2})^2+\sigma^2 \leq \frac{1}{4}\}$

[40] possibly extended to include points representing $a,b \to \infty$ with a/b constant, and $a,b \to 0$ with a/b constant, to include the all the limiting cases of the Beta family

The first and second non-central moments of $P \mid a, b$ are $\frac{a}{a+b}$ and $\frac{a(a+1)}{(a+b)(a+b+1)}$. These may be integrated analytically with respect to our ignorance $\text{Prior}_\theta(a, b)$ (first expanding in partial fractions with respect to b in the case of the second moment) to give the first two cases of equation (69). But the centrally important effect of our model is to represent the effect of observed failure behaviour on both the distribution of P_k, and perhaps even more of interest, the reliability function, or probability of a future period of failure-free behaviour of a given length. The prior reliability function is given from equations (73) and (78) by

$$\begin{aligned} \mathbf{P}[X_k > n] = \mathbf{E}[\mu_{0,n}] &= \int_{a_1}^{a_2}\int_{b_1}^{b_2} \frac{\beta(a, b+n)}{\beta(a,b)} \frac{db\,da}{(a_2-a_1)(b_2-b_1)} \\ &= \int_{a_1}^{a_2}\int_{b_1}^{b_2} \frac{b(b+1)\ldots(b+n-1)}{(a+b)(a+b+1)\ldots(a+b+n-1)} \frac{db\,da}{(a_2-a_1)(b_2-b_1)}, \end{aligned}$$

where the *first* failure of $\mathcal{A}_k$ occurs on the X_k^{th} demand.

Now to explore the effects of learning from observation we examine the realisations under these particular distributional assumptions of: firstly the posterior distributions for P_k given by equations (60–62); and secondly the predictions of X_k, the time to next failure of $\mathcal{A}_k$ using equations (79–81)[41]. In the most general case of arbitrary periods of observation of some finite number of previous sequences, each of the probabilities entailed by these questions takes the form of the ratio of a pair of integrals (over the chosen rectangle in the (a, b)-plane), where the integrands in the numerator and denominator are each equal to some product of terms of the form

$$\mu_{r,n-r}(a,b) = \mathrm{E}\big[P^r(1-P)^{n-r} \,|\, a, b\big] = \int_0^1 p^r(1-p)^{n-r} \frac{p^{a-1}(1-p)^{b-1}}{\beta(a,b)}\,dp = \frac{\beta(a+r, b+n-r)}{\beta(a,b)}$$
$$= \frac{a(a+1)\ldots(a+r-1)b(b+1)\ldots(b+n-r-1)}{(a+b)(a+b+1)\ldots\ldots\ldots\ldots\ldots(a+b+n-1)}$$

In practice, since this kind of inference is most likely to be called for in dealing with very high reliability systems, the values n_i of n used with these sequences are likely to be rather large, and the values of r are likely to be small, and ideally zero. So some very large products will be involved in the above term. We found that from the numerical point of view, both the asymptotic form of the log-gamma function, and also the Euler-Maclaurin series for sums of form

$$\sum_{j=0}^{n-1} \log\left(1 - \frac{y}{x+j}\right), \quad \text{where } 0 < y < x$$

were useful in approximating and bounding the integrals of these terms for large n. (See Appendix B.5 for details.) For the sake of illustrating the algebraic form of the formulas, however, we

[41]—given that we choose to concentrate on the no-failures case, for reasons of its interest as an upper bound on assurable reliabilities. To explore the case where past failures have been observed, we would simply use the obvious analogues of (79–81), derived similarly from (60–62) and (67)

give examples of the predictions of our model for hypothetical cases in which a very small number of observations have been seen. Suppose we wish to predict the probability that $\mathcal{A}_4$ will fail r times in its next 6 demands. In the absence of any knowledge of the past we obtain the distribution

$$R \sim \binom{6}{r} \int_{a_1}^{a_2} \int_{b_1}^{b_2} \frac{a(a+1)\dots(a+r-1)b(b+1)\dots(b+5-r)}{(a+b)(a+b+1)\dots\dots\dots\dots(a+b+5)} \frac{db\,da}{(a_2-a_1)(b_2-b_1)}$$

If we are now informed that $\mathcal{A}_4$ has in fact failed in the past 2 times out of 4, then our posterior distribution of P_4 is

$$P_4 \sim \frac{p_4^2(1-p_4)^2 \int_{a_1}^{a_2} \int_{b_1}^{b_2} \frac{p_4^{a-1}(1-p_4)^{b-1}}{\beta(a,b)} \frac{db\,da}{(a_2-a_1)(b_2-b_1)}}{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \frac{a(a+1)b(b+1)}{(a+b)(a+b+1)(a+b+2)(a+b+3)} \frac{db\,da}{(a_2-a_1)(b_2-b_1)}}$$

and our updated distribution for the number of failures in the next 6 demands on $\mathcal{A}_4$ is

$$R \sim \binom{6}{r} \frac{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \frac{a(a+1)\dots(a+r+1)b(b+1)\dots(b+7-r)}{(a+b)(a+b+1)\dots\dots\dots\dots(a+b+9)} \frac{db\,da}{(a_2-a_1)(b_2-b_1)}}{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \frac{a(a+1)b(b+1)}{(a+b)(a+b+1)(a+b+2)(a+b+3)} \frac{db\,da}{(a_2-a_1)(b_2-b_1)}}$$

If we retract the information about the past 2 out of 4 failures of $\mathcal{A}_4$ (i.e., suppose it has not been seen), and instead suppose that pairs $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$, have been observed to fail 0 times out of 2, 2 times out of 3 and 1 time out of 4, respectively, then our posterior distribution of P_4 is

$$P_4 \sim \frac{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \frac{p_4^{a-1}(1-p_4)^{b-1}}{\beta(a,b)} \left(\frac{b(b+1)}{(a+b)(a+b+1)}\right)\left(\frac{a(a+1)b}{(a+b)(a+b+1)(a+b+2)}\right)\left(\frac{ab(b+1)(b+2)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\frac{db\,da}{(a_2-a_1)(b_2-b_1)}}{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \left(\frac{b(b+1)}{(a+b)(a+b+1)}\right)\left(\frac{a(a+1)b}{(a+b)(a+b+1)(a+b+2)}\right)\left(\frac{ab(b+1)(b+2)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\frac{db\,da}{(a_2-a_1)(b_2-b_1)}}$$

Now, the updated distribution for R in the next 6 demands on $\mathcal{A}_4$ is

$$R \sim \binom{6}{r} \times$$

$$\frac{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \left(\frac{a(a+1)\dots(a+r-1)b(b+1)\dots(b+5-r)}{(a+b)(a+b+1)\dots\dots\dots\dots(a+b+5)}\right)\left(\frac{b(b+1)}{(a+b)(a+b+1)}\right)\left(\frac{a(a+1)b}{(a+b)(a+b+1)(a+b+2)}\right)\left(\frac{ab(b+1)(b+2)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\frac{db\,da}{(a_2-a_1)(b_2-b_1)}}{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \left(\frac{b(b+1)}{(a+b)(a+b+1)}\right)\left(\frac{a(a+1)b}{(a+b)(a+b+1)(a+b+2)}\right)\left(\frac{ab(b+1)(b+2)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\frac{db\,da}{(a_2-a_1)(b_2-b_1)}}$$

If this information about $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ is supplemented by the knowledge that $\mathcal{A}_4$ has failed 2 times out of 4 in the past, then the two corresponding updated distributions are

$$P_4 \sim \frac{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \frac{p_4^{a+1}(1-p_4)^{b+1}}{\beta(a,b)} \left(\frac{b(b+1)}{(a+b)(a+b+1)}\right)\left(\frac{a(a+1)b}{(a+b)(a+b+1)(a+b+2)}\right)\left(\frac{ab(b+1)(b+2)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\frac{db\,da}{(a_2-a_1)(b_2-b_1)}}{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \left(\frac{a(a+1)b(b+1)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\left(\frac{b(b+1)}{(a+b)(a+b+1)}\right)\left(\frac{a(a+1)b}{(a+b)(a+b+1)(a+b+2)}\right)\left(\frac{ab(b+1)(b+2)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\frac{db\,da}{(a_2-a_1)(b_2-b_1)}}$$

and

$$R \sim \binom{6}{r} \times$$

$$\frac{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \left(\frac{a(a+1)\dots(a+r+1)b(b+1)\dots(b+7-r)}{(a+b)(a+b+1)\dots\dots\dots\dots(a+b+9)}\right)\left(\frac{b(b+1)}{(a+b)(a+b+1)}\right)\left(\frac{a(a+1)b}{(a+b)(a+b+1)(a+b+2)}\right)\left(\frac{ab(b+1)(b+2)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\frac{db\,da}{(a_2-a_1)(b_2-b_1)}}{\int_{a_1}^{a_2} \int_{b_1}^{b_2} \left(\frac{a(a+1)b(b+1)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\left(\frac{b(b+1)}{(a+b)(a+b+1)}\right)\left(\frac{a(a+1)b}{(a+b)(a+b+1)(a+b+2)}\right)\left(\frac{ab(b+1)(b+2)}{(a+b)(a+b+1)(a+b+2)(a+b+3)}\right)\frac{db\,da}{(a_2-a_1)(b_2-b_1)}}$$

The above example is intended to provide an illustration of the general form of the results for this Beta $f_p(\cdot|a,b)$ case with prior Prior_θ uniform on a rectangle. Table 4 shows some results that are more representative of what we might see when dealing with real safety-critical systems. These illustrative numerical results are based upon the observation of three previous sequences, each for a period of 10^7 demands without a single failure. In Table 4 we can see how various different

Region of Uniform Prior				Given no Data		Given no failure of this ⟨product, envir.⟩		Given no failure of previous 3 ⟨product, envir.⟩s		Given failure neither of this nor of previous 3 ⟨product, envir.⟩s	
a_1	a_2	b_1	b_2	$\mathrm{E}[P_4]$	$R(10^7)$	$\mathrm{E}[P_4]$	$R(10^7)$	$\mathrm{E}[P_4]$	$R(10^7)$	$\mathrm{E}[P_4]$	$R(10^7)$
0	1	1	2	.2384	.6229E-1	.3966E-1	.9585	.1388E-1	.7498	.1047E-1	.9893
0	1	1	10	.1037	.6828E-1	.1577E-1	.9547	.5398E-2	.7499	.4062E-2	.9883
0	1	1	100	.2077E-1	.8048E-1	.3020E-2	.9469	.1019E-2	.7500	.7655E-3	.9862
0	1	1	1000	.3207E-2	.9877E-1	.4636E-3	.9355	.1556E-3	.7500	.1168E-3	.9831
0	2	1	2	.3692	.3114E-1	.3966E-1	.9585	.1388E-1	.7498	.1047E-1	.9893
0	2	1	10	.1781	.3414E-1	.1578E-1	.9547	.5398E-2	.7499	.4062E-2	.9883
0	2	1	100	.3833E-1	.4024E-1	.3020E-2	.9469	.1019E-2	.7500	.7655E-3	.9862
0	2	1	1000	.6091E-2	.4939E-1	.4637E-3	.9355	.1556E-3	.7500	.1168E-3	.9831
.01	.0101	10	10.1	.9990E-3	.8700	.9990E-3	.9931	.9990E-3	.8700	.9990E-3	.9931
0	b/999	1	1000	.5002E-3	.1824	.2056E-3	.9401	.9494E-4	.7545	.7593E-4	.9832
0	b/99999	1	1000	.5000E-5	.9689	.4947E-5	.9977	.4843E-5	.9703	.4791E-5	.9978
0	b/9999999	1	1000	.5000E-7	.99968	.4999E-7	.999977	.4998E-7	.99968	.4998E-7	.999977

.XXXXE-n means $0.XXXX \times 10^{-n}$

Table 4: Effect on Reliability Predictions of Observation of Non-Failure of Previous ⟨product, environment⟩ Pairs

assumptions for Prior_θ affect the strength of the inferences concerning a fourth sequence in the same family which can be drawn from this sort of evidence of high reliability of three previous, similar ⟨product, environment⟩ pairs.

All the results in the Table involve assuming uniform distributions over different regions of the $\langle a, b\rangle$–space. We have excluded values of b smaller than one, since these entail Beta distributions with infinite density at 1; but we have allowed values of a smaller than one, since infinite density at the origin seems plausible. The region in the positive quadrant where a and b are both large can also be ruled out, since any point here corresponds to a Beta distribution with very small variance—i.e. it implies that different sequences will have essentially identical probabilities of failure upon demand, which runs counter to the spirit of this whole exercise.

The first nine rows of the Table involve several rectangles of the kind described above. The ninth row shows a small rectangle, effectively approximating to a known point value for $\langle a, b\rangle$. Rows 10 to 12 show thin 'wedges' adjacent to the b-axis. The informal reasoning here is that it may be reasonable to believe a priori that the mean $\mathrm{E}[P|a,b]$ of the distribution of probability of failure on demand does not exceed a certain value $0 \leq \mathrm{E}[P|a,b] \leq M < 1$, say, and this is equivalent to the restriction to $\frac{a}{b} \leq \frac{M}{1-M}$. We used $M = 10^{-3}$, 10^{-5}, and 10^{-7}. Once again, all points in the wedge are given equal weight.

In the Table we show how 'the reliability' of a ⟨product, environment⟩ pair $\mathcal{A}_4$ is affected by the

type of evidence that could be available. For brevity here we have chosen to present the mean of the distribution of P_4, and the reliability function evaluated at 10^7 demands (i.e. the probability of surviving this number of demands), in each of the four cases: given no data; given only evidence of failure-free operation of this sequence; given only evidence of failure-free working of earlier sequences; and given both these latter items of evidence.

The most interesting and important results concern the different predictions of future operational behaviour, expressed as the probability $R(10^7)$ of surviving 10^7 further demands without failure: the information from the perfect working of previous sequences makes only a modest contribution to our confidence in the current sequence when compared with actual evidence of failure-free working during that sequence itself (compare columns 8 and 10). Thus when we only have evidence from the previous $\mathcal{A}_i$, $1 \leq i \leq 3$, although this is of extensive perfect working for each, it only allows us to claim, in the case of the rectangular priors, about 0.75 probability of similarly extensive perfect working (i.e. surviving 10^7 demands) for the new sequence[42].

The evidence from previous perfect working during the *same* sequence, however, is more informative. It allows us to be much more confident that this product will work perfectly in this environment in the future: the probability of it surviving 10^7 demands, given that it has already survived 10^7 demands, exceeds 0.9 in all cases.

On the other hand, the small increase in confidence that comes from experience of previous sequences may be useful in the case of safety-critical systems, especially as it is likely to come with little or no cost to developers. Thus, in the first row of the Table, the *a priori* belief of the 10^7 demand survival is .062, this increases to .96 after we have actually seen the (product, environment) survive 10^7 demands, and to .99 when we are told, in addition, that three other (product, environment) pairs have also survived 10^7 demands. Putting it another way, this evidence of survival in previous sequences has reduced the chance of a failure in the next 10^7 demands by a factor of 4 (from .04 to .01) compared with the result based only on the evidence from operational experience of this sequence.

We have shown the columns for the means of the various distributions for P_4 mainly as a warning that these can be misleading if used to represent 'the reliability' of the pair $\mathcal{A}_4$. Thus the mean probability of failure on demand can be quite large (0.24 in the first line prior distribution), but still the chance of surviving 10^7 demands may be non-negligible (0.063 in this case). The informal reason is that the distribution is such that the mean is not a good summary statistic, and in particular cannot be used in a geometric distribution to approximate to the more complex model that applies

[42] We conjecture that some limiting result may be indicated here : perhaps the probability that sequence $\mathcal{A}_k$ will survive its first X demands, given that $k-1$ previous sequences have done so, tends to $(k-1)/k$ as $X \to \infty$.

here.

In fact, decreasing values of $\mathrm{E}[P_4]$ do not necessarily imply increasing chance of surviving 10^7 demands, as might naively be expected: see, for example, columns 7 and 8 of rows 1 to 4. Imagine that we have two experts, let us call them James and Peter, represented by two different prior distributions (rows of the Table), who observe the system to survive for 10^7 demands. They are then asked to tell us how reliable the system is. If the question is posed as 'what is the mean of P_4?', then James is more optimistic than Peter; if, however, the question is posed as 'what is the chance of surviving a further 10^7 demands', Peter is more optimistic than James. Such (only apparent) paradoxes underline the importance of using the right formulation for our purposes when we ask questions about the reliability of a system.

5.3.6 Some Remarks about Expressing Reliability in Terms of the Similar Products Model Structure

The use here of Bayesian reasoning, representing the modeller's uncertainty about an 'unkown, true reliability' parameter such as P_i raises some interesting features of the general question mentioned in §2.3.5. This question concerns the strict interpretation of a concept such as 'reliability' in terms of the precise mathematical model structure. We have taken the decision to extend an originally Bernoulli Trials process model with simple geometrically distributed time-to-failure distributions by including explicitly within the mathematics a probabilistic representation of our own subjective uncertainty about each Bernoulli trials parameter P_i. One of the consequences of this decision is that our subjective uncertainty about the P_i may never now remain static, so long as some form of observation is allowed to take place. Hence, as has been apparent in the formulas derived above (e.g. (79–81)), our predictive distributions of time to next failure are no longer restricted to the geometric family. Then if we retain the notion of geometric time to failure as a psychological standard of comparison, we must be careful when we speak of 'a 10 to the minus 5 system' to be clear whether we intend 'a system whose probability of failure on *the next* demand is identical to that of a geometric random variable with parameter 10^{-5}'; or 'a system whose MTTF is that of a geometric random variable with parameter 10^{-5}'; or perhaps 'a system whose median (or, say, upper 99.9 percentile) is that of a geometric random variable with parameter 10^{-5}'; etc.

It follows from (67) that the Bayesian predictive distribution of the process of future failures of a particular sequence, based on our model, will always take the form of a mixture of Bernoulli trials process distributions. Such mixture processes are *exchangeable*[43]. (Conversely [23, p217] states that

[43] See footnote 9 on p117.

there are no other exchangeable, boolean valued, infinite random sequences than those obtained by mixing Bernoulli trials processes.) A few properties of these mixture processes were given earlier in equations (52–55), with $f_p(\cdot|\theta)$ playing the role of mixing distribution in these equations. Since these mixed processes form a more general class than the class of Bernoulli trials processes used as a model for each $\mathcal{A}_k$ *given its parameter* p_k, the theoretical possibility is introduced that it may also require some care to *compare* two different predictions which may emanate from our model (i.e. resulting from two different hypothetical findings from observation). How do we state unambiguously that one observation scheme gives rise to a prediction of 'higher program reliability' than another? If we were dealing with pure Bernoulli trials process predictions, then we would be able to say, for two predictions with parameters (i.e., per-demand failure probabilities) say π_1 and π_2, with $\pi_1 < \pi_2$, that prediction 1 predicts 'higher reliability' than prediction 2 in every possible sense: mean time to failure, median time to failure, failure rate, reliability function, etc. On the other hand for many exchangeable process predictions such as will be produced by a Bayesian analysis of our model, the mean time to failure does not exist (is infinite). Also we may well find that the median time to failure of prediction 1 may be greater than the median for prediction 2, whilst the order of say the 75%-iles could be reversed, i.e. the two reliability functions[44] obtained from two different observation schemes, such as two of equations (79–81), as functions of[45] n'_k, could conceivably cross over, so that prediction 1 asserts better short term reliability than prediction 2 but the comparison might turn out to be reversed for longer term reliability predictions.

We might choose to compare predictions based on different observation assumptions in terms of their instantaneous 'reliability' measure, given (for the observation scenarios we have considered explicitly) by the $n'_k = 1$ case[45] of expressions (79,81). However, in doing so we should bear in mind that such a measure does not necessarily tell us which prediction has the highest up-to-date 'mission survival probability'[44] for a mission of a given length[45] $n'_k > 1$. It may be possible to overcome some of these difficulties by suitable restrictions upon the mathematical forms of Prior_θ and f_p, but these would need to be 'obviously reasonable' in their own right. Clearly it would be wrong for example to force an unreasonable (i.e. not believed) prior upon a human expert.

---

[44] Reliability function (or 'mission survival probability') is the r'_k=0 case of (79,81) (or r_k=0 case of (78,80)), thought of as a function of the mission length n'_k (or n_k).

[45] or n_k in the notation of equations (78,80)

Chapter 6

Summary of Main Conclusions

This thesis develops some enhancements to existing techniques for software reliability prediction, and predictive quality assessment. Firstly, in the frequently treated case in which we use as sole source of data the failure vs. execution time history of a single product operating within a stable environment (which may incorporate debugging activity), we have developed a method of transferring to the context of coarse failure-count data the u-residual-based *recalibration* technique studied previously [2, 6, 10, 11, 59, 58] within the context of continuous predictive c.d.f.s for prediction of software inter-failure time sequences. With this motivation, a method of defining a "u-plot" for any PFS applied to a scalar random process in discrete time has been given. This definition extends the usual definition given in [2] in two senses:–

(i) by applying to the case of a general predictive distribution, rather than being restricted to continuously distributed predictions only;

(ii) by introducing flexibility into the definition of the u-plot using (a) unequal weights w_i for different observed u_i and (b) bounds ϵ and $\frac{1}{\delta}$ on the derivative of the fit, if smoothing is employed.

This "Modified U-Plot" reduces to the previous case on setting $w_i = w_j$ for all i, j and $\epsilon = \delta = 0$. Apart from this widening of the scope of application, the purposes of the modified plot are as for the familiar plot in [2] and [59]:–

(i) to assist in the analysis of predictive performance. (There is probably little point here in smoothing or in using unequal w_i.)

(ii) to produce a means of recalibration for any PFS.

We have investigated the performance of the resulting recalibrated PFS experimentally for prediction of software "failure-count" sequences—for which the predictive c.d.f.s are purely discrete. There is evidence for the following conclusions concerning the performance of the recalibration procedure on the data sets and PFSs examined:–

(i) The recalibrated PFS is almost invariably "better calibrated" than the raw PFS in the sense that it produces a modified u-plot (unsmoothed, with weights w_i all equal) which is closer to the 45°-line.

(ii) Assessed in terms of other measures of comparative predictive performance, the recalibration procedure *sometimes* gives a dramatic improvement in quality of prediction. In other instances, the results are less conclusive in favour of recalibrated predictions. Greater improvement in prediction seems more often to result from recalibrating in cases where the modified u-plot of the raw predictions deviates further to one side of the 45°-line, (say for K-dist$\geq$ .15, as a rough guide).

(iii) Variability in the weights w_i was investigated here only by allowing them to decrease exponentially for less recent u_i, thus introducing a "decaying memory" effect in the prediction of the $\{U_n\}$ sequence. As would be expected, using too small a scale factor, r, results in the recalibrated predictor becoming swamped by the 'noise' in the last few observed failure counts with a consequent degradation in predictive performance. However, it seems generally to be the case that the value for r which gives the best recalibrated predictor is strictly less than 1, (as assessed by K-dist and PLR—The χ^2 distance measure provides less support for this conclusion.) The value of this best r seems to depend on the other chosen characteristics of the recalibrator, i.e. on whether a smoother is used and on the values selected for the bounds ϵ and δ.

(iv) When using a smoothed recalibrator, the introduction of small positive values for either or both of the bounds ϵ and δ, generally improves the predictive quality as measured by the χ^2-distance or the PLR, particularly when a smaller r is used. No particular values for these bounds appear to give a performance which is best consistently over different failure-data sets and raw PFSs. It is sometimes found that, without the incorporation of a positive value for ϵ in the smoother, events may occur (fairly early in the sequence of observations) to which the recalibrated predictor assigns a predictive probability of zero. This, of course, results in a log prequential likelihood value of $-\infty$.

In this thesis, we also have discussed some approaches to augmenting the sources of data that might be used by mathematical models for software reliability assessment and prediction. One major motivation for seeking some formal approach of this kind is to make the process of assessing highly reliable systems, such as for example some safety-critical systems, more open to analysis. Currently, particularly in those cases where complex software is involved, such reliability assessments have a high degree of informality and rely on expert judgement. In these cases it may be difficult to analyse in any precise manner how the final judgement has been reached, and much has to be taken on trust based on the achievements, qualifications, reputation and experience of the experts involved. There is some evidence of experts being unduly optimistic about their judgemental abilities [33]. A more formal means of argumentation would, provided it remained accessible to domain experts and was able to take some account of their assessments, have the advantage that assumptions and reasoning processes would become visible and could be recorded and later scrutinized and checked by others. With this aim, we have firstly proposed that some existing general purpose regression models which have found diverse application elsewhere might be employed to extend the scope of data input to software reliability predictors. Three different categories of application have been distinguished in which "explanatory variables" are used as data supplementary to observed failure histories vs. time. Of these three, we believe that one holds the greatest promise of yielding positive practical results for at least some instances of the software reliability prediction problem. This would involve the incorporation of data capturing variation in execution environment for a single software product, either between installations or over time at each single installation. The desire to apply such models in this application so as to validate or refute them empirically is frustrated by the lack of data. Two alternative explanations for this lack of data are: It may reflect a realistic perception that our understanding of and ability to predict software reliability cannot be improved by the availability of such data sufficiently to recoup the costs involved in its collection and analysis; On the other hand, the absence of data might result from a failure to appreciate its potential value, or uncertainty as to how best to proceed with its collection and analysis.

Our second proposed approach to modeling data other than the failure vs. execution time history of a software product operating in a particular environment treats a small part of this general problem of incorporating diverse empirical evidence by providing a representation, and means of composition, of just one important type of additional evidence that is commonly used to make claims for software reliability: evidence from previous experience of testing other pairs $\langle$product, environment$\rangle$ that are held to be 'similar' to the pair for which predictions are now specifically required. Whilst we make no great claims for the realism of the precise assumptions of what we have called the 'similar

products' model, our analysis does indicate the way in which a formal model of this kind could be used to question whether an optimistic conclusion drawn from past experience might be ill-founded. Essentially, if you were to claim that great trust could be placed in a *particular system* because of past experience of *other environments or systems*, you would have to justify this by trying to claim that your independence assumptions and prior distributions were reasonable. Our 'similar products' model depends upon the reasonableness of our probabilistic representation of a notion of statistical 'similarity' between different demand sequences. In this we have employed the idea of a random sample of unobserved per-demand failure probabilities P_i drawn independently from an *unknown* parent distribution – about which we have Bayesian prior beliefs – as a proposed method to formalize the kind of extremely informal claims that experts make when they argue that the failure behaviour of one demand sequence can be used as a means of inferring the likely behaviour of another. Justification of such assumptions of similarity in particular cases is, of course, outside the direct scope of our studies—presumably it will come, in the case of software, from knowledge of the application domain (the problems being solved were similar), the development process (the methods used were similar), the design teams (they were the same or of comparable competence), etc. As far as prior distributions are concerned, it is clear that some of the examples we have used could be said to be 'unreasonable' in the sense that they represent beliefs about the reliability, prior to seeing any evidence, that are very strong. In presenting this 'similar products' model, we have produced some purely illustrative numerical examples. Clearly further work is needed to identify classes of practically 'plausible' prior distributions.

Despite these caveats about assumptions, we believe that our 'similar products' model might be used to provide a curb on the enthusiasm of experts: specifically, the use of 'similarity' arguments to make stronger claims than would be warranted via the model should be treated with suspicion.

The 'similar products' model illustrates that comparisons between different predicted reliabilities can be highly dependent upon the precise way in which those reliabilities are formulated. This observation resembles similar conclusions obtained in a different context, concerning stopping rules for software testing [63].

Chapter 7

Suggestions for Further Work

With regard to software reliability prediction based purely on past reliability behaviour of a single ⟨product, environment⟩ combination, the 'Modified U-Plot' work contained in chapters 3 and 4 of this thesis stimulates the following questions and ideas for potential improvements and further investigations:–

(i) For the case of simulated failure data, the results presented in this thesis were each obtained by the simulation, using pseudo-random numbers, of only *one single* failure count sequence from the given reliability growth model, and are thus subject to the effects of random variation. The use of simulated data from one model as input to another, whilst less convincing in one obvious sense than the application of models to the prediction of *real* software failure data, nevertheless does provide an opportunity to remove this source of uncertainty. More exact statements of relative performance of the various recalibrators could be obtained by analysing a large random sample of data sets generated independently from the same model, (i.e. same joint distribution for the failure-count sequence). This would be particularly interesting in cases where the "true" PFS is used as a standard of comparison. For example, large-sample mean and standard-error plots of the discrete log prequential likelihood ratios taken over a batch of process realisations based on different seeds to the pseudo-random number generator would be capable of providing more definite conclusions on the relative performance of different choices of the recalibrator parameters r, ϵ, and δ over various time sub-ranges of a simulated failure data process.

(ii) Again in the case specifically of *simulated* failure-count (or inter-failure time) data, it would be interesting to experiment with recalibration of the "true" PFS. Presumably no improvement in

predictive quality should be achievable by this means. To confirm that there is no improvement would be a useful validation of our predictive quality assessment methodology. It would also be of interest to examine the nature and extent of any degradation in performance through recalibration here. (The previous point about the value of a statistical analysis of the results of repeated runs using different random number seeds is clearly relevant again here.)

(iii) It seems likely that improved prediction would be obtained by optimising the choice of ϵ, δ and r by perhaps maximising past prequential likelihood. This idea is supported particularly by the fact that, although values $\epsilon > 0$, $\delta > 0$, and $r < 1$ usually seem to result in improved prediction, the experimentally optimal values of those tried in Chapter 4 for these parameters (see Table 3) vary from one data set to another. The only obstacle in the path of such a refinement would be that of computational load, but this is not a serious restriction in many cases—for example, all three of the raw PFSs used in the numerical work of Chapter 4 rely on "ML plug-in" prediction which reduces numerically, in these cases, to a search for the unique zero of a single monotonic scalar function over an interval. (The monotonic function is in each case explicit and in closed form.)

(iv) Perhaps, without going to the lengths suggested in (iii), there is an argument for allowing ϵ and δ to be decreasing functions of n (the index of the current observation). The reasoning behind this suggestion is explained on p74. (See footnote 33.)

(v) Any new measures of predictive quality that might be found for discrete (or mixed) PFSs would assist in assessing these recalibrators.

(vi) In all of the numerical examples of Chapter 4, the failure-count data used was in fact produced from a complete set of explicit inter-failure times (whether real or simulated). Thus, unrealistically, it was possible to *choose* the failure-count time intervals—and these were taken to be of *equal length*. It is not clear what the effect on the performance of the recalibration procedure would be if these intervals varied in length during the period of observation of reliability, as they are likely to do in practice when only failure-count data is being recorded. (For example, the elapsed CPU time of a computer system between the taking of successive cumulative failure-count recordings may well be variable.) It is suspected that the effect is likely to be detrimental to the performance of the recalibrator. This is because, for a given PFS and failure process, the true distribution for U_n conditional on observed failure counts, m^{n-1}, will in general be affected by the length d_n of the interval during which the failure count M_n is to be observed. Thus, in transferring from inter-failure to failure count prediction, an

additional source of variation in the conditional distribution of U_n has been introduced, wherever the intervals vary in length[1]. The recalibration procedures described above include no means of taking account of such an effect. The form of this relationship between interval length and distribution of U_n may be difficult to analyse, being determined by details of the PFS and of how it takes account of interval length in its prediction of failure count. Thus a further difficulty is introduced into any attempt to produce a predictive distribution for U_n empirically on the basis only of past observed behaviour of u^{n-1}, (which is the basis of the recalibration technique). To take a much simplified case, consider a failure process which is known to follow an HPP law[2] with rate parameter say μ, and assume that it is predicted using a PFS $\mathcal{P}$ which is the ML plug-in HPP PFS for the failure-count process determined by a given sequence of observation intervals. Then it follows that, in the notation of Chapter 3, the predictive c.d.f., F_n^M, (i.e. the conditional distribution of M_n given $M^{n-1} = m^{n-1}$ under $\mathcal{P}$) is Poisson with parameter $\widehat{\mu} d_n$, say, where d_n is the length of the n^{th} interval during which M_n failures will be counted, and $\widehat{\mu}$ is the ML estimate, based on the observations m^{n-1}, of μ, the HPP rate parameter. Given our artificial assumption that the true distribution of the failure process is HPP[μ], it follows that under the consequent distribution $\mathcal{Q}$ for the failure-count process $\{M_n\}$, the true conditional c.d.f., H_n, say, of M_n given $M^{n-1} = m^{n-1}$ is Poisson with parameter μd_n. It can be easily shown using definition (27) that the true conditional c.d.f., $G^{S^{\mathcal{Q}}}$ say, of U_n given $M^{n-1} = m^{n-1}$ is then the piecewise linear function defined by the values

$$G^{S^{\mathcal{Q}}}(u|m^{n-1}) = \begin{cases} 0, & \text{if } u \leq 0; \\ 1, & \text{if } u \geq 1; \\ H_n(m|m^{n-1}), & \text{if } u = F_n^M(m|m^{n-1}),\ m = 0, 1, 2, \ldots \end{cases}$$

at its vertices. The dependence of this c.d.f. on the length, d_n, of the next failure-count interval is illustrated for a particular example in Fig. 11. Here, the true c.d.f. of U_n given $M^{n-1} = m^{n-1}$ is plotted for interval lengths, $d_n = 100, 200, 300, 600, 1000$. In this figure, the true rate of the failure process is $\mu = 10^{-2}$, and the ML estimate based on m^{n-1} is assumed to be $\widehat{\mu} = 1.2 \times 10^{-2}$. No attempt has been made in this thesis to find a way in which the procedure for recalibrating a particular raw PFS for a failure-count process could be changed in order to sensibly allow for such unequal duration of the counting intervals. A crude solution might be to use smaller values for the weights w_j the greater the discrepancy between the the length d_j of the time interval during which m_j was counted, and the length d_n of time allowed

[1] With any luck, in practice this problem might perhaps be to some extent alleviated by the probable use (by the collectors of failure-count data) of shorter failure count interval lengths d_n during periods when the software failure rate is higher.

[2] known to us, for the sake of this example, but not known to the person making the predictions

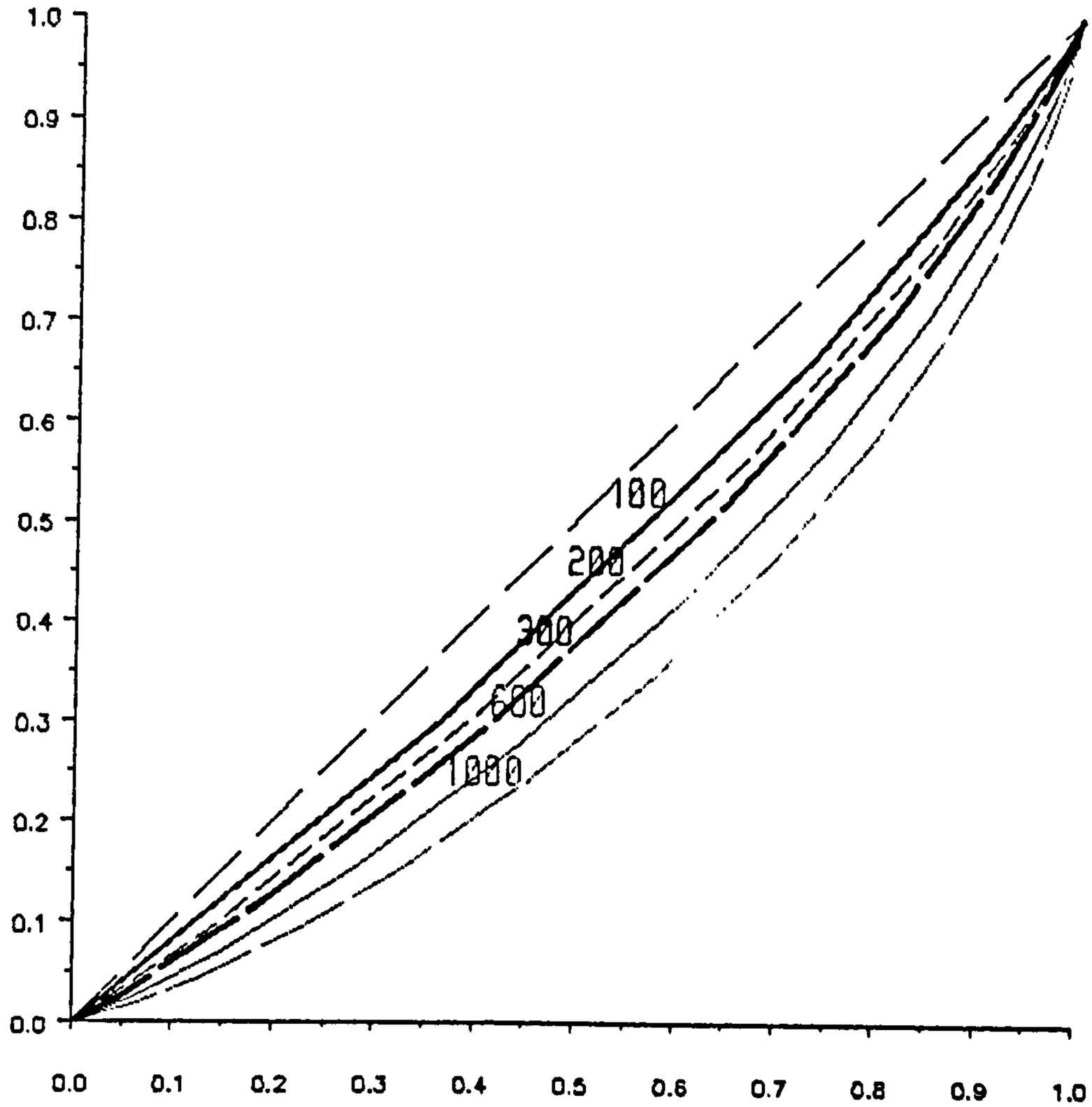


Figure 11: True one-step-ahead conditional c.d.f. of U_n in terms of interval length d_n

for the count M_n currently being predicted. It seems plausible that there might be a more sophisticated, practical way of taking into account the effect of interval length in forming a plot with which to recalibrate, though preliminary thoughts along these lines seem to suggest the question of whether, in the case of failure-count processes, the next count M_n—depending as it does on a given interval length—is really the quantity of greatest interest for prediction.

(vii) Chapter 3 is focused mainly on the problem of one-step-ahead prediction of the observed scalar sequence itself. This problem can be extended in at least two ways: (a) to longer term prediction, and (b) to prediction of other related quantities. It seems likely that some kind of similar recalibration approach to improving raw predictions might prove useful in either of these two larger problem areas. See §3.7 for a suggested approach to such extensions. In the special context of failure-count observation, examples of such extensions include prediction of the continuous variables: mean-time-to-failure, failure-rate, and reliability function, either immediately or after a specified further time has elapsed, as well as quantities such as time required to achieve a target reliability. For further details on some of these alternative quantities for prediction (discussed in the context of a particular model) see for example [53, §3 pp316-319].

(viii) The recalibrator described in §3.6—including the two extensions, (weights, w_i, and bounds, ϵ & $\frac{1}{\delta}$)—was defined for any scalar PFS and could be applied, in particular, to the problem of inter-failure time prediction. That is, investigations of the kind reported in for example [6, 11], and [59] could be repeated on the same data sets after the incorporation of either or both of these refinements in the recalibrator.

(ix) The following suggestion is unrelated to the main purpose of the current thesis but arises on noting the degree of success achieved by the use of decreasing weights $\{w_i\}$ in the definition of the recalibrator. The use of these weights can be viewed simply as an ad hoc method of making each prediction of the PFS more responsive to observations in the recent than in the distant past—in the belief that the modelling assumptions made are not accurate enough to hold well over longer time spans. Perhaps something similar could be done at the stage of raw PFS, rather than being left until the recalibration phase. For example, could an ML-plug in PFS be improved by changing the form of the objective function, (log likelihood), from

$$\ell(\theta) = \sum_i \log p_i(x_i|x^{i-1};\theta)$$

to

$$\ell(\theta) = \sum_i w_i \log p_i(x_i|x^{i-1};\theta),$$

where $w_i \geq 0$? Since the usual large-sample asymptotic optimality properties of ML estimators are unlikely to be very applicable in the case of software reliability growth, it does not seem clear that any important theoretical property would be lost by such a change, the value of which would be assessed in terms of performance measures of the resulting PFS.

(x) As another aside, we mention an alternative and simpler approach to the problem of forming predictions from failure-count observations and of recalibrating those predictions: The suggestion, probably more valid when the failure-count time intervals are many and short, would be simply to invent individual inter-failure times by distributing the failures counted, using some suitable rule, throughout each failure-count interval. Inter-failure time PFSs could then be applied to this data using familiar techniques (see [2, 6, 10, 11, 59, 58]), which could include recalibration methods.

(xi) Related to the previous suggestion, there is a well known general iterative statistical estimation method, known as the *Expectation-Maximisation* or EM-algorithm, designed for fitting a model when part of the usual observation has been lost. Perhaps this could be applied when faced with failure-count data to repeatedly:-

1. estimate (using the theoretical mean) that part of the log-likelihood function which involves precise inter-failure times, given model parameter estimates and observed failure-counts and;
2. then use existing ML software for fitting continuous inter-failure time model parameters which maximise the resulting likelihood; and so on, back to 1.

The main proposals for further work arising from the study of methods of incorporating other information in software reliability predictions are contained in §§5.2.1.3,5.2.1.4 & 5.2.3. The first is to produce more detailed data requirements and data collection procedures for characterising software operational environments of software which runs either at multiple installation sites, or under measurably time-varying conditions at a single site. The second is to acquire some such data, either by deliberate experimentation on a small scale, or from cooperative users and maintainers of real-world application software, and to experiment with the model fitting, identification and checking procedures discussed in §2.7.3.

Regarding the 'similar products' model of §5.3, further work is needed to identify classes of 'plausible' prior distributions, even for the case in which the expert professes 'complete prior ignorance'. For example, in §5.3.5.1 or §5.3.5.2, rather than addressing the raw $\langle a, b \rangle$ parameters, it may be easier for the subject to think in terms of a reparameterisation – the mean and coefficient of variation are possibilities. Another area of future work concerns the further exploration of the impact of different kinds of evidence upon the conclusions. For example, in our examples in §5.3 we concentrated most of our attention on what is in many respects the most interesting case : that of complete perfection of operation of the previous sequences. This is the best news that it is possible to have, but it would be interesting to look more carefully at some cases where there have been failures in the earlier sequences. In particular, there are some interesting 'calculus of variations' style mathematical optimization problems that arise – for example what are the extrema of the conclusions concerning the reliability of a given product within the family as a function of shape of prior distribution? One might fix certain basic properties of the prior distribution, such as the mean initial value of P_i, or the variance of this distribution, or both. One might constrain to prior distribution unimodality or to prior distributions having continuous pdf on $[0, 1]$. Taking such an assumption as a constraint for mathematical optimization problems, what happens as you vary the shape of the 'prior beliefs entity' $\langle \{f_p(\cdot|\theta)\,;\, \theta \in \Theta\}, \text{Prior}_\theta \rangle$ within the abstract space of functions in which it lies? By posing and solving mathematical problems such as this, it might become possible to gain insight into the limits of the possible influence that prior belief can exert over the model conclusions emanating from our 'similar products' model.

Appendix A

Graphical Output from Data Analysis

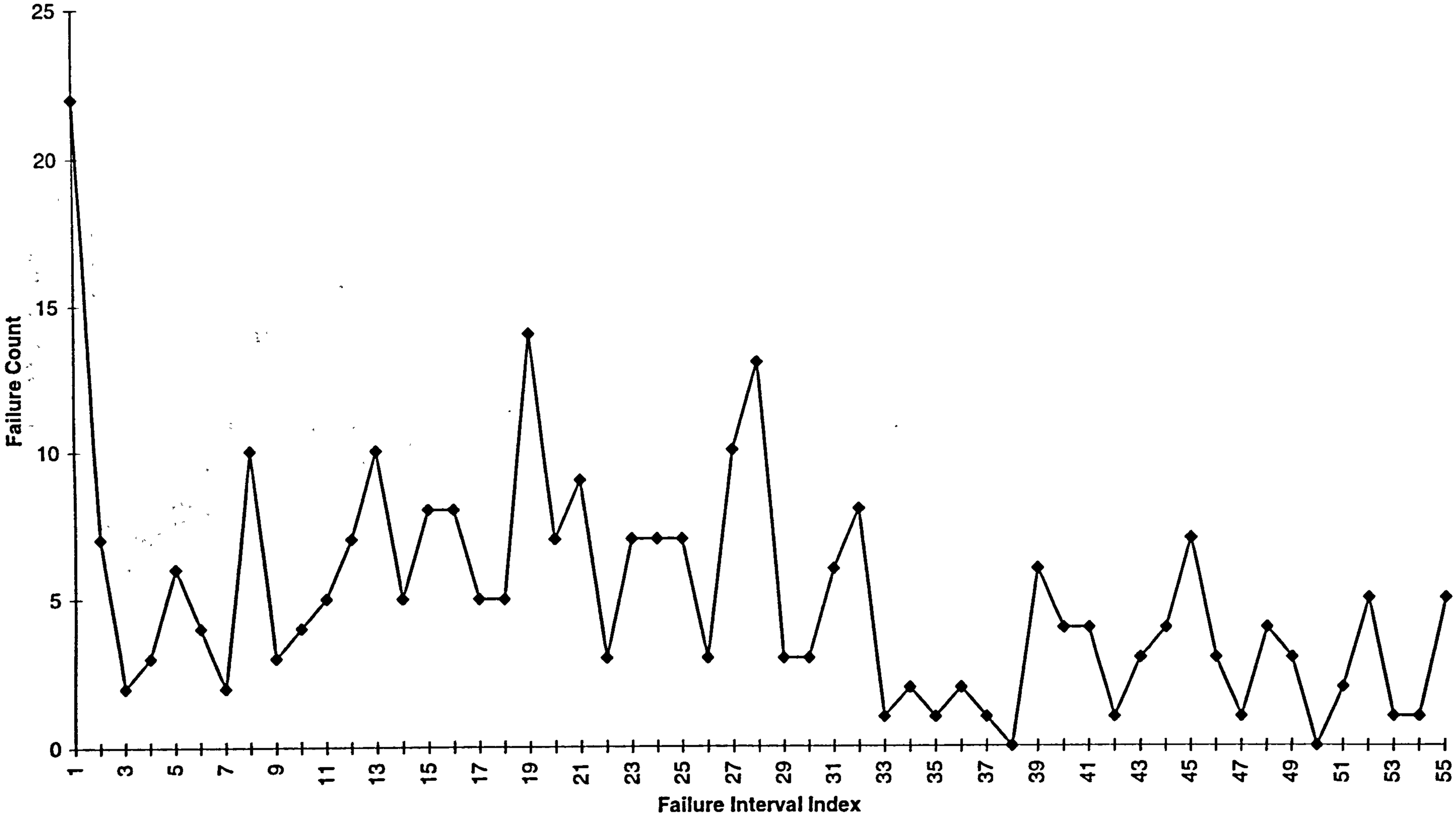

Failure Count Data: SS3 - Interval Length 1,000,000 seconds
Failure Count
25
20
15
10
5
0
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
Failure Interval Index

Figure 12

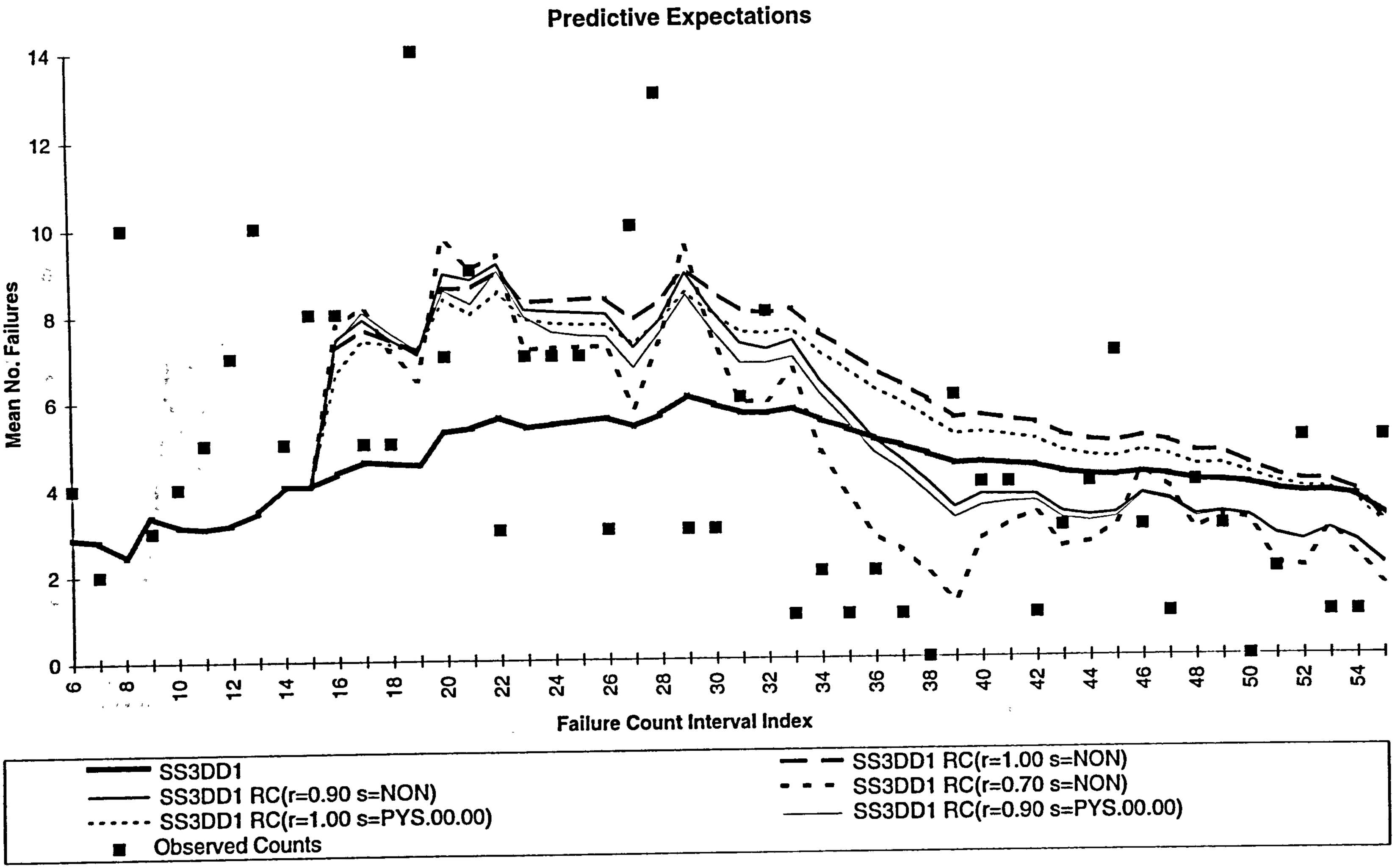

Predictive Expectations
Mean No. Failures
0
2
4
6
8
10
12
14
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
Failure Count Interval Index
SS3DD1
SS3DD1 RC(r=0.90 s=NON)
SS3DD1 RC(r=1.00 s=PYS.00.00)
Observed Counts
SS3DD1 RC(r=1.00 s=NON)
SS3DD1 RC(r=0.70 s=NON)
SS3DD1 RC(r=0.90 s=PYS.00.00)

Figure 13a

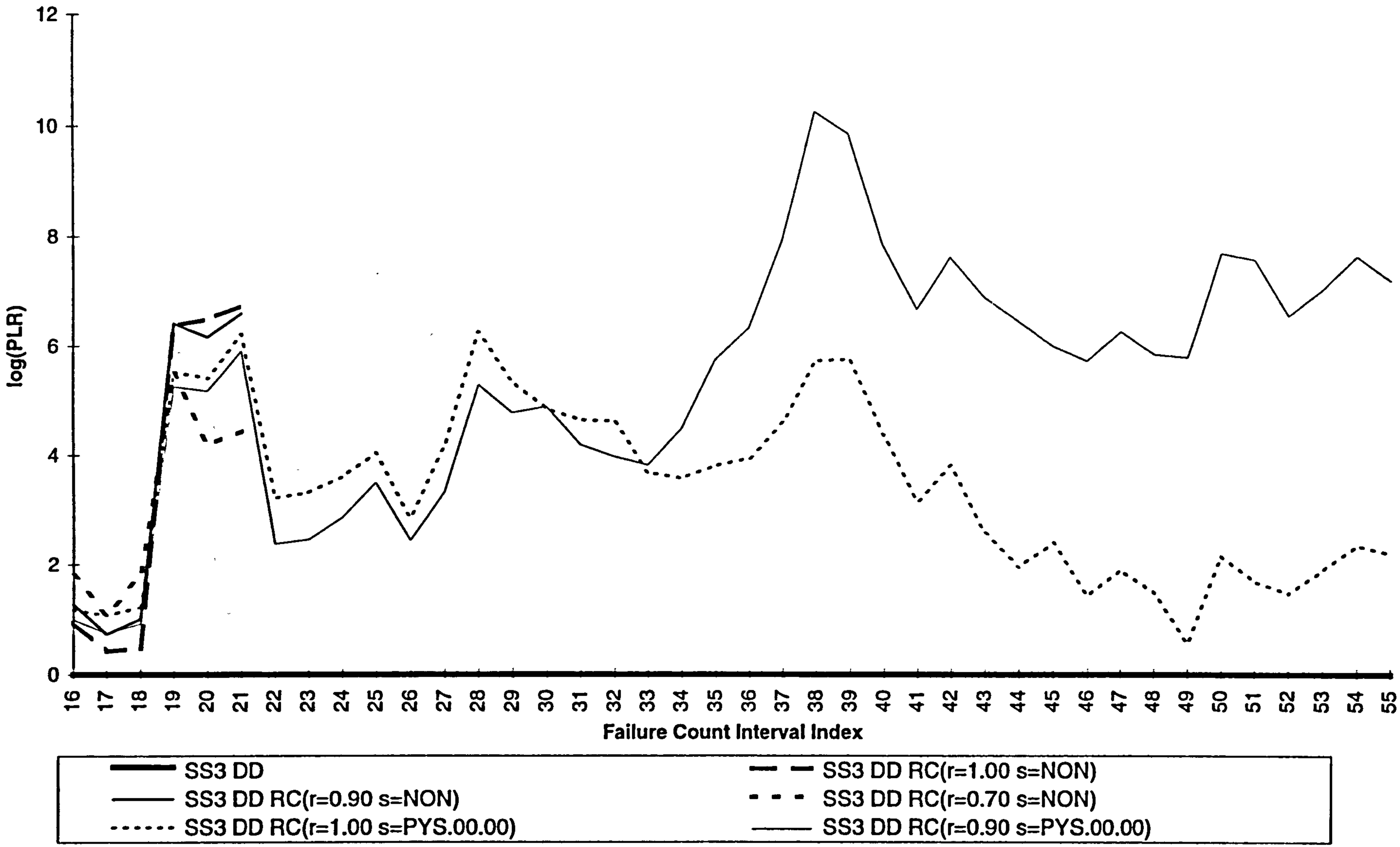

Discrete Log PLR -- vs. SS3 DD
log(PLR)
12
10
8
6
4
2
0
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
Failure Count Interval Index
SS3 DD
SS3 DD RC(r=0.90 s=NON)
SS3 DD RC(r=1.00 s=PYS.00.00)
SS3 DD RC(r=1.00 s=NON)
SS3 DD RC(r=0.70 s=NON)
SS3 DD RC(r=0.90 s=PYS.00.00)

Figure 13b

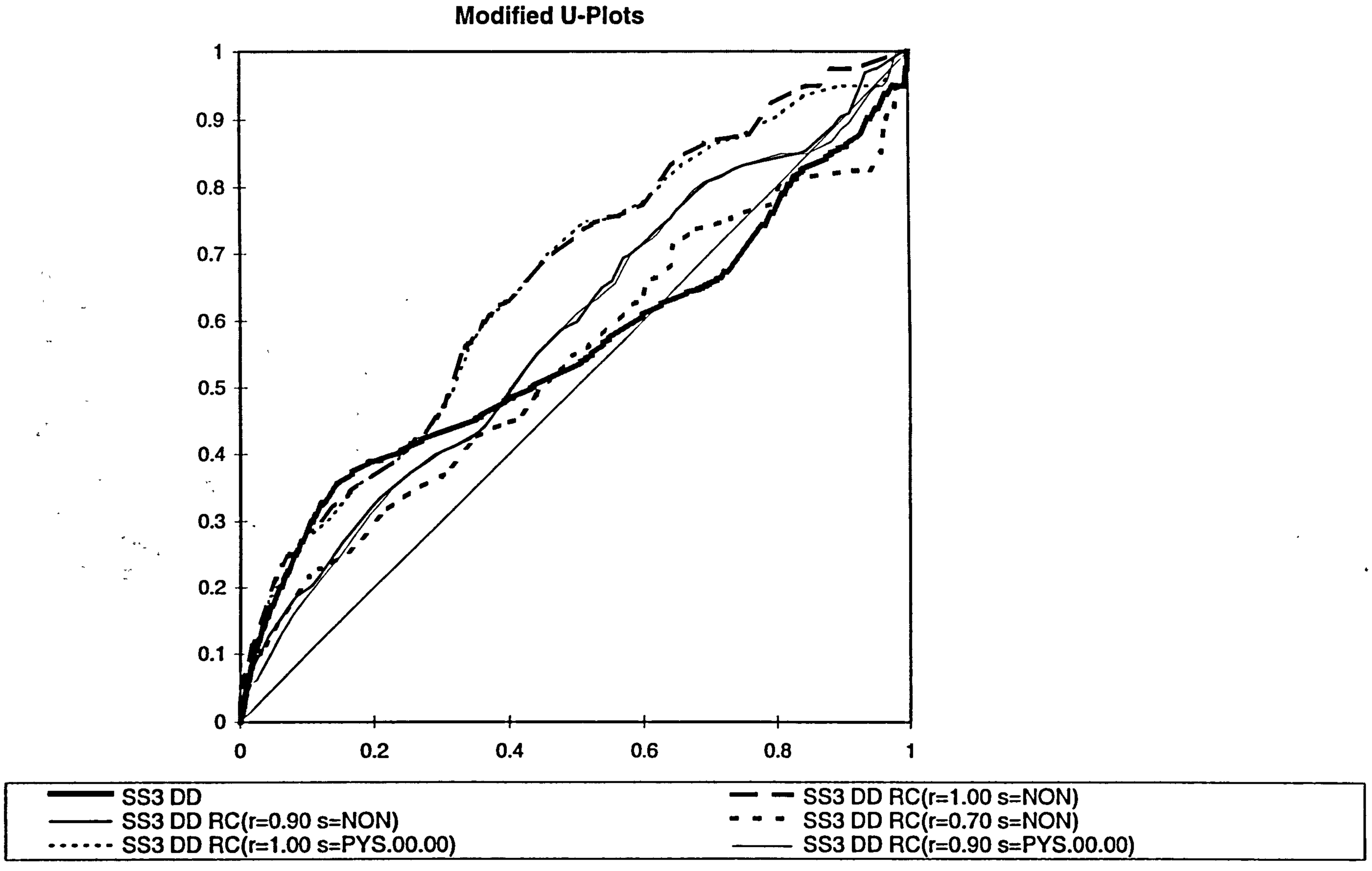

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
SS3 DD
SS3 DD RC(r=0.90 s=NON)
SS3 DD RC(r=1.00 s=PYS.00.00)
SS3 DD RC(r=1.00 s=NON)
SS3 DD RC(r=0.70 s=NON)
SS3 DD RC(r=0.90 s=PYS.00.00)

Figure 13c

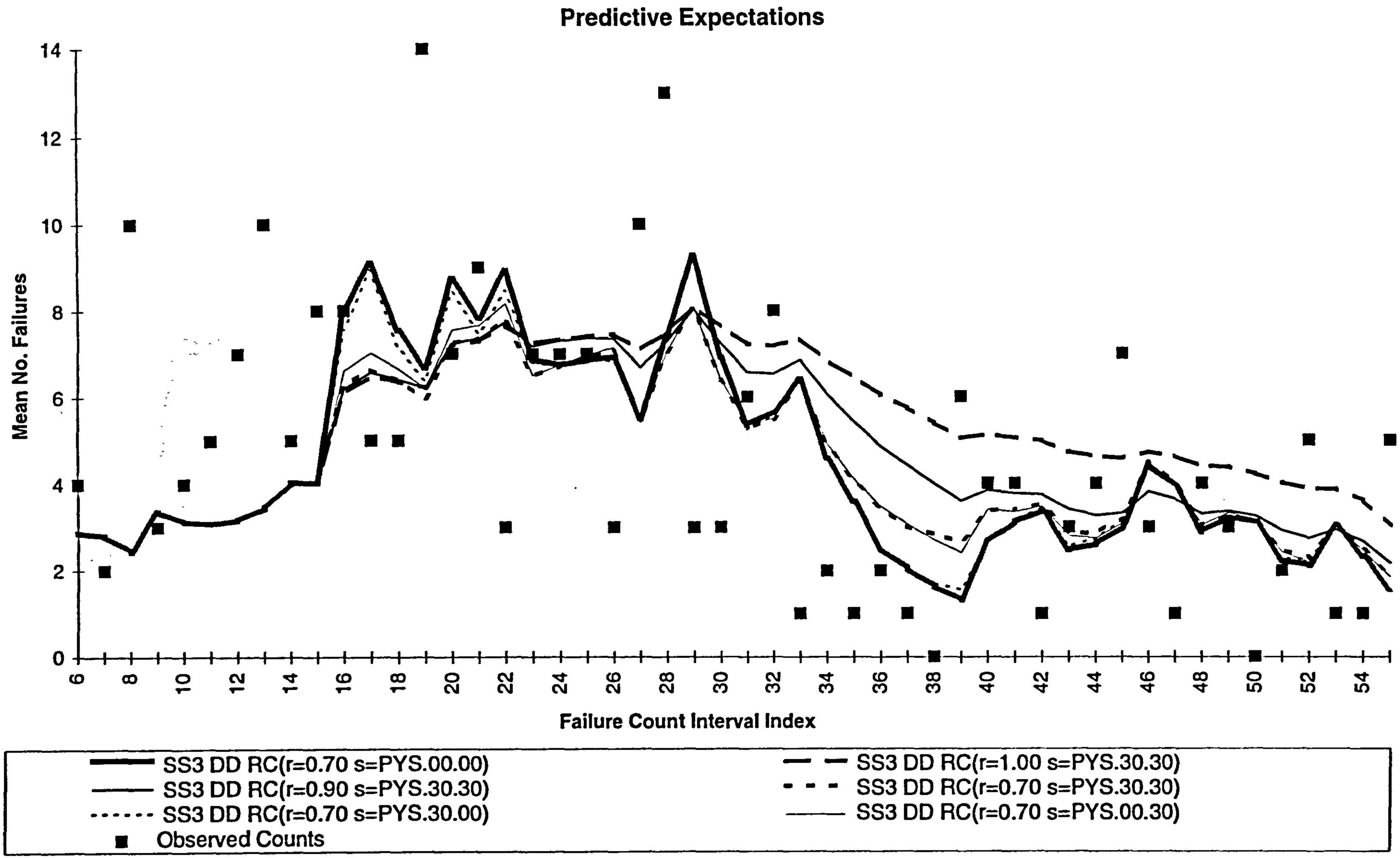

Predictive Expectations
Mean No. Failures
14
12
10
8
6
4
2
0
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
Failure Count Interval Index
SS3 DD RC(r=0.70 s=PYS.00.00)
SS3 DD RC(r=0.90 s=PYS.30.30)
SS3 DD RC(r=0.70 s=PYS.30.00)
Observed Counts
SS3 DD RC(r=1.00 s=PYS.30.30)
SS3 DD RC(r=0.70 s=PYS.30.30)
SS3 DD RC(r=0.70 s=PYS.00.30)

Figure 14a

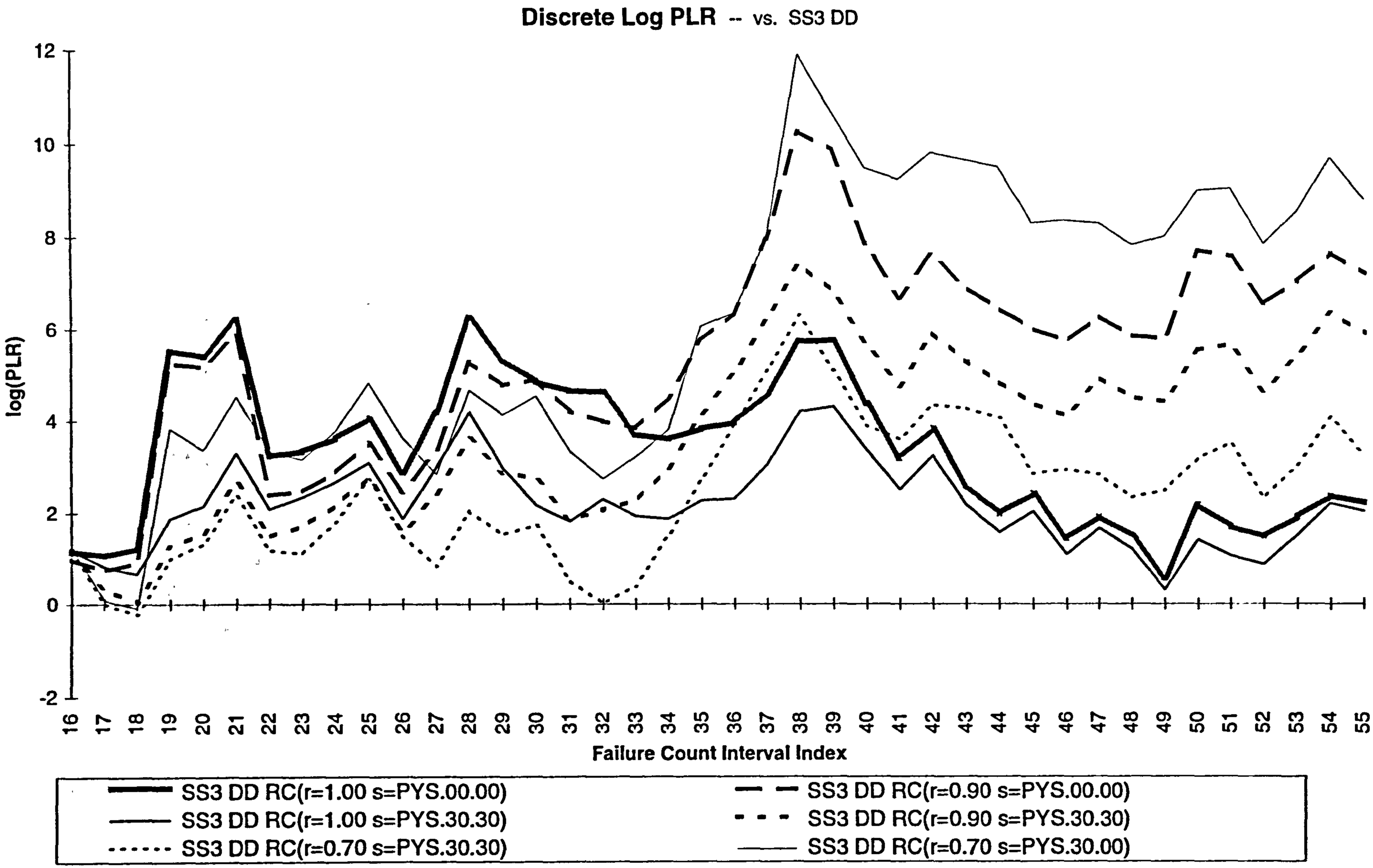

Discrete Log PLR -- vs. SS3 DD
log(PLR)
12
10
8
6
4
2
0
-2
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
Failure Count Interval Index
SS3 DD RC(r=1.00 s=PYS.00.00)
SS3 DD RC(r=1.00 s=PYS.30.30)
SS3 DD RC(r=0.70 s=PYS.30.30)
SS3 DD RC(r=0.90 s=PYS.00.00)
SS3 DD RC(r=0.90 s=PYS.30.30)
SS3 DD RC(r=0.70 s=PYS.30.00)

Figure 14b

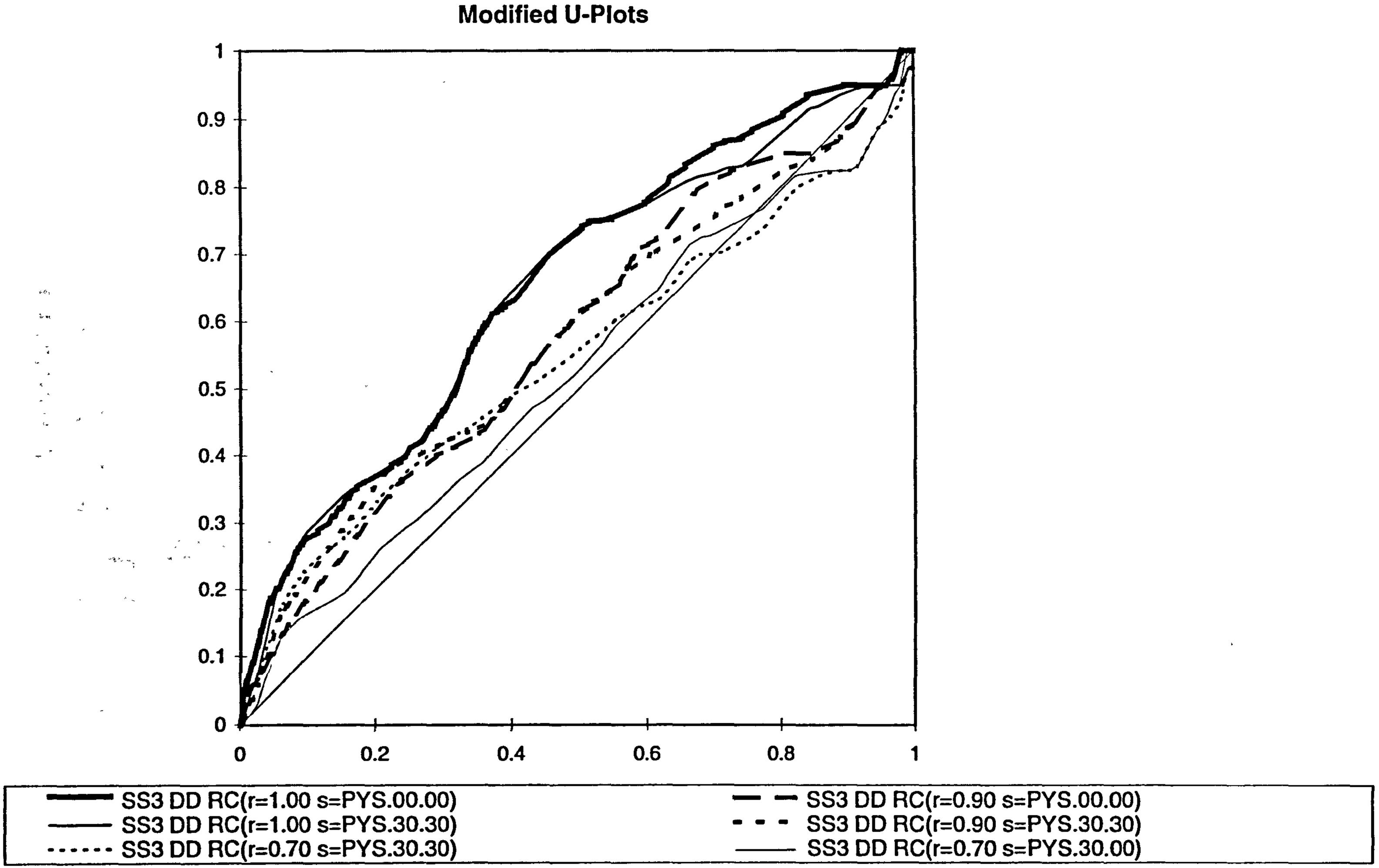

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
SS3 DD RC(r=1.00 s=PYS.00.00)
SS3 DD RC(r=0.90 s=PYS.00.00)
SS3 DD RC(r=1.00 s=PYS.30.30)
SS3 DD RC(r=0.90 s=PYS.30.30)
SS3 DD RC(r=0.70 s=PYS.30.30)
SS3 DD RC(r=0.70 s=PYS.30.00)

Figure 14c

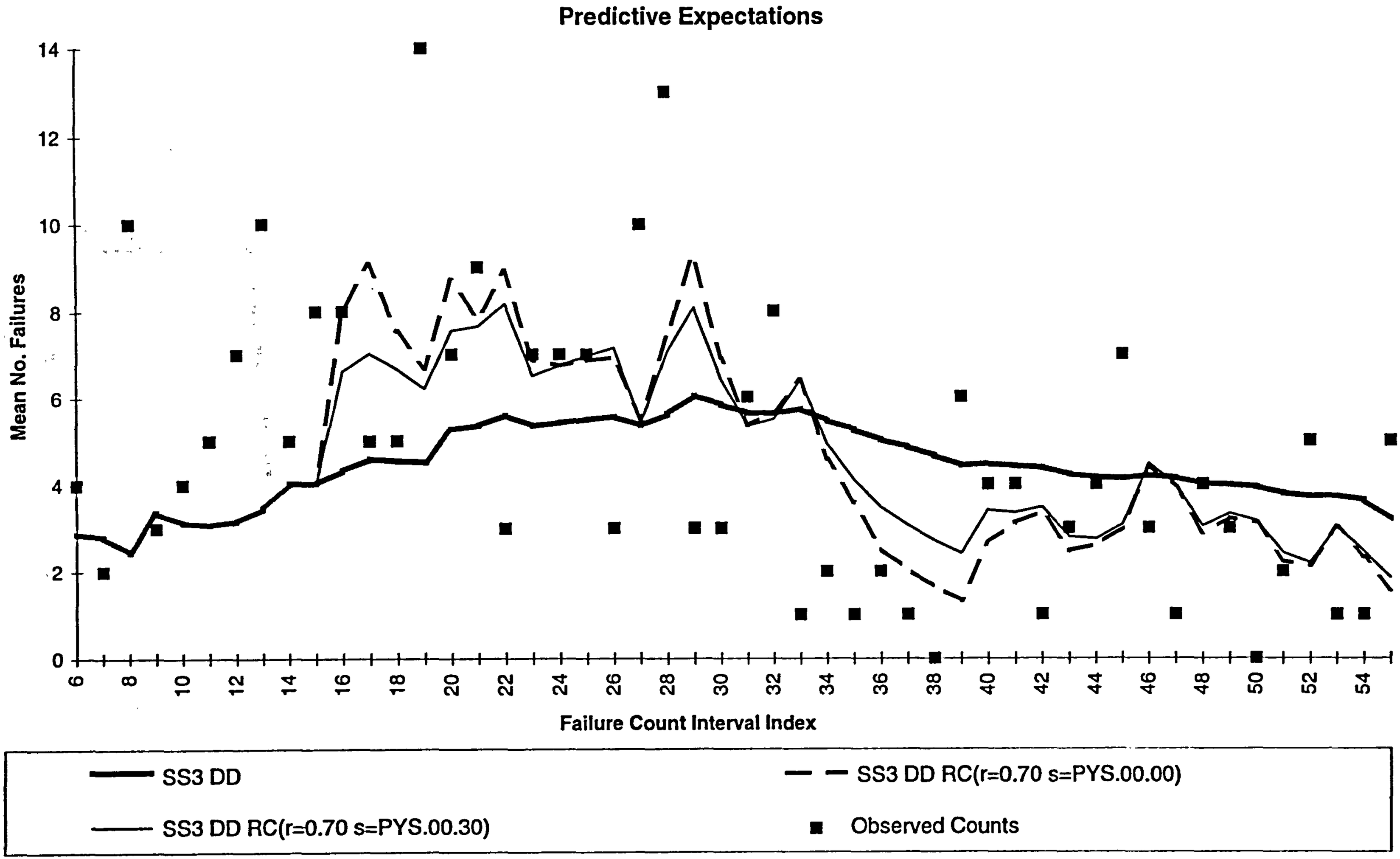

Predictive Expectations
Mean No. Failures
14
12
10
8
6
4
2
0
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
Failure Count Interval Index
SS3 DD
SS3 DD RC(r=0.70 s=PYS.00.00)
SS3 DD RC(r=0.70 s=PYS.00.30)
Observed Counts

Figure 15a

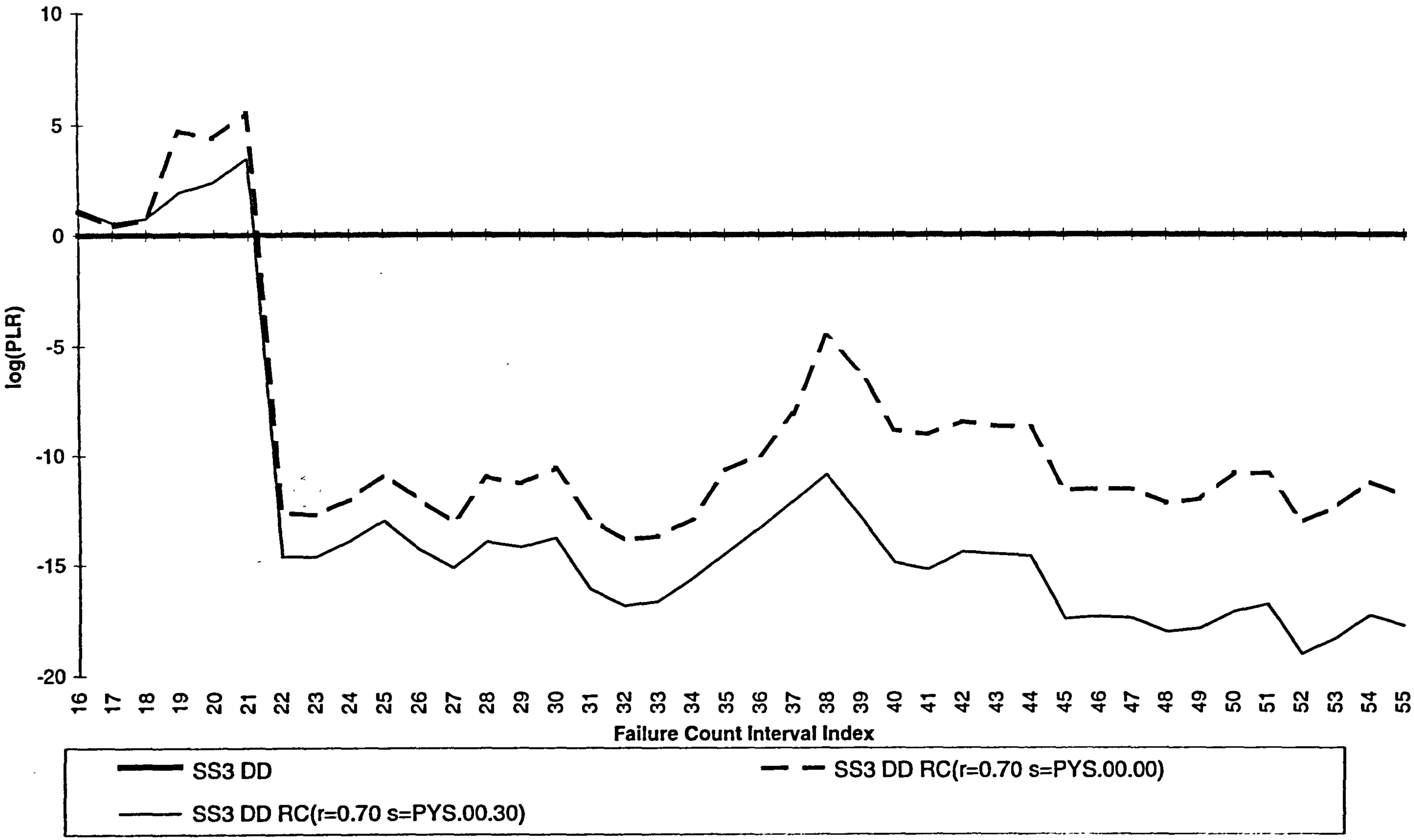

Discrete Log PLR -- vs. SS3 DD
log(PLR)
10
5
0
-5
-10
-15
-20
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
Failure Count Interval Index
SS3 DD
SS3 DD RC(r=0.70 s=PYS.00.00)
SS3 DD RC(r=0.70 s=PYS.00.30)

Figure 15b

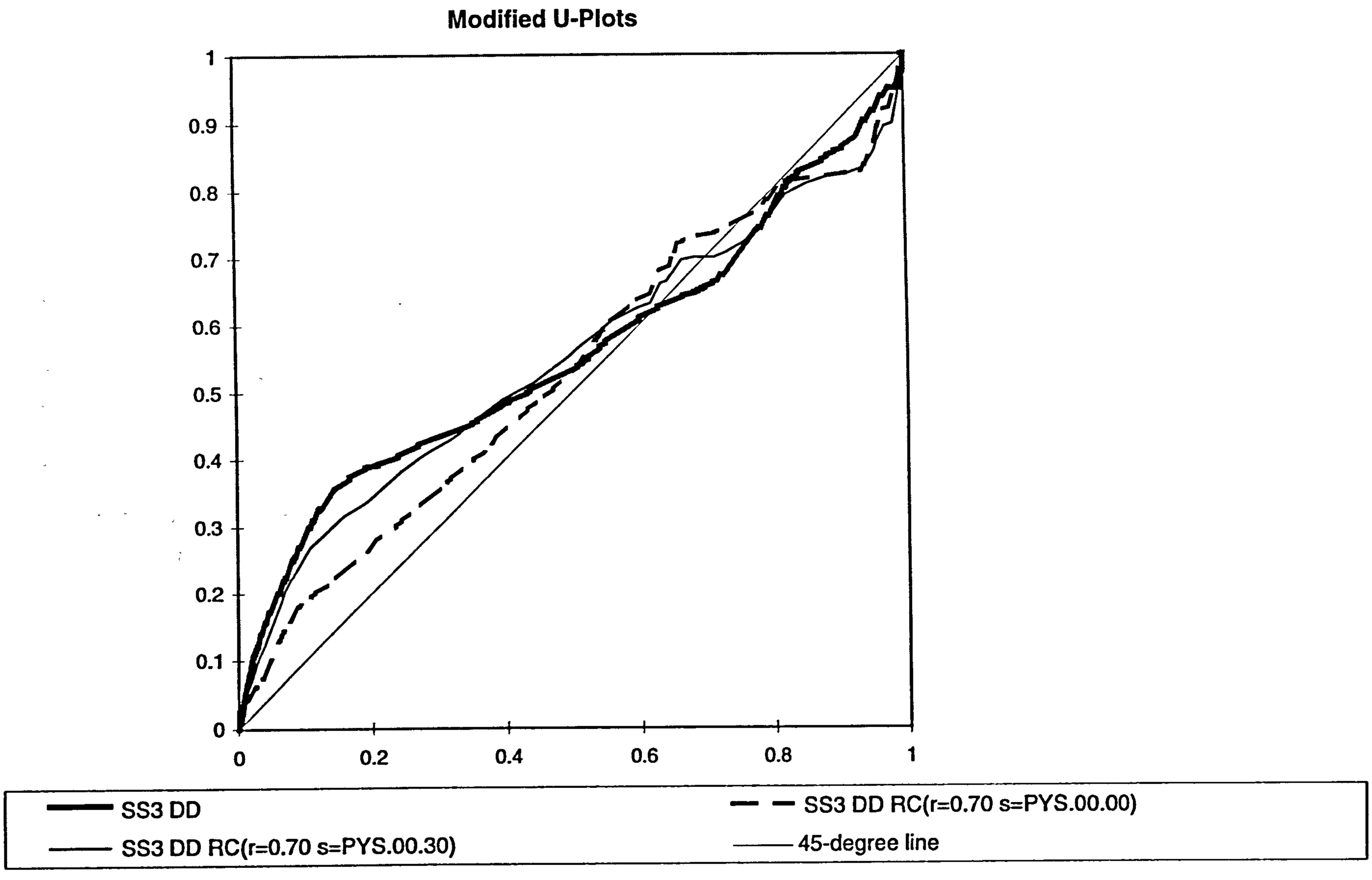

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
SS3 DD
SS3 DD RC(r=0.70 s=PYS.00.00)
SS3 DD RC(r=0.70 s=PYS.00.30)
45-degree line

Figure 15c

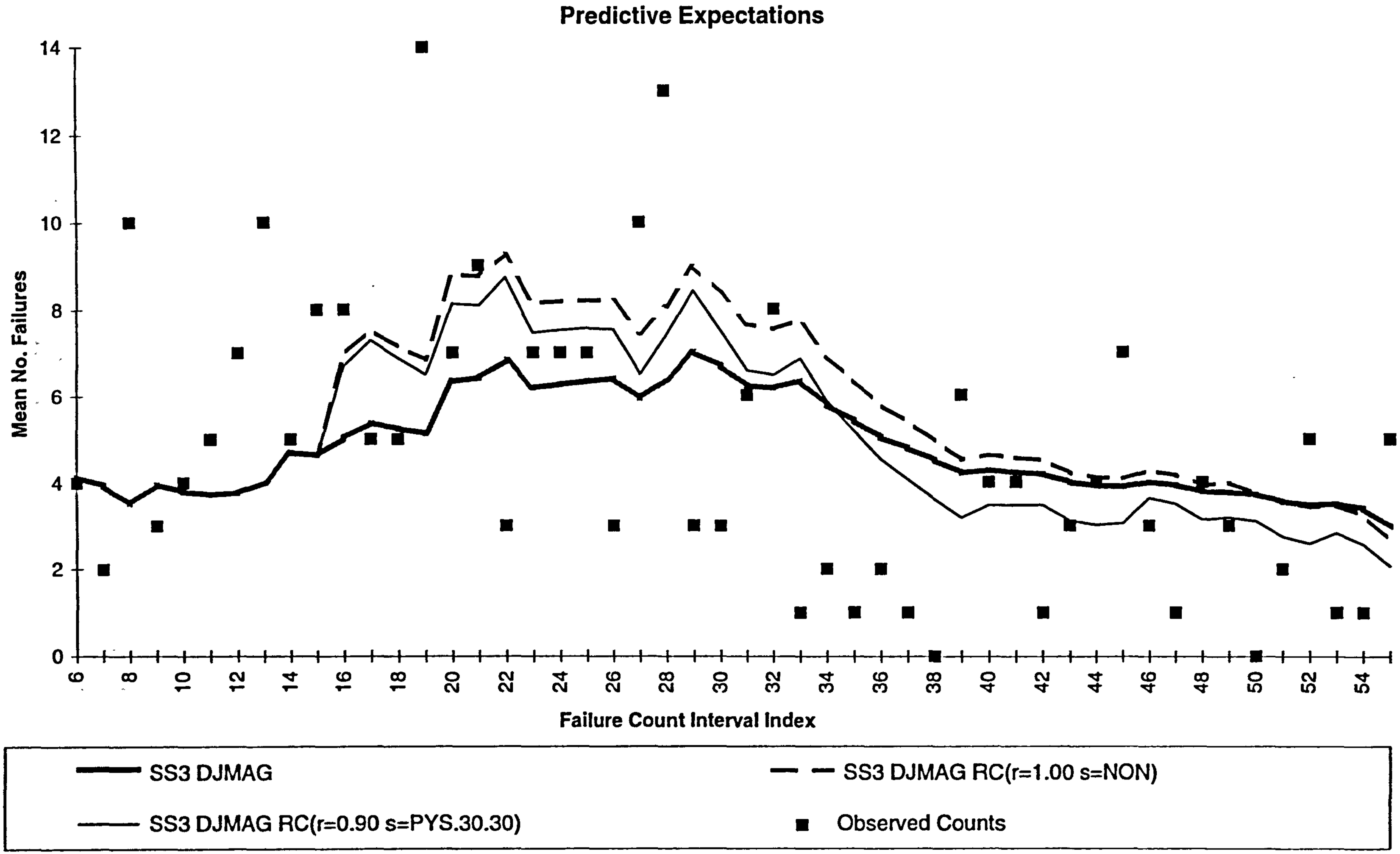

Predictive Expectations
Mean No. Failures
0
2
4
6
8
10
12
14
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
Failure Count Interval Index
SS3 DJMAG
SS3 DJMAG RC(r=1.00 s=NON)
SS3 DJMAG RC(r=0.90 s=PYS.30.30)
Observed Counts

Figure 16a

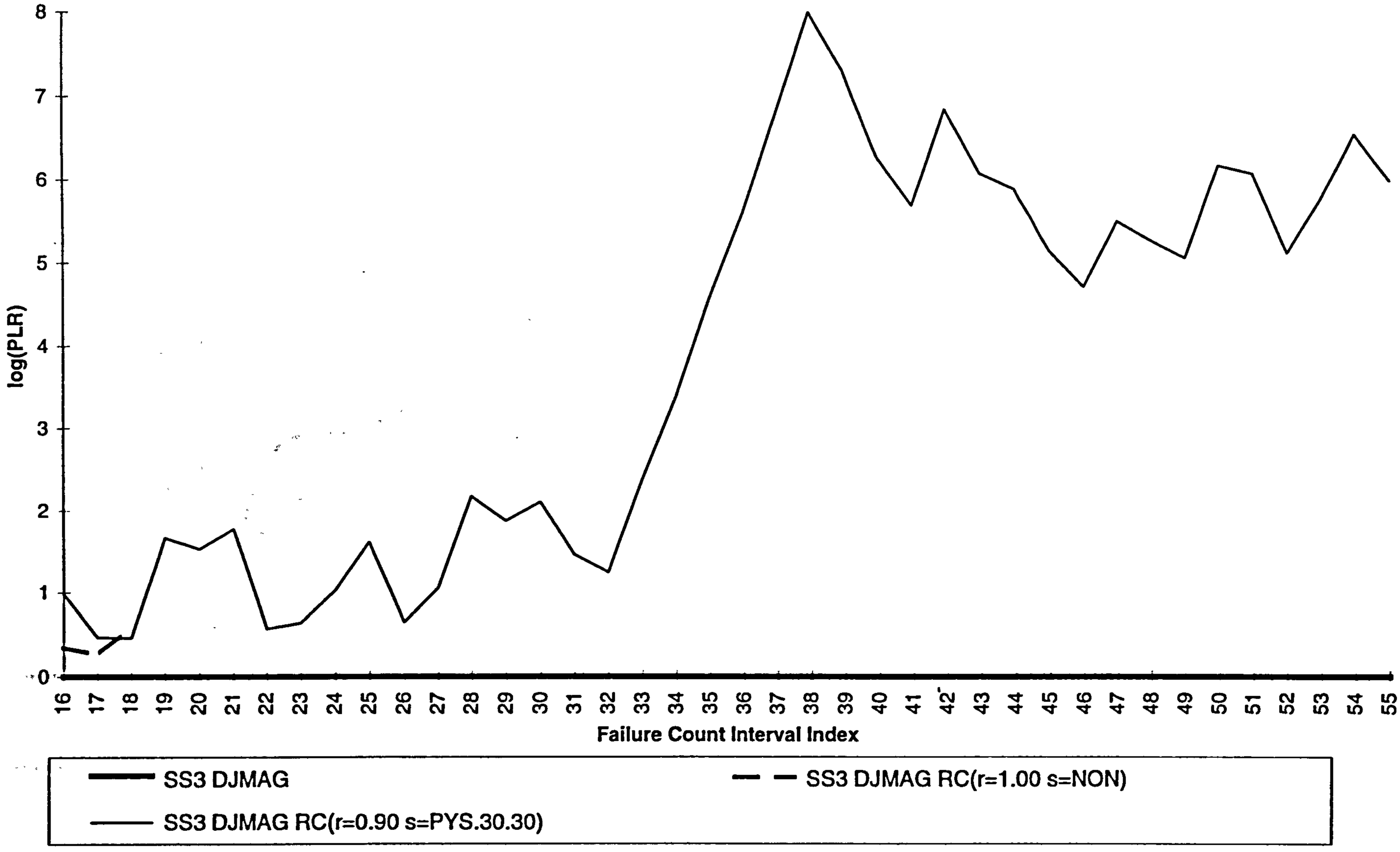

Discrete Log PLR -- vs. SS3 DJMAG
log(PLR)
8
7
6
5
4
3
2
1
0
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
Failure Count Interval Index
SS3 DJMAG
SS3 DJMAG RC(r=1.00 s=NON)
SS3 DJMAG RC(r=0.90 s=PYS.30.30)

Figure 16b

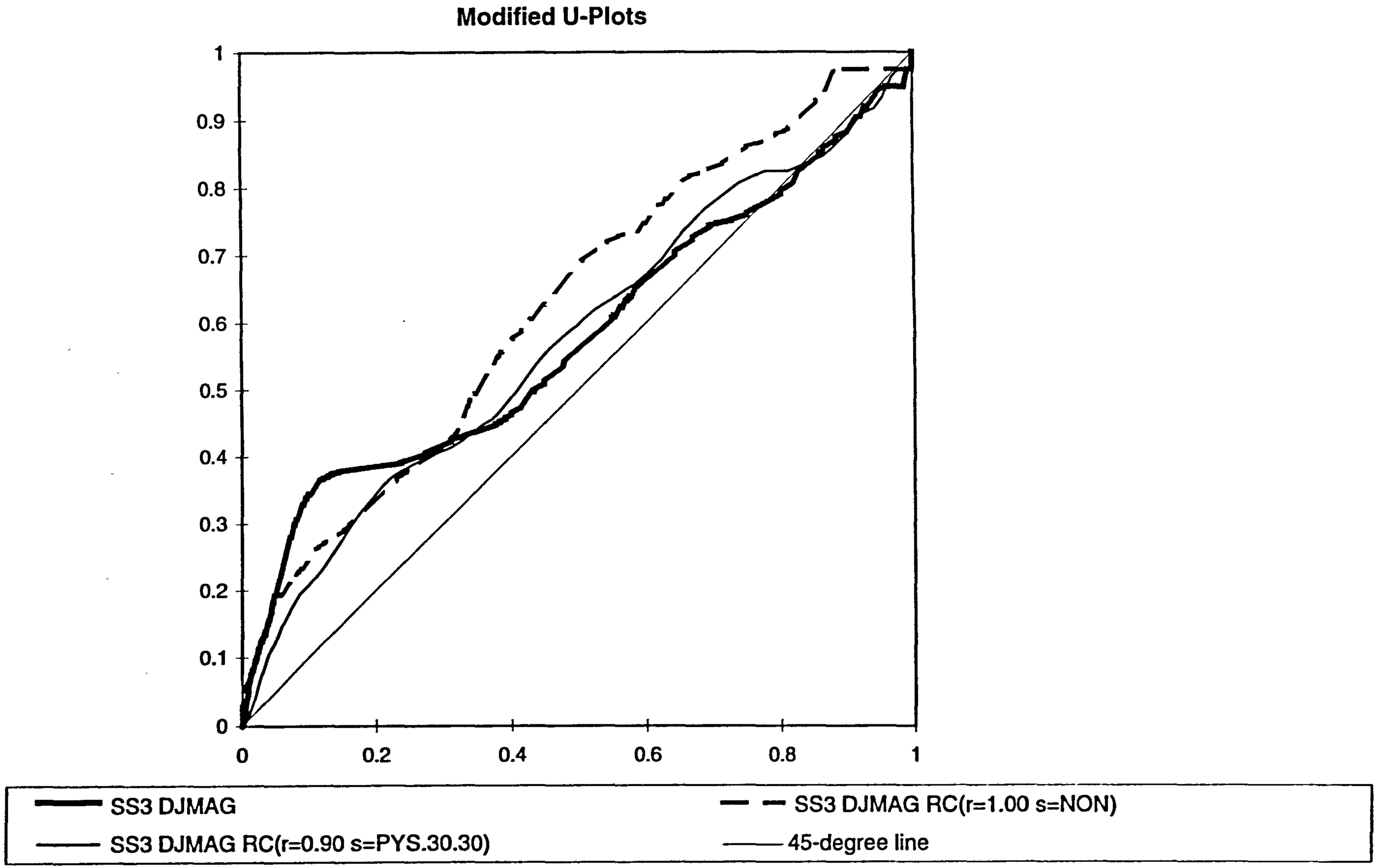

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
SS3 DJMAG
SS3 DJMAG RC(r=1.00 s=NON)
SS3 DJMAG RC(r=0.90 s=PYS.30.30)
45-degree line

Figure 16c

Failure Count Data: SYS1 - Interval Length 1,000 seconds

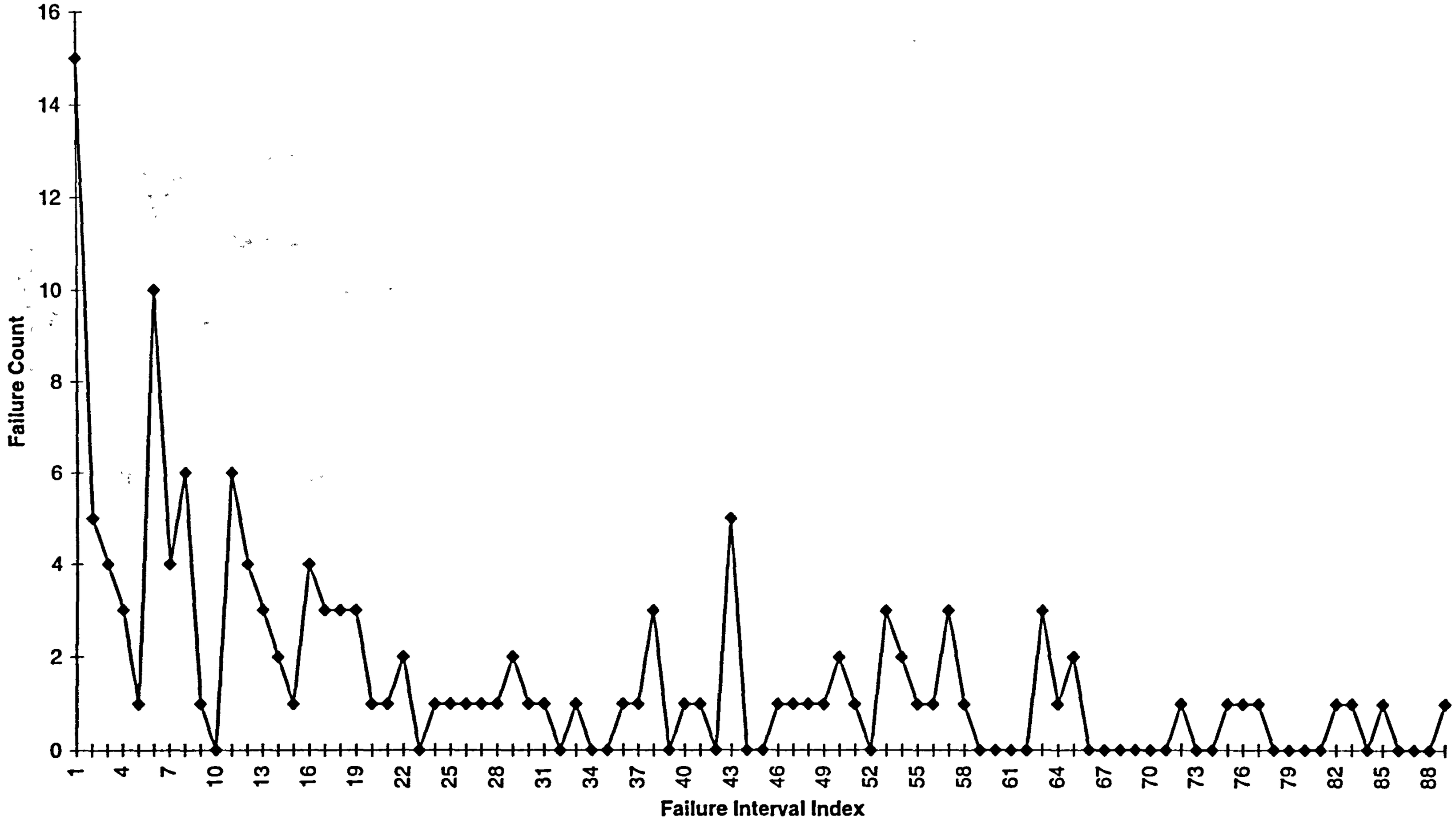


Figure 17

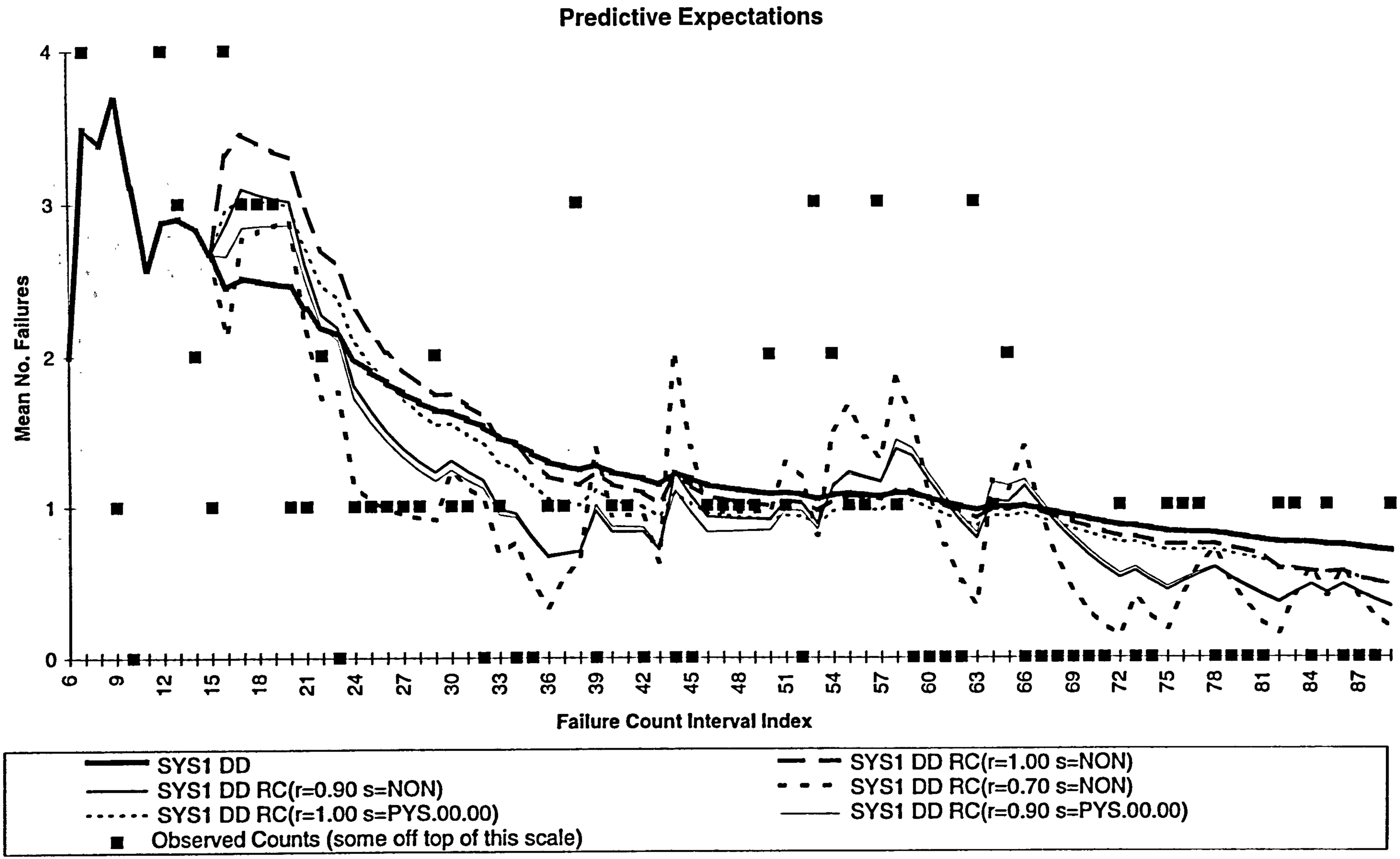

Predictive Expectations
Mean No. Failures
4
3
2
1
0
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
51
54
57
60
63
66
69
72
75
78
81
84
87
Failure Count Interval Index
SYS1 DD
SYS1 DD RC(r=0.90 s=NON)
SYS1 DD RC(r=1.00 s=PYS.00.00)
Observed Counts (some off top of this scale)
SYS1 DD RC(r=1.00 s=NON)
SYS1 DD RC(r=0.70 s=NON)
SYS1 DD RC(r=0.90 s=PYS.00.00)

Figure 18a

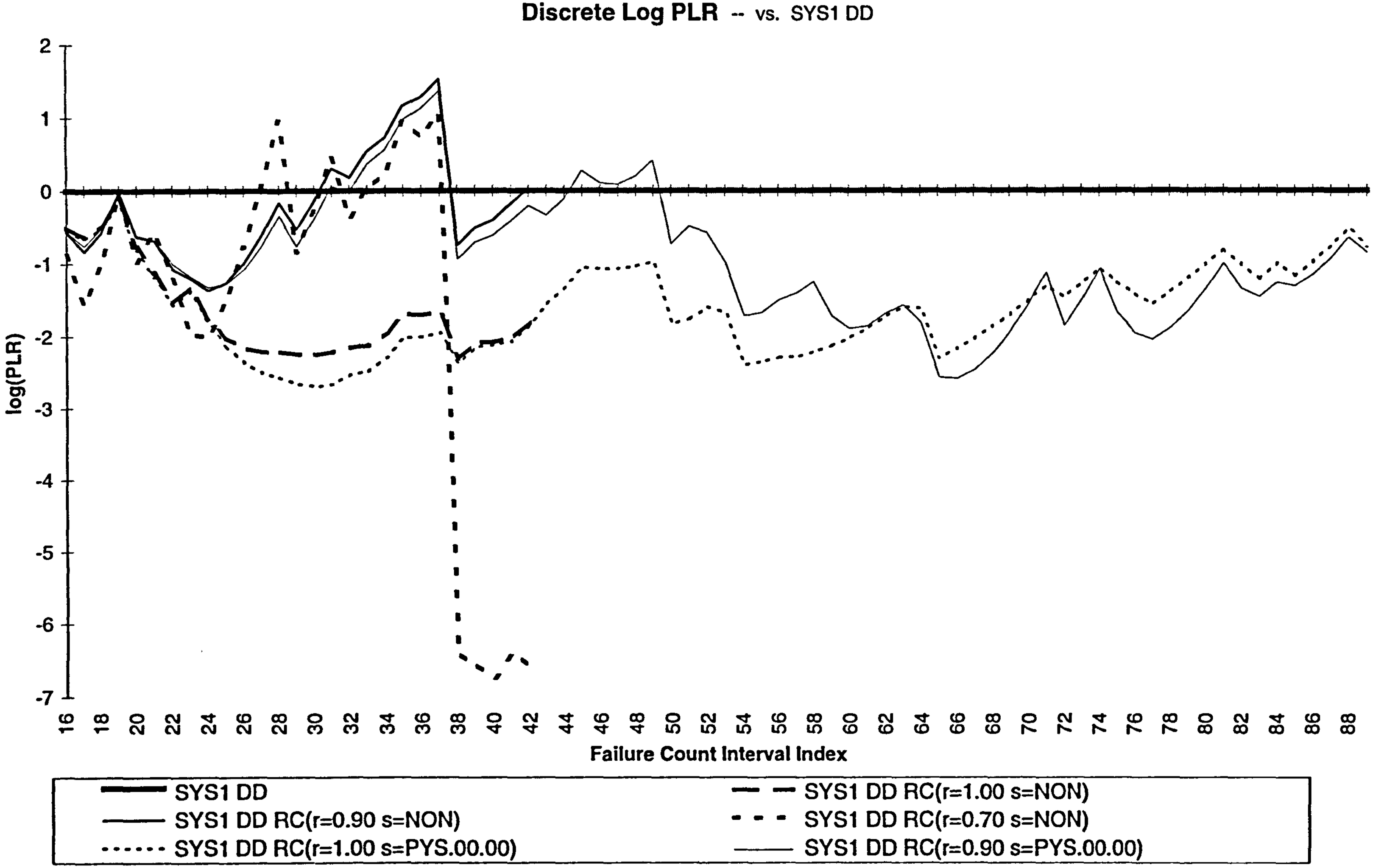

Discrete Log PLR -- vs. SYS1 DD
log(PLR)
2
1
0
-1
-2
-3
-4
-5
-6
-7
16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88
Failure Count Interval Index
SYS1 DD
SYS1 DD RC(r=0.90 s=NON)
SYS1 DD RC(r=1.00 s=PYS.00.00)
SYS1 DD RC(r=1.00 s=NON)
SYS1 DD RC(r=0.70 s=NON)
SYS1 DD RC(r=0.90 s=PYS.00.00)

Figure 18b

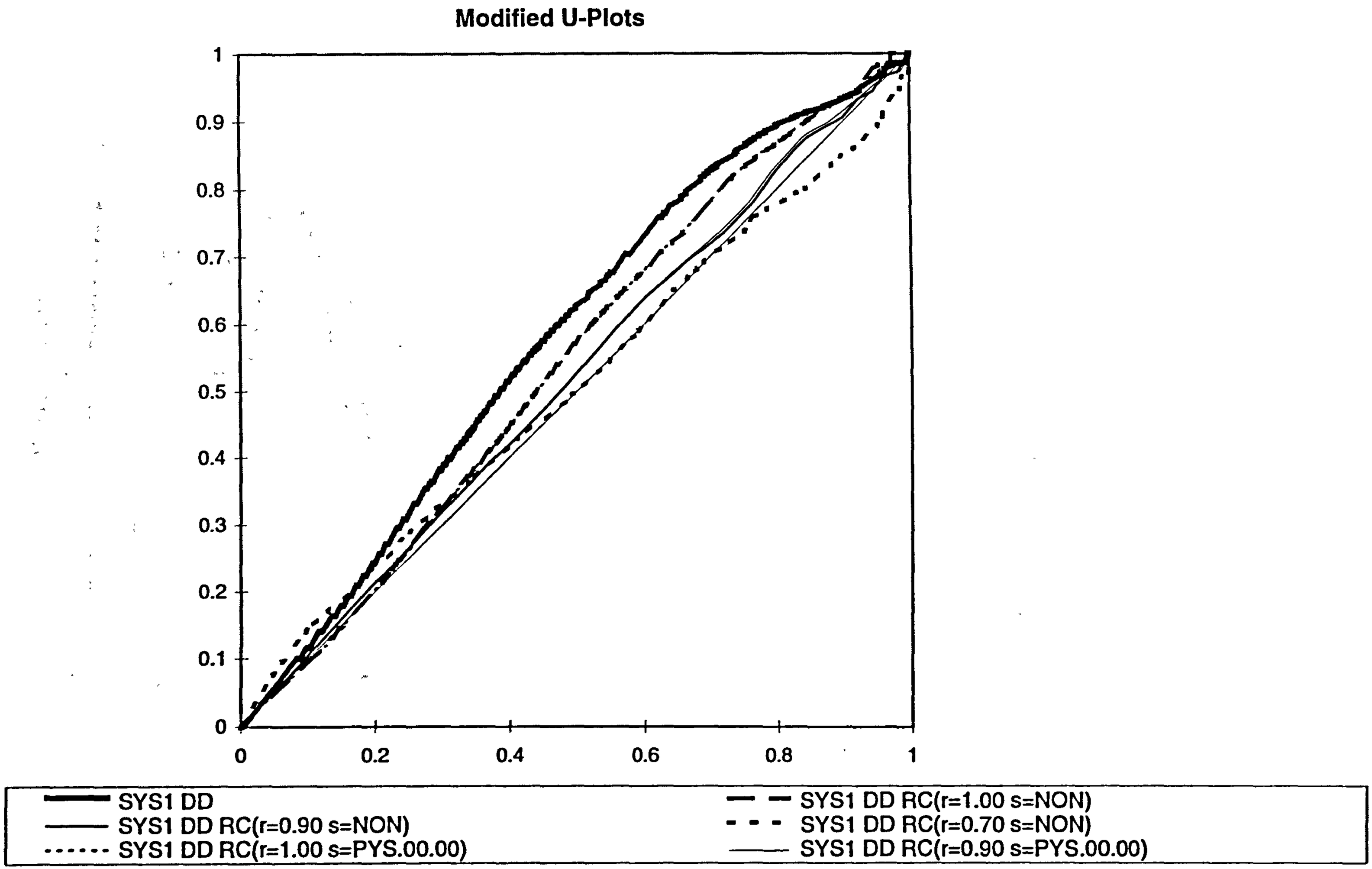

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
SYS1 DD
SYS1 DD RC(r=0.90 s=NON)
SYS1 DD RC(r=1.00 s=PYS.00.00)
SYS1 DD RC(r=1.00 s=NON)
SYS1 DD RC(r=0.70 s=NON)
SYS1 DD RC(r=0.90 s=PYS.00.00)

Figure 18c

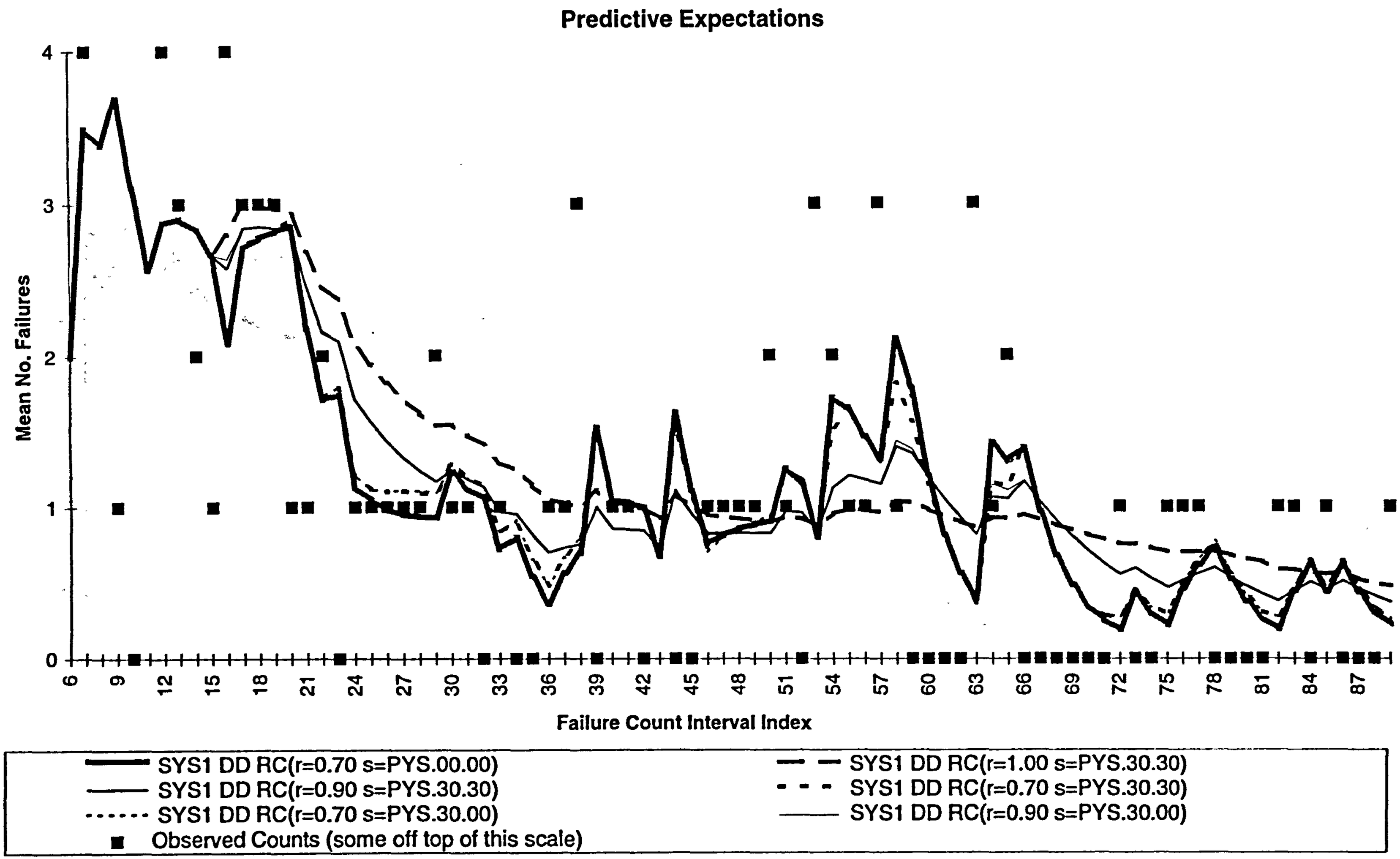

Predictive Expectations
Mean No. Failures
4
3
2
1
0
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
51
54
57
60
63
66
69
72
75
78
81
84
87
Failure Count Interval Index
SYS1 DD RC(r=0.70 s=PYS.00.00)
SYS1 DD RC(r=0.90 s=PYS.30.30)
SYS1 DD RC(r=0.70 s=PYS.30.00)
Observed Counts (some off top of this scale)
SYS1 DD RC(r=1.00 s=PYS.30.30)
SYS1 DD RC(r=0.70 s=PYS.30.30)
SYS1 DD RC(r=0.90 s=PYS.30.00)

Figure 19a

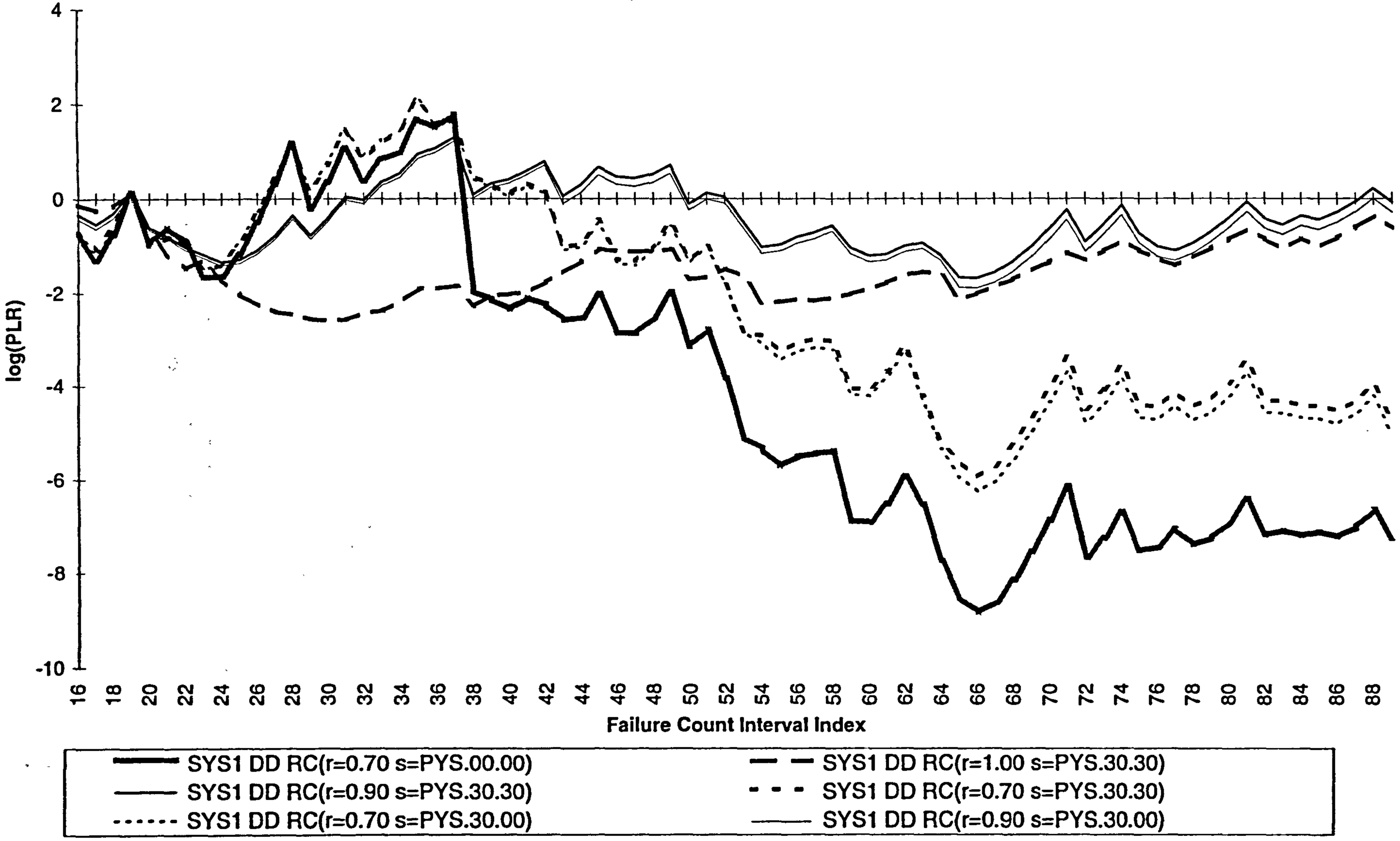

Discrete Log PLR -- vs. SYS1 DD
log(PLR)
4
2
0
-2
-4
-6
-8
-10
16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88
Failure Count Interval Index
SYS1 DD RC(r=0.70 s=PYS.00.00)
SYS1 DD RC(r=0.90 s=PYS.30.30)
SYS1 DD RC(r=0.70 s=PYS.30.00)
SYS1 DD RC(r=1.00 s=PYS.30.30)
SYS1 DD RC(r=0.70 s=PYS.30.30)
SYS1 DD RC(r=0.90 s=PYS.30.00)

Figure 19b

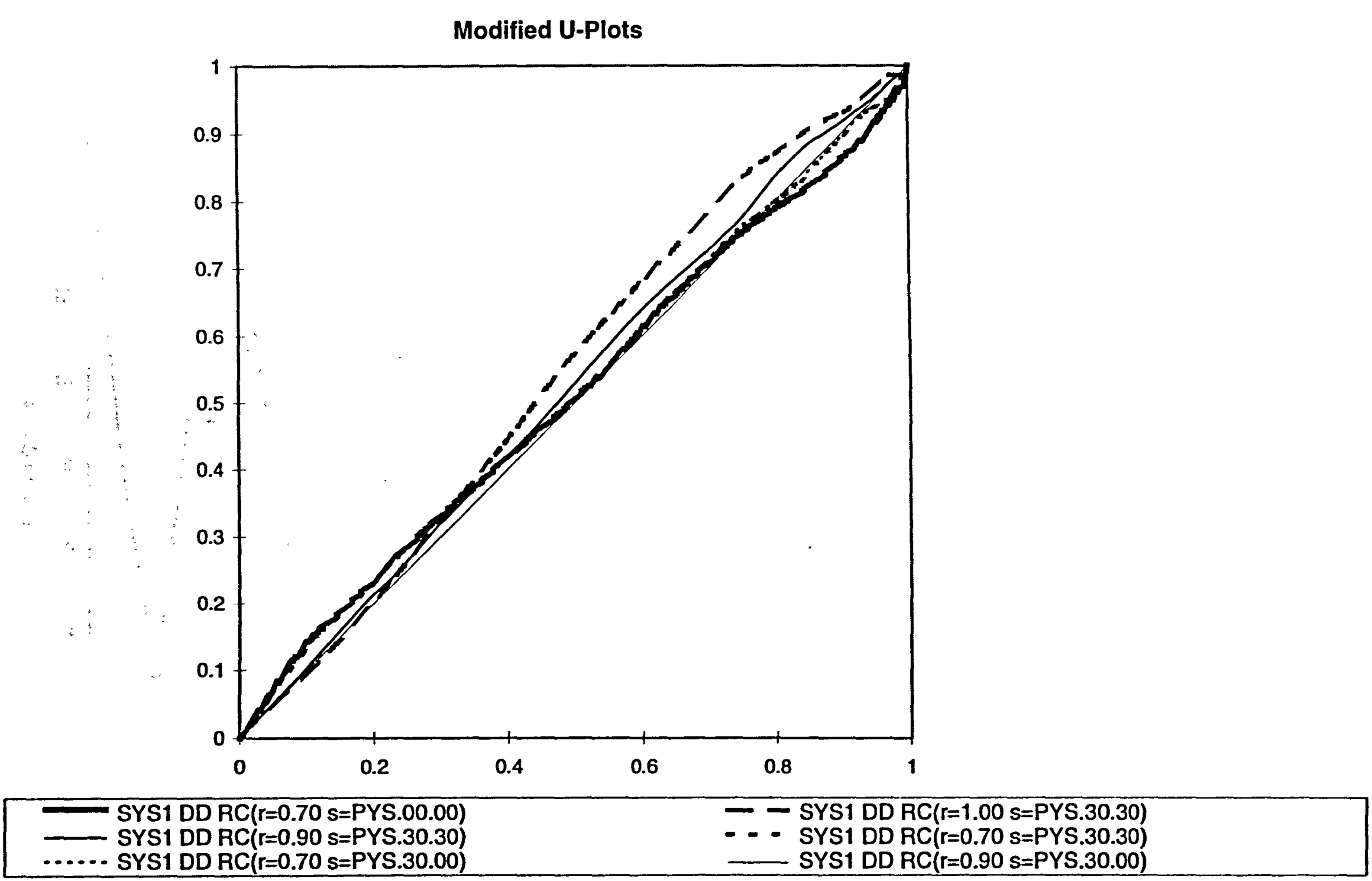

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
SYS1 DD RC(r=0.70 s=PYS.00.00)
SYS1 DD RC(r=0.90 s=PYS.30.30)
SYS1 DD RC(r=0.70 s=PYS.30.00)
SYS1 DD RC(r=1.00 s=PYS.30.30)
SYS1 DD RC(r=0.70 s=PYS.30.30)
SYS1 DD RC(r=0.90 s=PYS.30.00)

Figure 19c

Failure Count Data: AAA - Interval Length 400

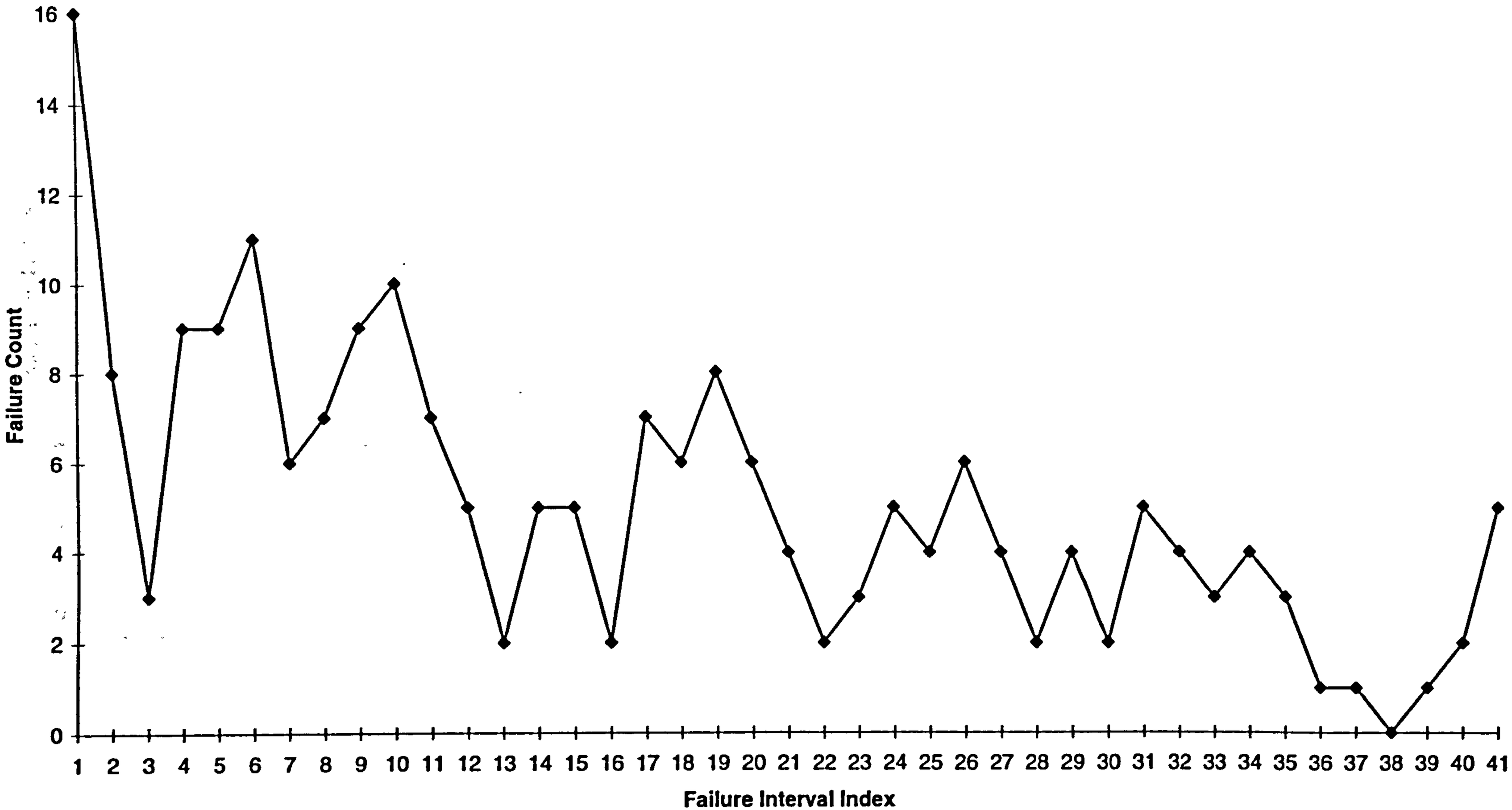


Figure 20

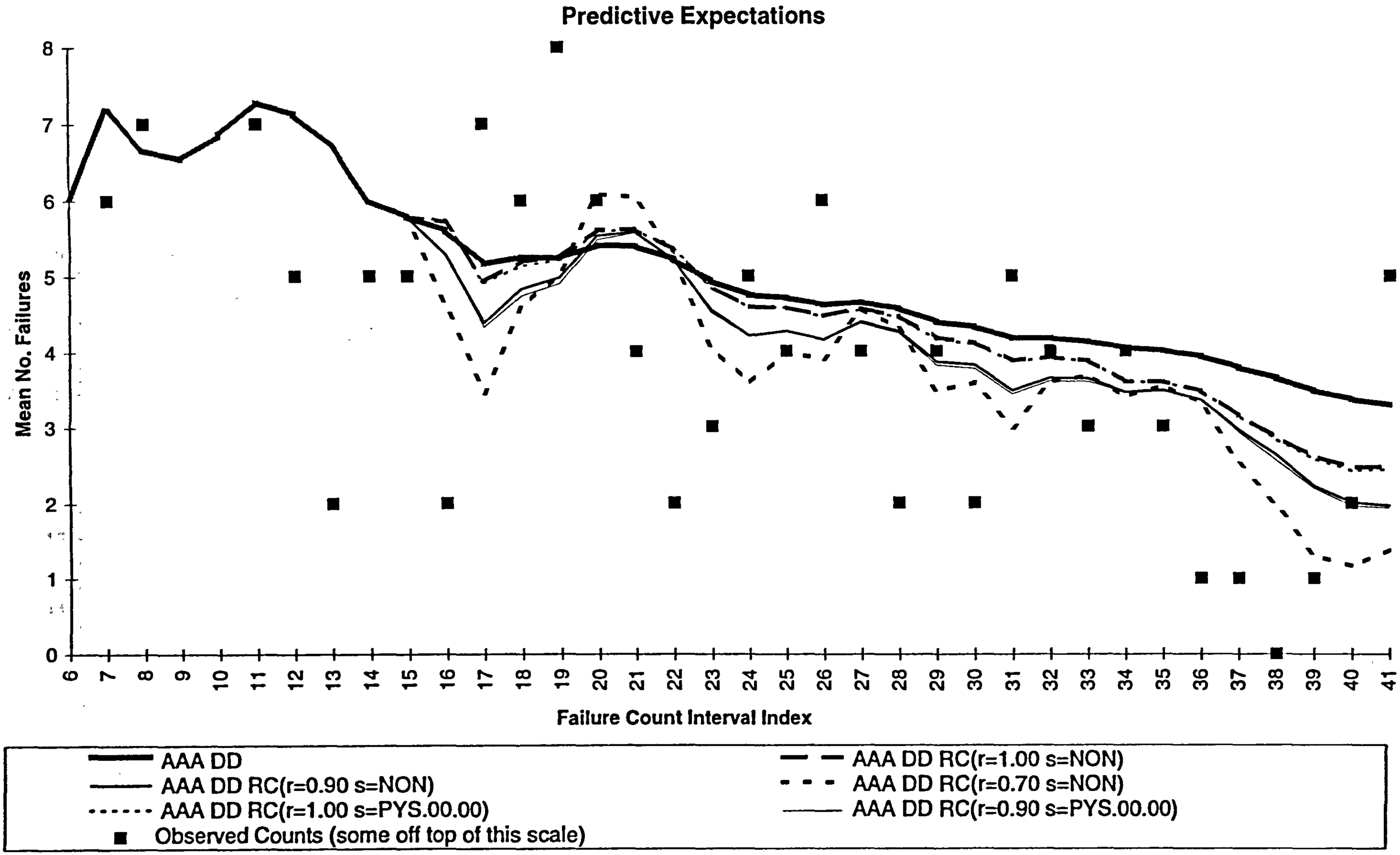

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
7
8
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
Failure Count Interval Index
AAA DD
AAA DD RC(r=0.90 s=NON)
AAA DD RC(r=1.00 s=PYS.00.00)
Observed Counts (some off top of this scale)
AAA DD RC(r=1.00 s=NON)
AAA DD RC(r=0.70 s=NON)
AAA DD RC(r=0.90 s=PYS.00.00)

Figure 21a

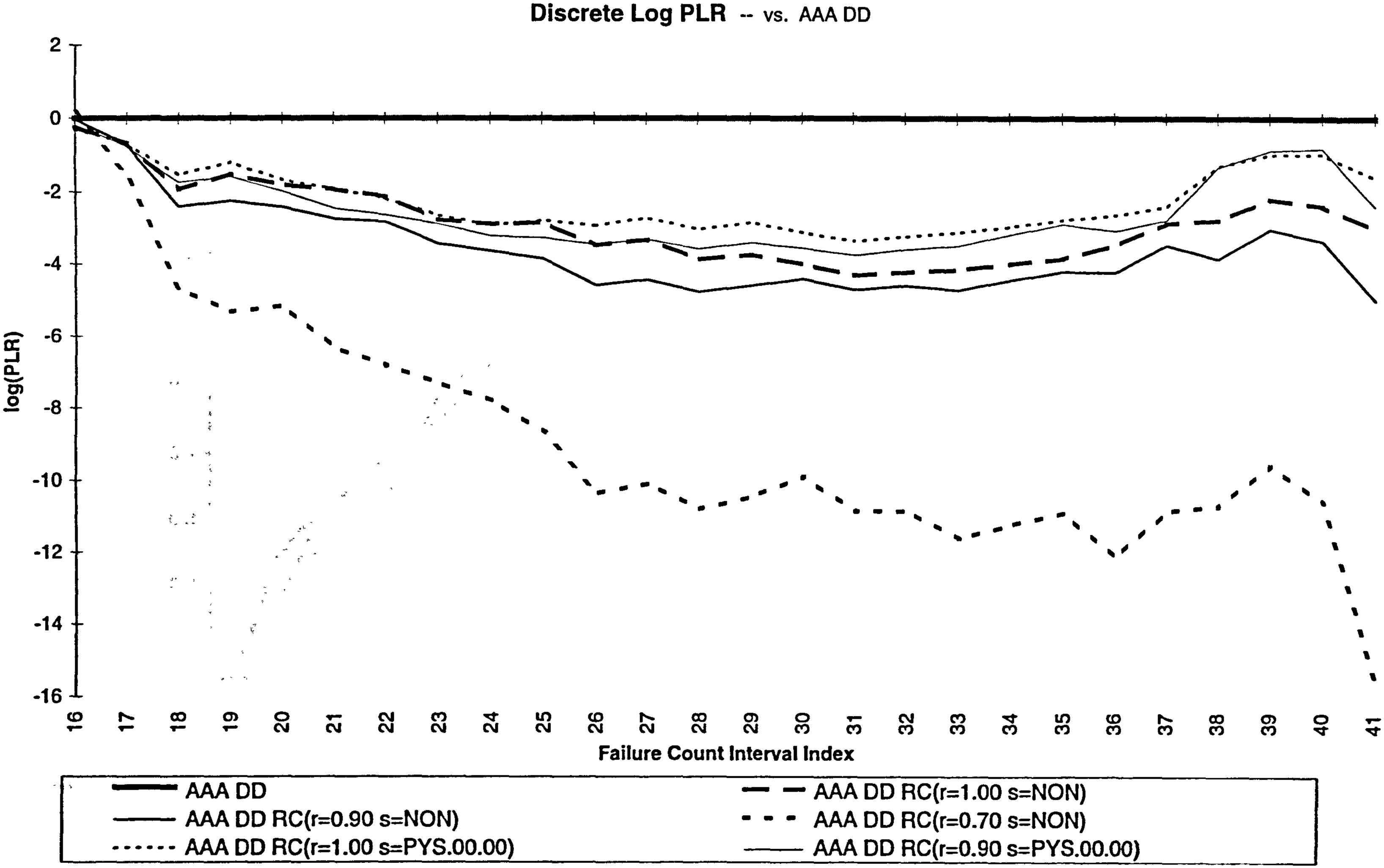

Discrete Log PLR -- vs. AAA DD
log(PLR)
2
0
-2
-4
-6
-8
-10
-12
-14
-16
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
Failure Count Interval Index
AAA DD
AAA DD RC(r=0.90 s=NON)
AAA DD RC(r=1.00 s=PYS.00.00)
AAA DD RC(r=1.00 s=NON)
AAA DD RC(r=0.70 s=NON)
AAA DD RC(r=0.90 s=PYS.00.00)

Figure 21b

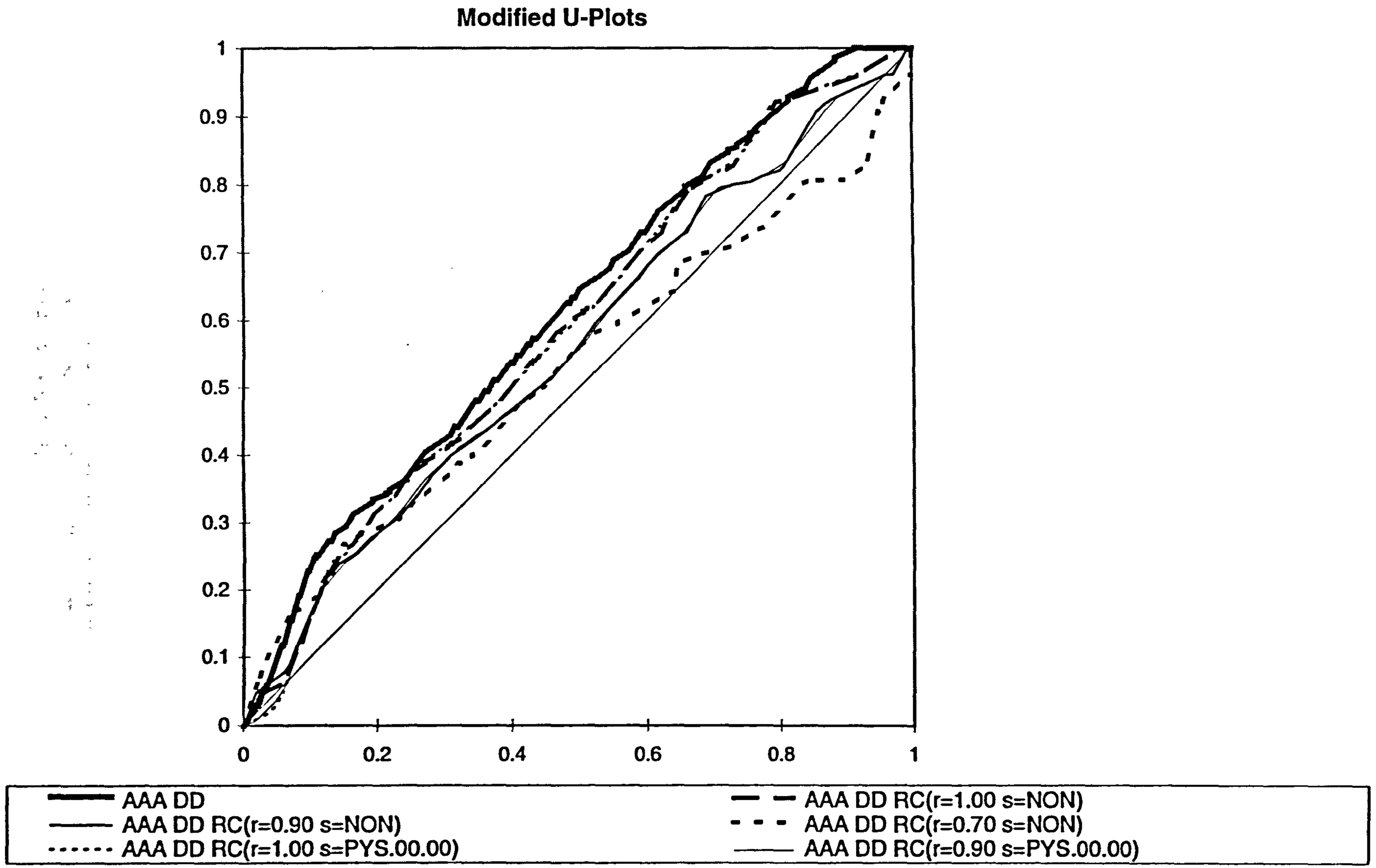

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
AAA DD
AAA DD RC(r=0.90 s=NON)
AAA DD RC(r=1.00 s=PYS.00.00)
AAA DD RC(r=1.00 s=NON)
AAA DD RC(r=0.70 s=NON)
AAA DD RC(r=0.90 s=PYS.00.00)

Figure 21c

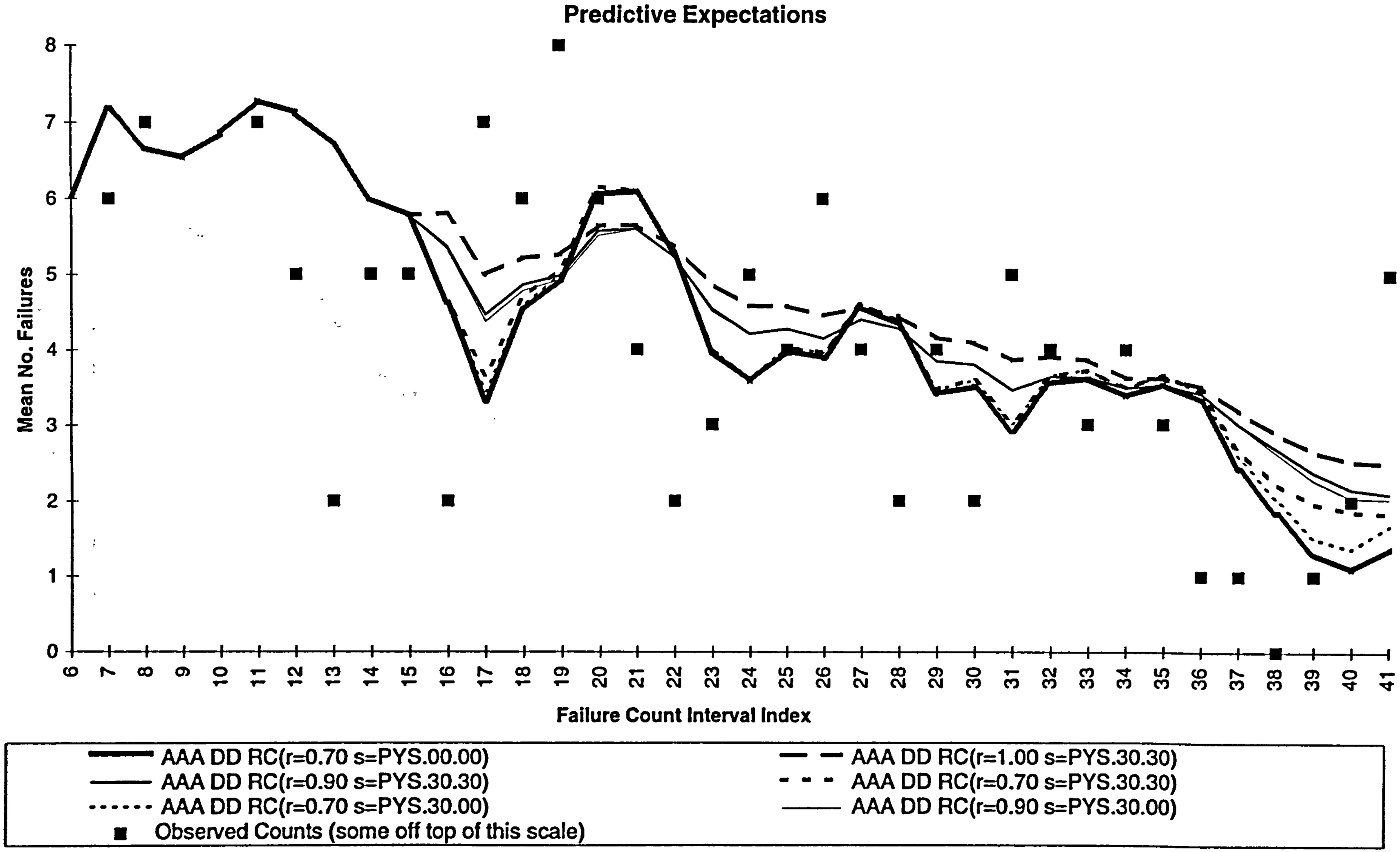

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
7
8
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
Failure Count Interval Index
AAA DD RC(r=0.70 s=PYS.00.00)
AAA DD RC(r=0.90 s=PYS.30.30)
AAA DD RC(r=0.70 s=PYS.30.00)
Observed Counts (some off top of this scale)
AAA DD RC(r=1.00 s=PYS.30.30)
AAA DD RC(r=0.70 s=PYS.30.30)
AAA DD RC(r=0.90 s=PYS.30.00)

Figure 22a

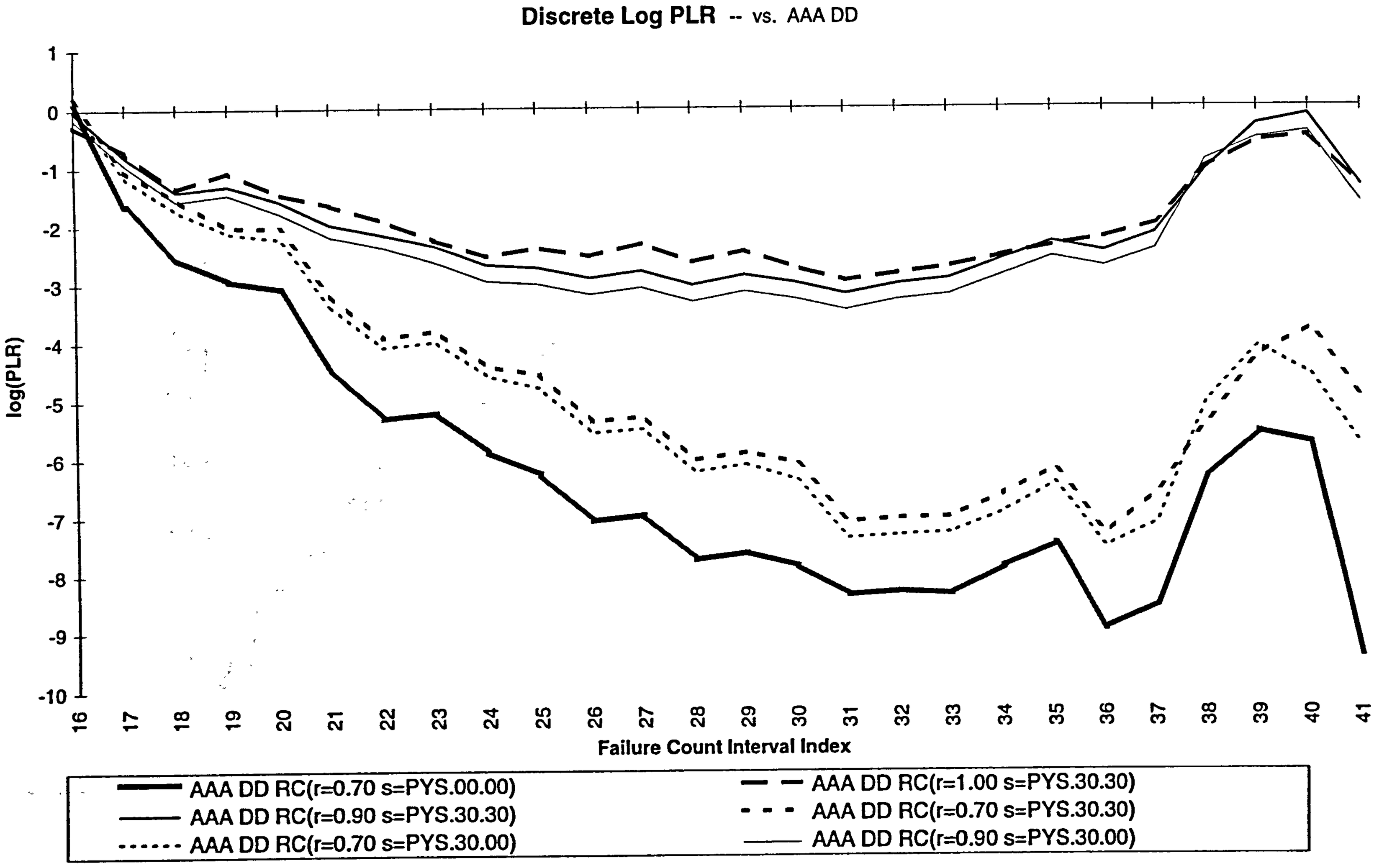

Discrete Log PLR -- vs. AAA DD
log(PLR)
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
Failure Count Interval Index
AAA DD RC(r=0.70 s=PYS.00.00)
AAA DD RC(r=0.90 s=PYS.30.30)
AAA DD RC(r=0.70 s=PYS.30.00)
AAA DD RC(r=1.00 s=PYS.30.30)
AAA DD RC(r=0.70 s=PYS.30.30)
AAA DD RC(r=0.90 s=PYS.30.00)

Figure 22b

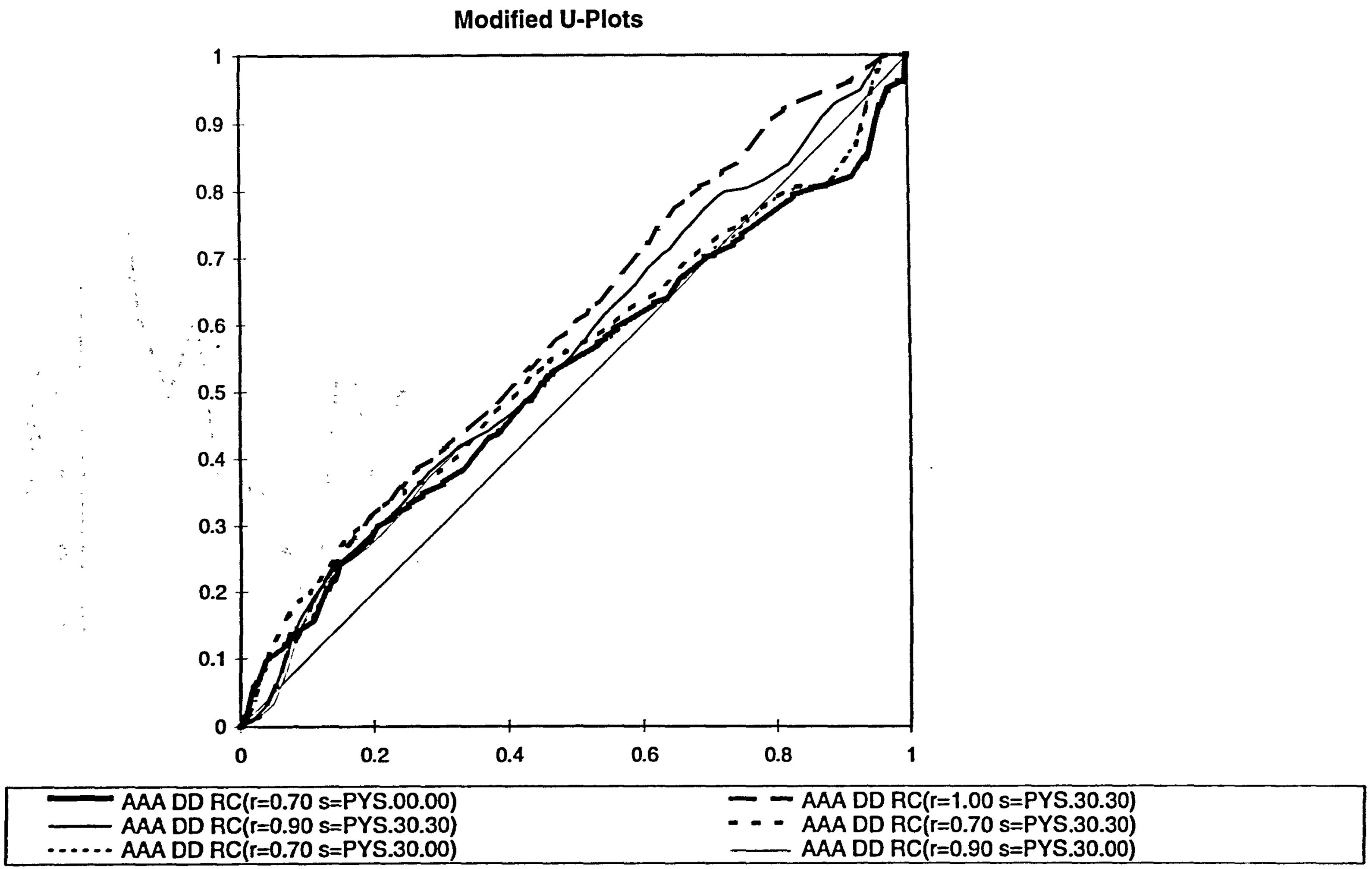

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
AAA DD RC(r=0.70 s=PYS.00.00)
AAA DD RC(r=0.90 s=PYS.30.30)
AAA DD RC(r=0.70 s=PYS.30.00)
AAA DD RC(r=1.00 s=PYS.30.30)
AAA DD RC(r=0.70 s=PYS.30.30)
AAA DD RC(r=0.90 s=PYS.30.00)

Figure 22c

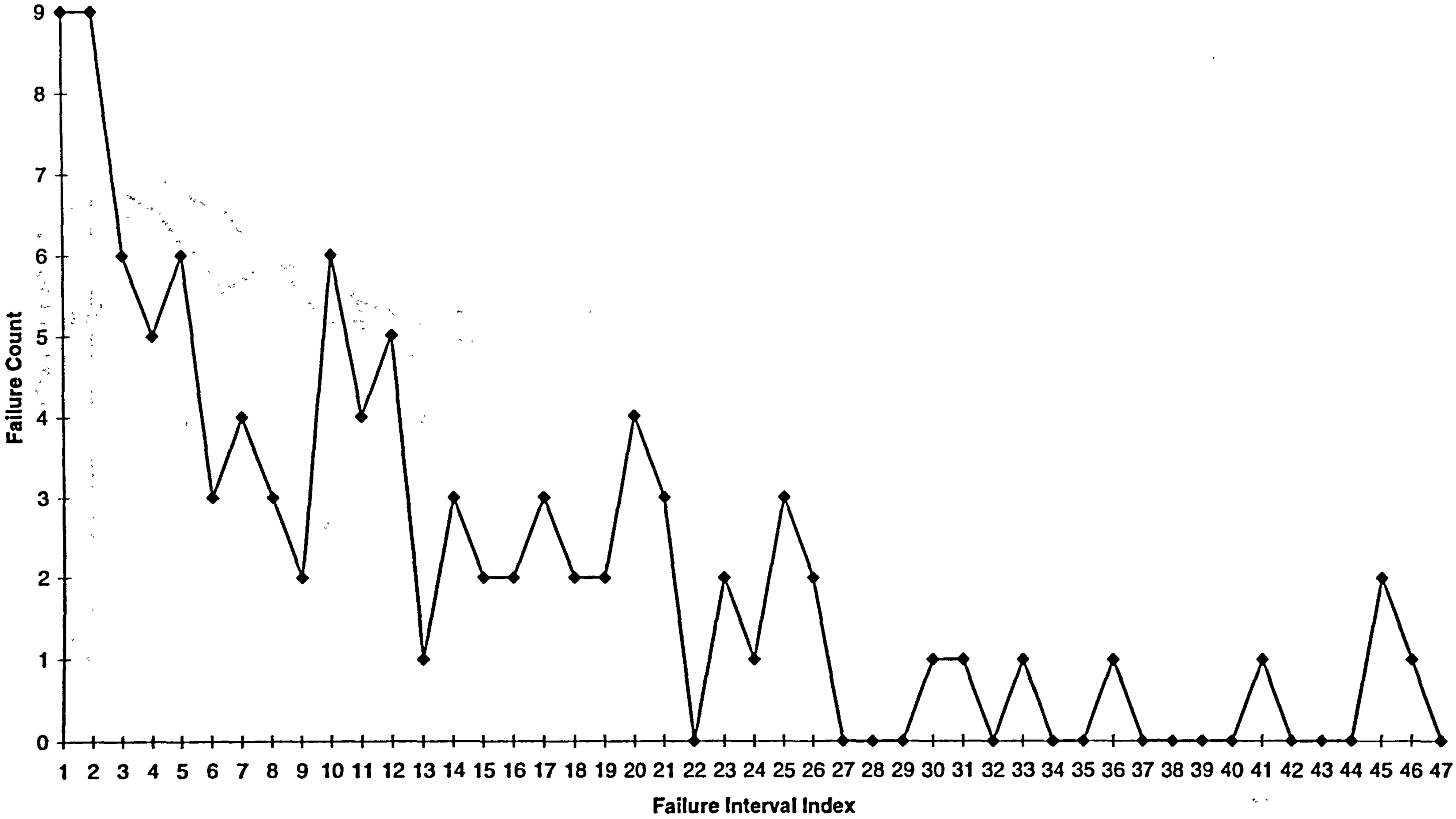

Failure Count Data: JM1 - Simulated. N=106, phi=0.00007, Interval Length 1,000
Failure Count
0
1
2
3
4
5
6
7
8
9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
Failure Interval Index

Figure 23

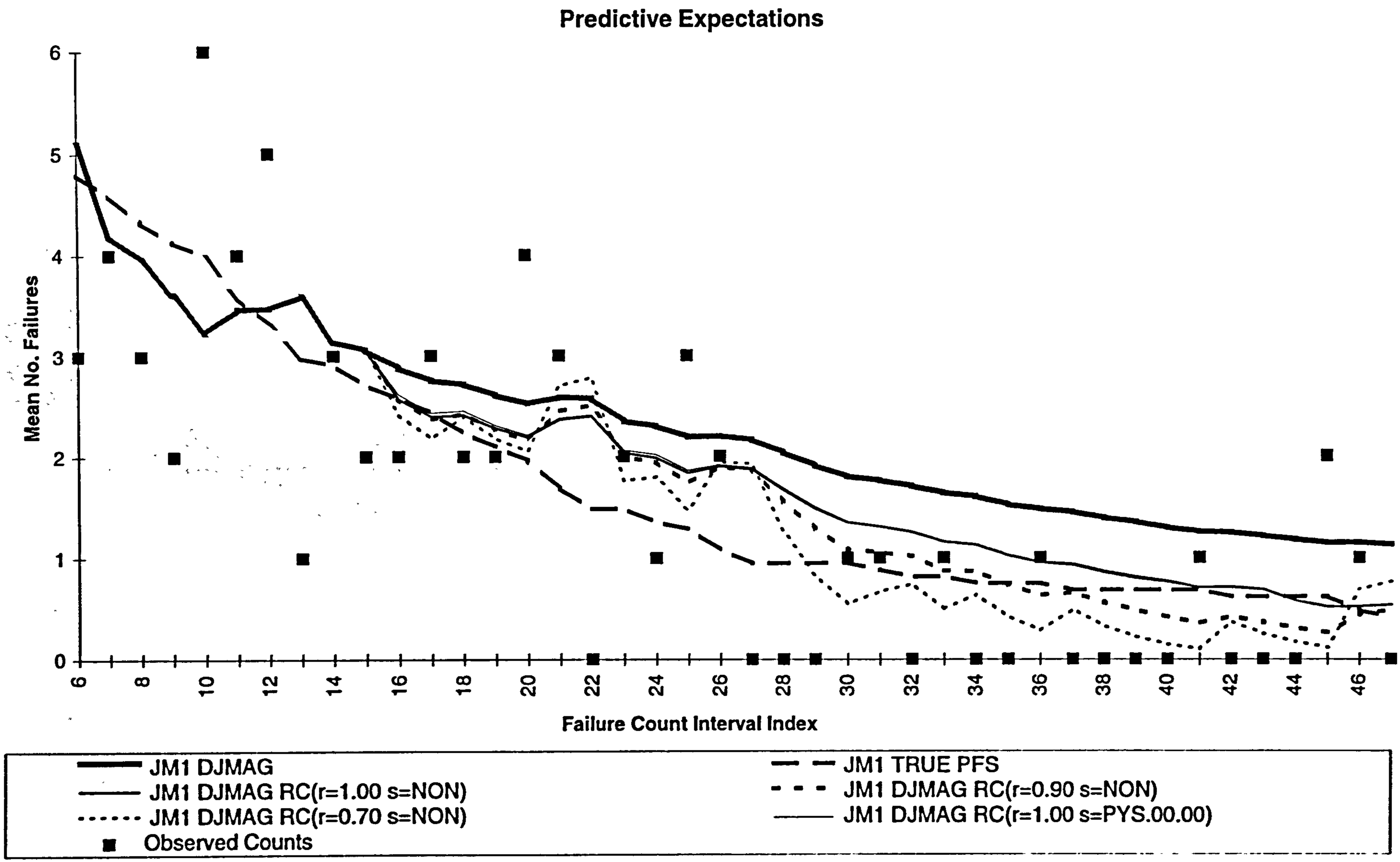

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
Failure Count Interval Index
JM1 DJMAG
JM1 DJMAG RC(r=1.00 s=NON)
JM1 DJMAG RC(r=0.70 s=NON)
Observed Counts
JM1 TRUE PFS
JM1 DJMAG RC(r=0.90 s=NON)
JM1 DJMAG RC(r=1.00 s=PYS.00.00)

Figure 24a

Discrete Log PLR -- vs. JM1 DJMAG

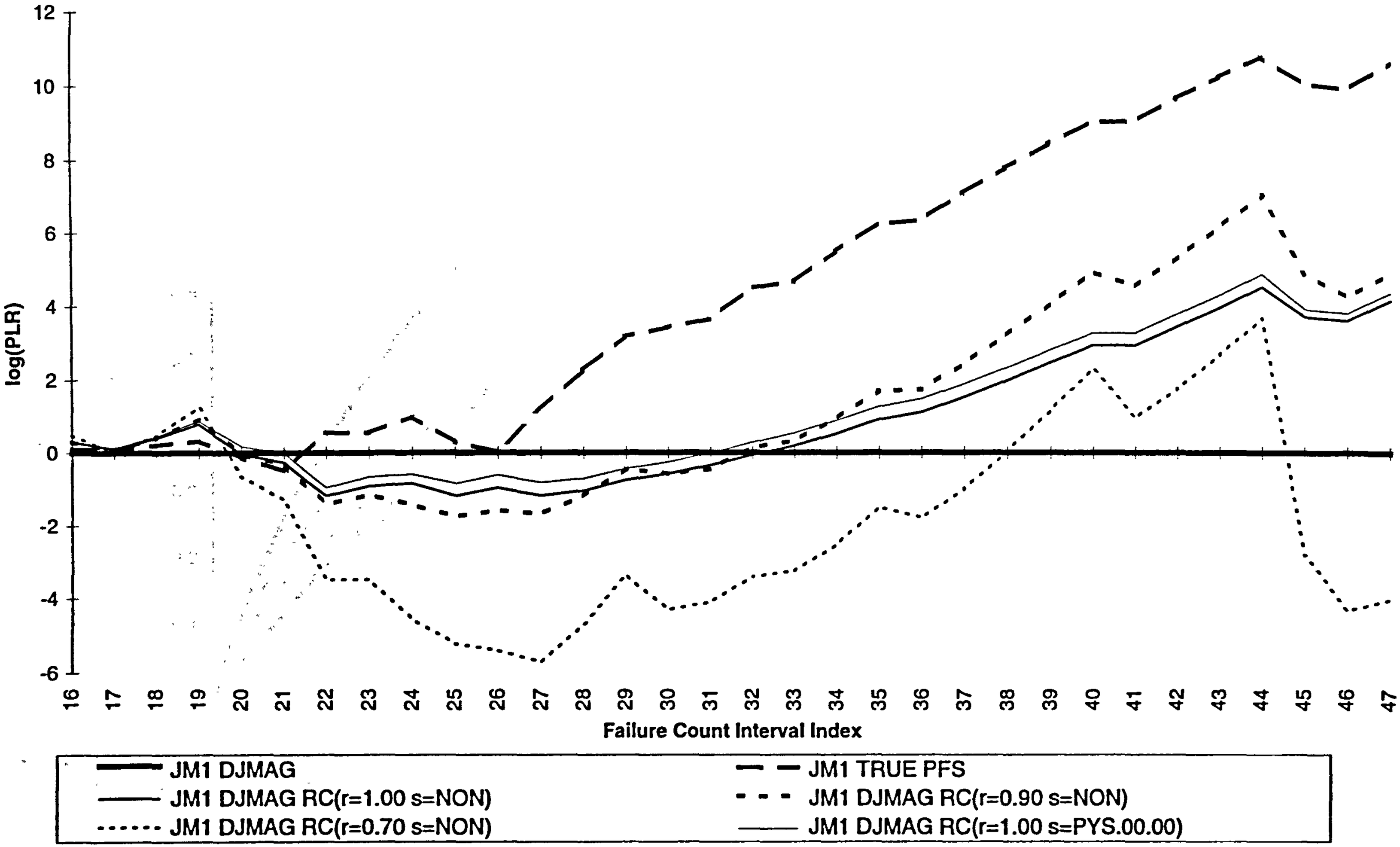


Figure 24b

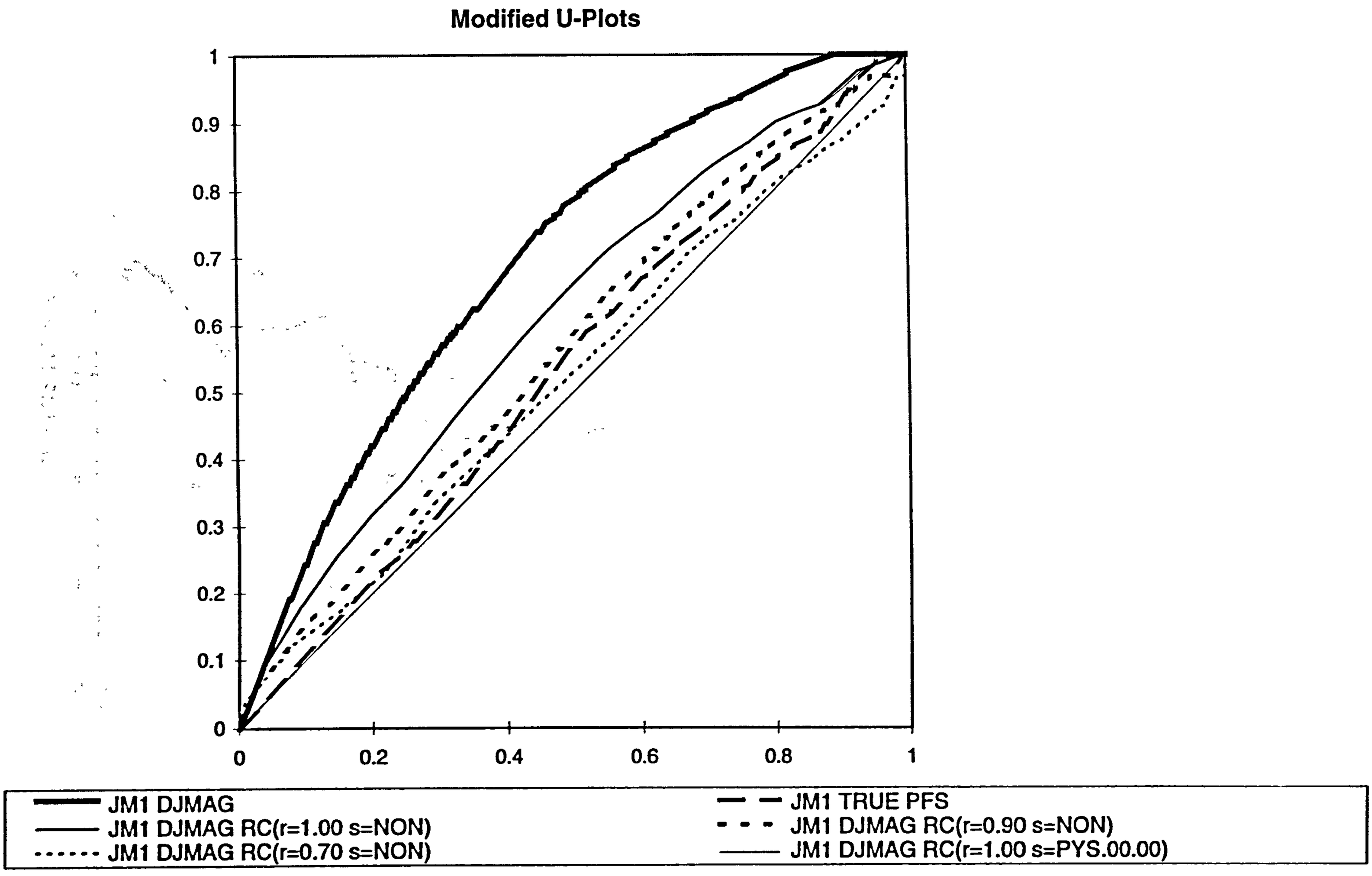

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
JM1 DJMAG
JM1 DJMAG RC(r=1.00 s=NON)
JM1 DJMAG RC(r=0.70 s=NON)
JM1 TRUE PFS
JM1 DJMAG RC(r=0.90 s=NON)
JM1 DJMAG RC(r=1.00 s=PYS.00.00)

Figure 24c

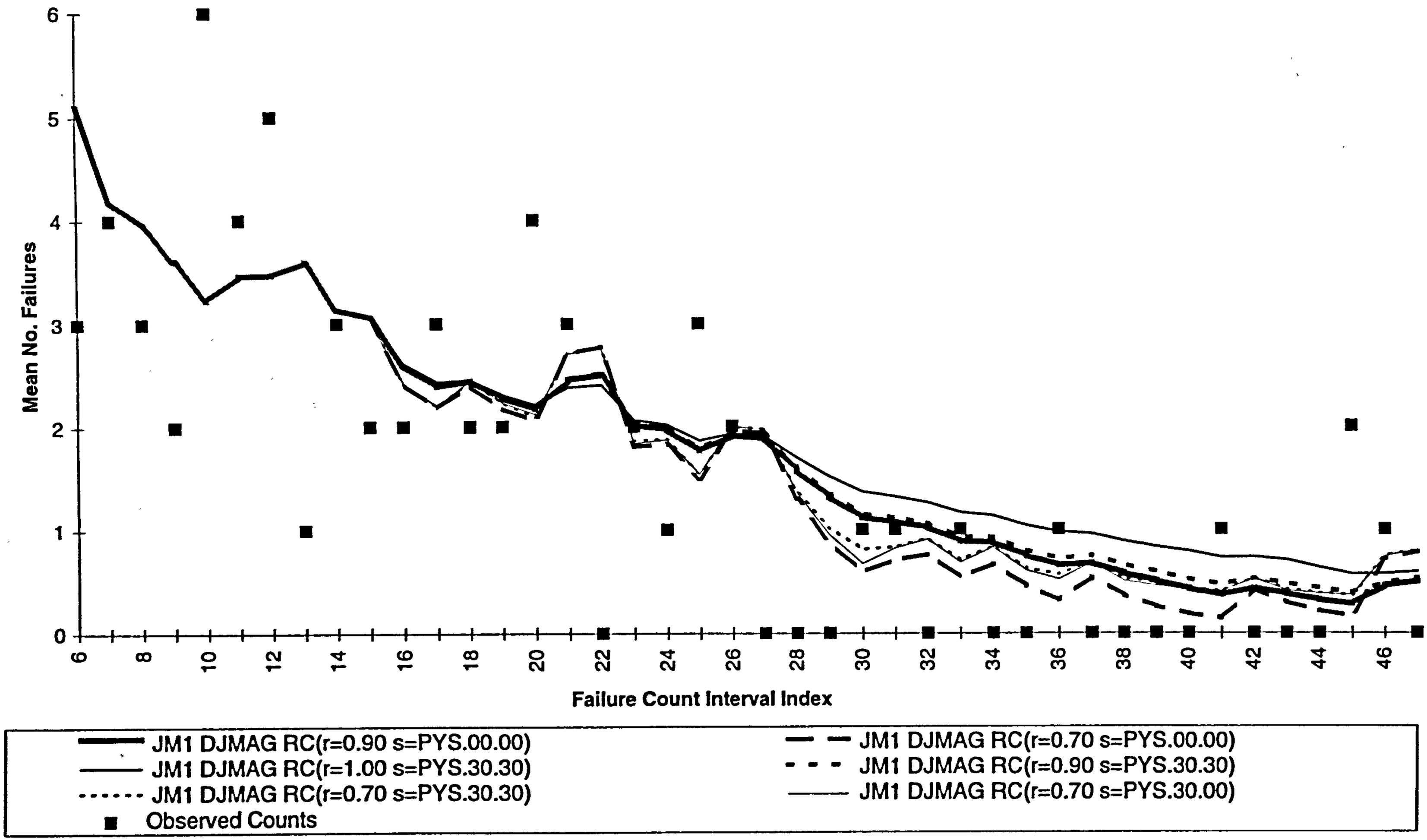

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
Failure Count Interval Index
JM1 DJMAG RC(r=0.90 s=PYS.00.00)
JM1 DJMAG RC(r=1.00 s=PYS.30.30)
JM1 DJMAG RC(r=0.70 s=PYS.30.30)
Observed Counts
JM1 DJMAG RC(r=0.70 s=PYS.00.00)
JM1 DJMAG RC(r=0.90 s=PYS.30.30)
JM1 DJMAG RC(r=0.70 s=PYS.30.00)

Figure 25a

Discrete Log PLR -- vs. JM1 DJMAG

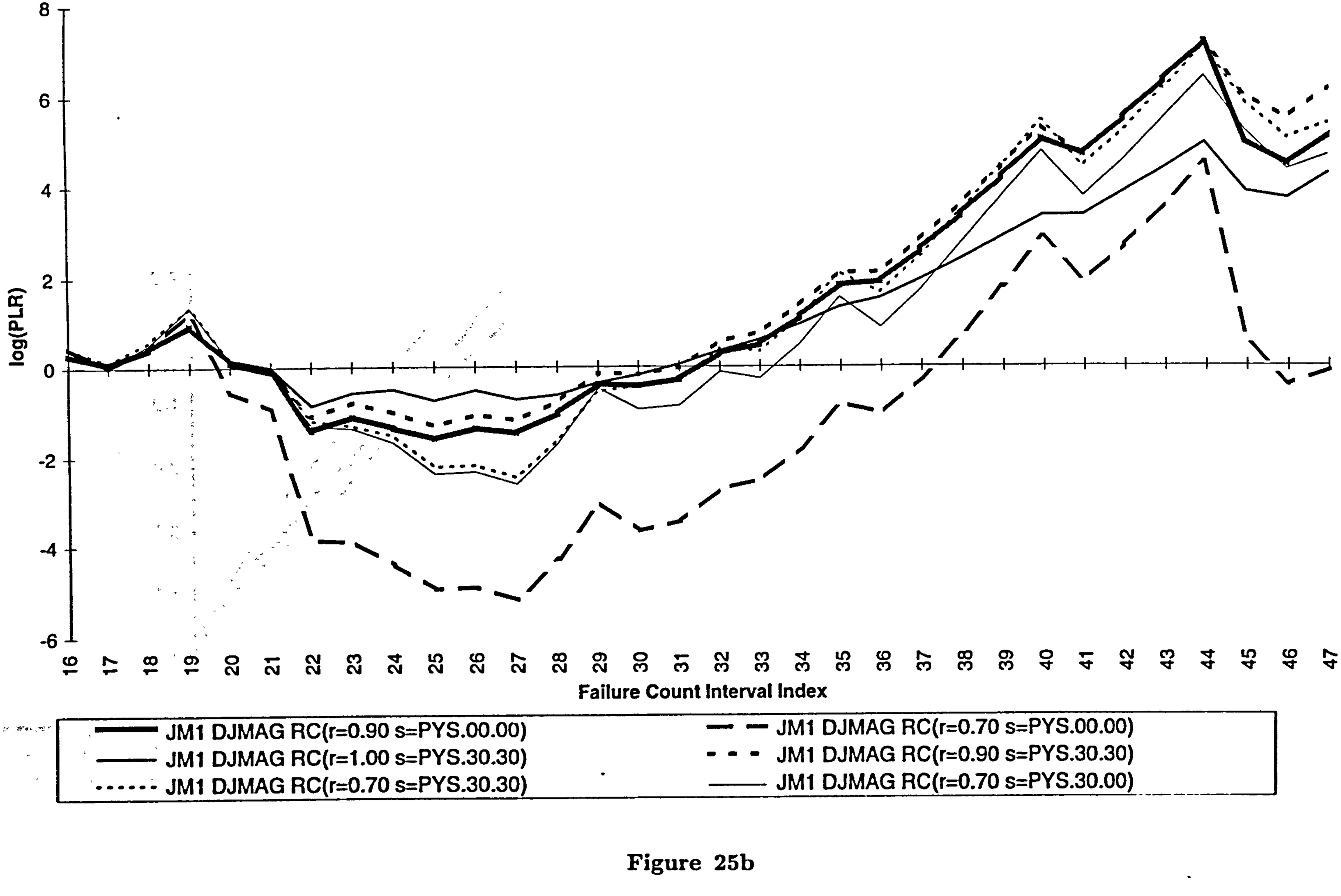


Figure 25b

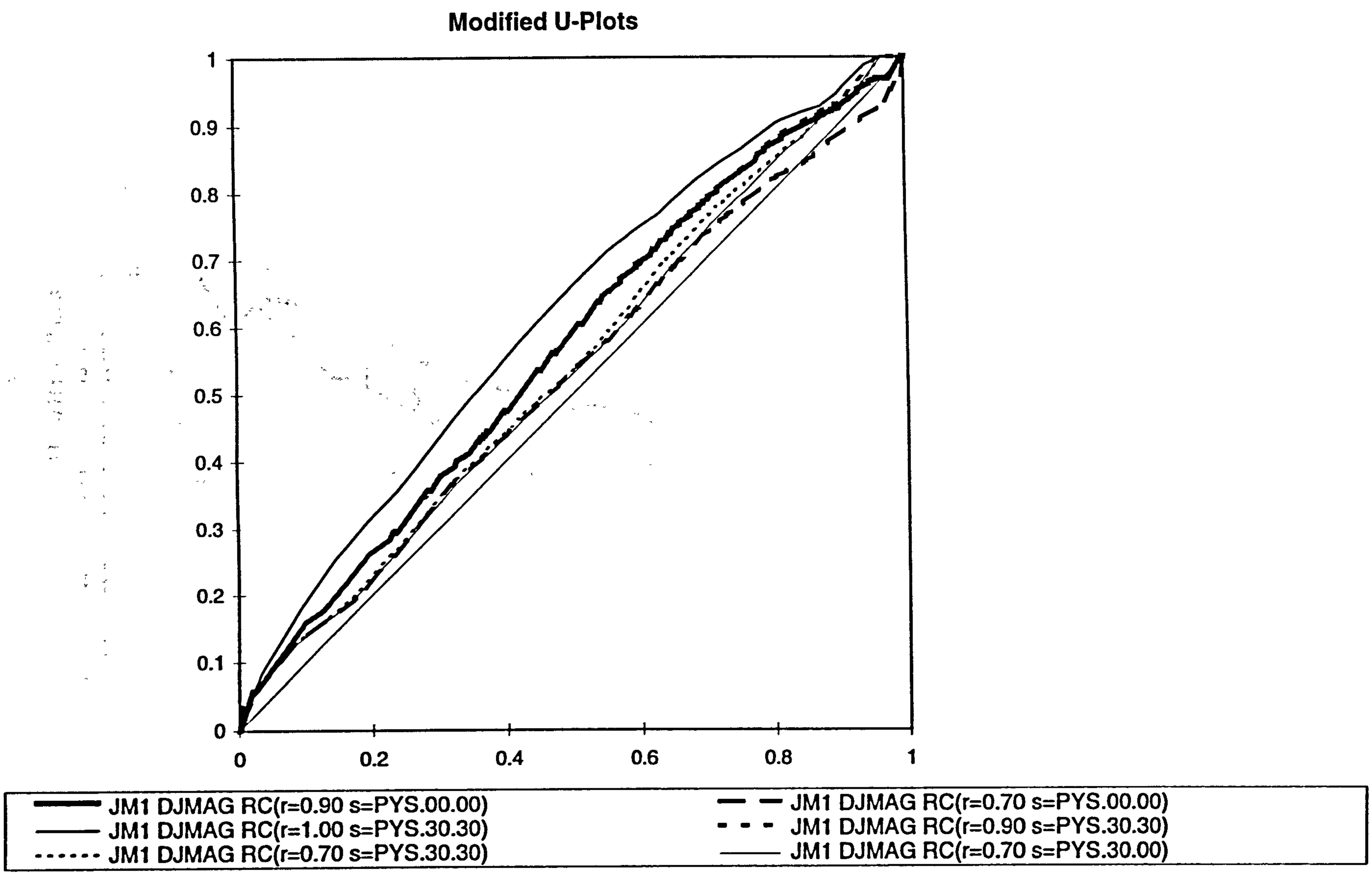

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
JM1 DJMAG RC(r=0.90 s=PYS.00.00)
JM1 DJMAG RC(r=1.00 s=PYS.30.30)
JM1 DJMAG RC(r=0.70 s=PYS.30.30)
JM1 DJMAG RC(r=0.70 s=PYS.00.00)
JM1 DJMAG RC(r=0.90 s=PYS.30.30)
JM1 DJMAG RC(r=0.70 s=PYS.30.00)

Figure 25c

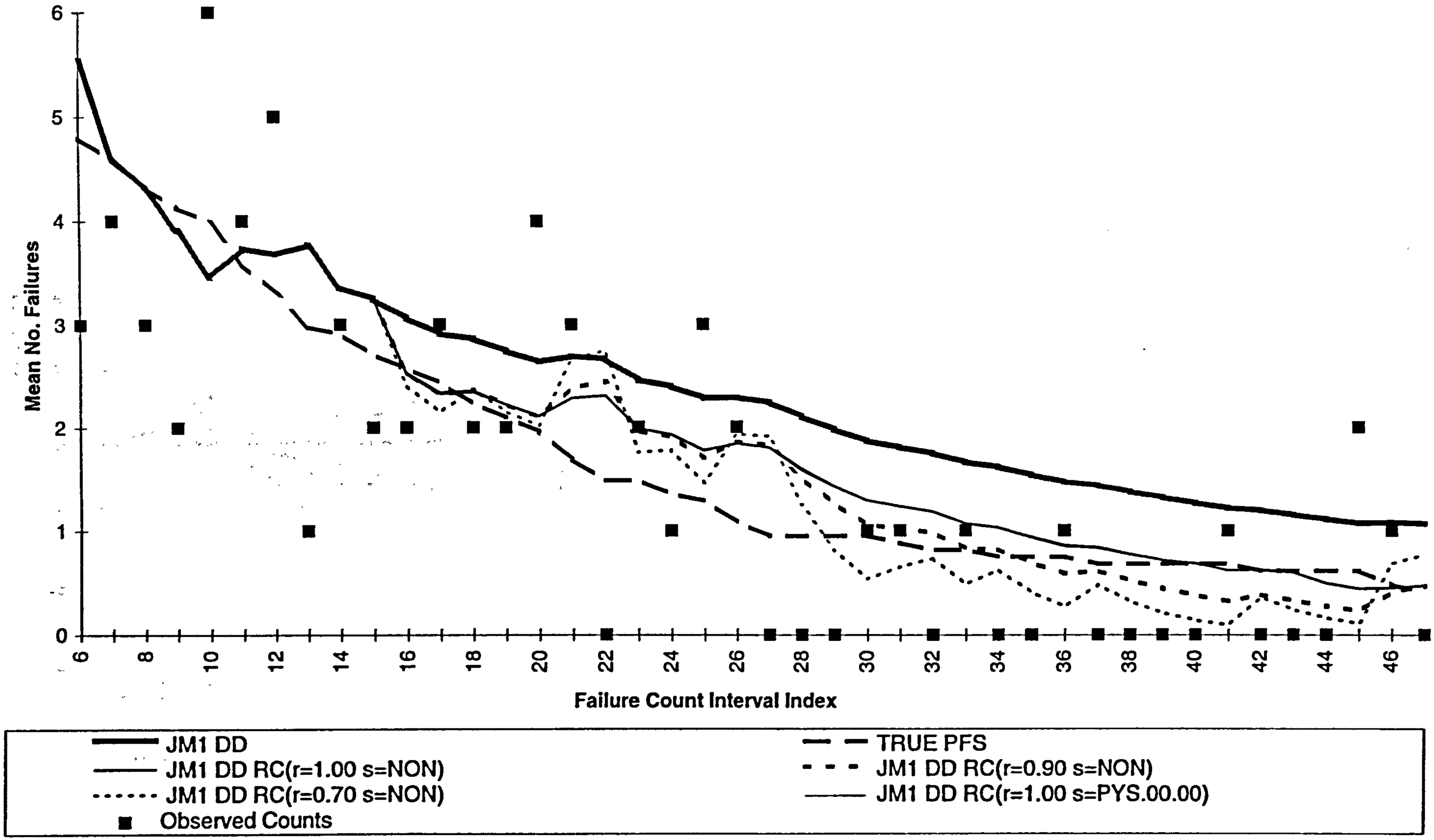

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
Failure Count Interval Index
JM1 DD
JM1 DD RC(r=1.00 s=NON)
JM1 DD RC(r=0.70 s=NON)
Observed Counts
TRUE PFS
JM1 DD RC(r=0.90 s=NON)
JM1 DD RC(r=1.00 s=PYS.00.00)

Figure 26a

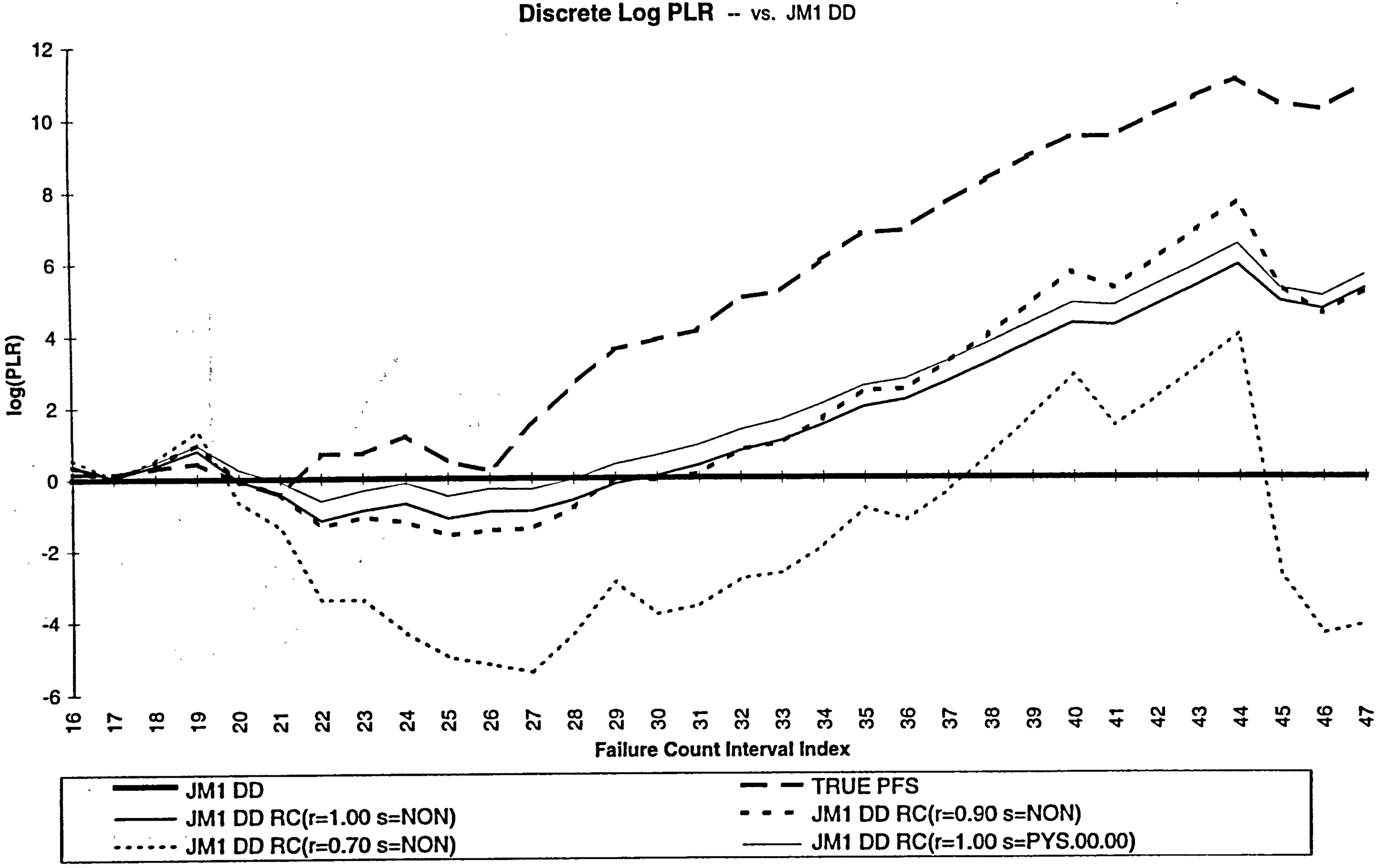

Discrete Log PLR -- vs. JM1 DD
log(PLR)
12
10
8
6
4
2
0
-2
-4
-6
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
Failure Count Interval Index
JM1 DD
JM1 DD RC(r=1.00 s=NON)
JM1 DD RC(r=0.70 s=NON)
TRUE PFS
JM1 DD RC(r=0.90 s=NON)
JM1 DD RC(r=1.00 s=PYS.00.00)

Figure 26b

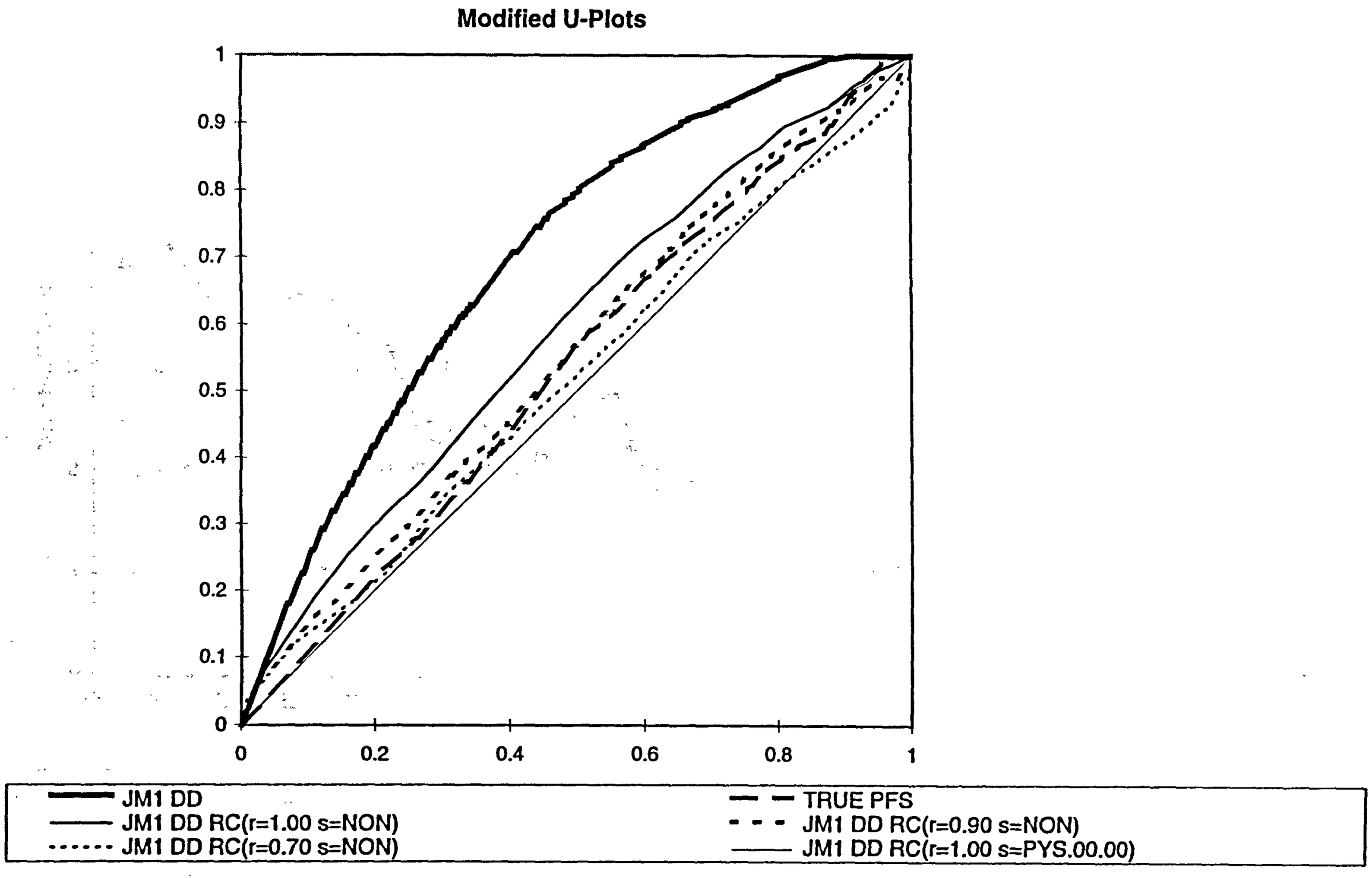

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
JM1 DD
JM1 DD RC(r=1.00 s=NON)
JM1 DD RC(r=0.70 s=NON)
TRUE PFS
JM1 DD RC(r=0.90 s=NON)
JM1 DD RC(r=1.00 s=PYS.00.00)

Figure 26c

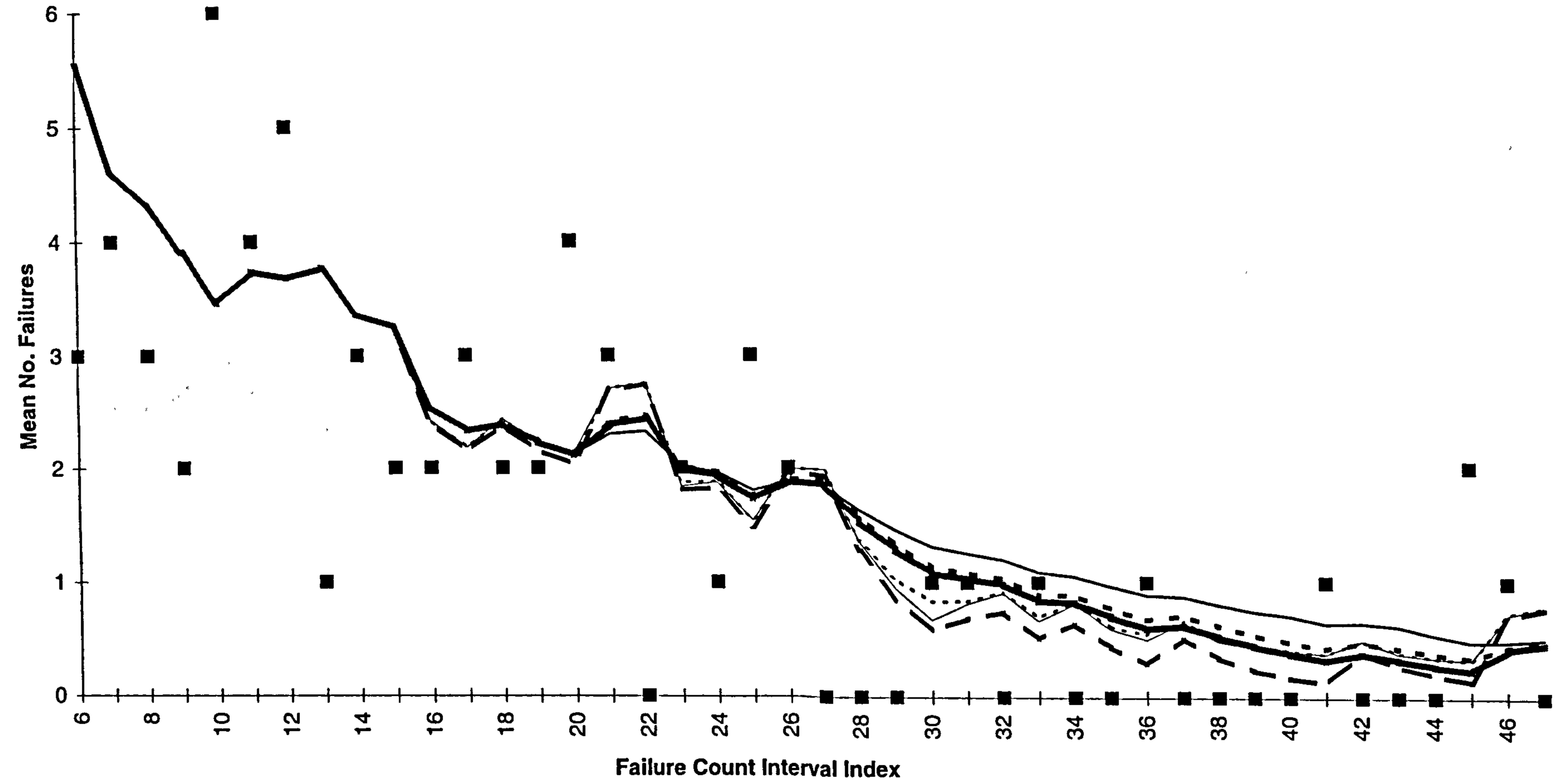

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
Failure Count Interval Index
JM1 DD RC(r=0.90 s=PYS.00.00)
JM1 DD RC(r=1.00 s=PYS.30.30)
JM1 DD RC(r=0.70 s=PYS.30.30)
Observed Counts
JM1 DD RC(r=0.70 s=PYS.00.00)
JM1 DD RC(r=0.90 s=PYS.30.30)
JM1 DD RC(r=0.70 s=PYS.30.00)

Figure 27a

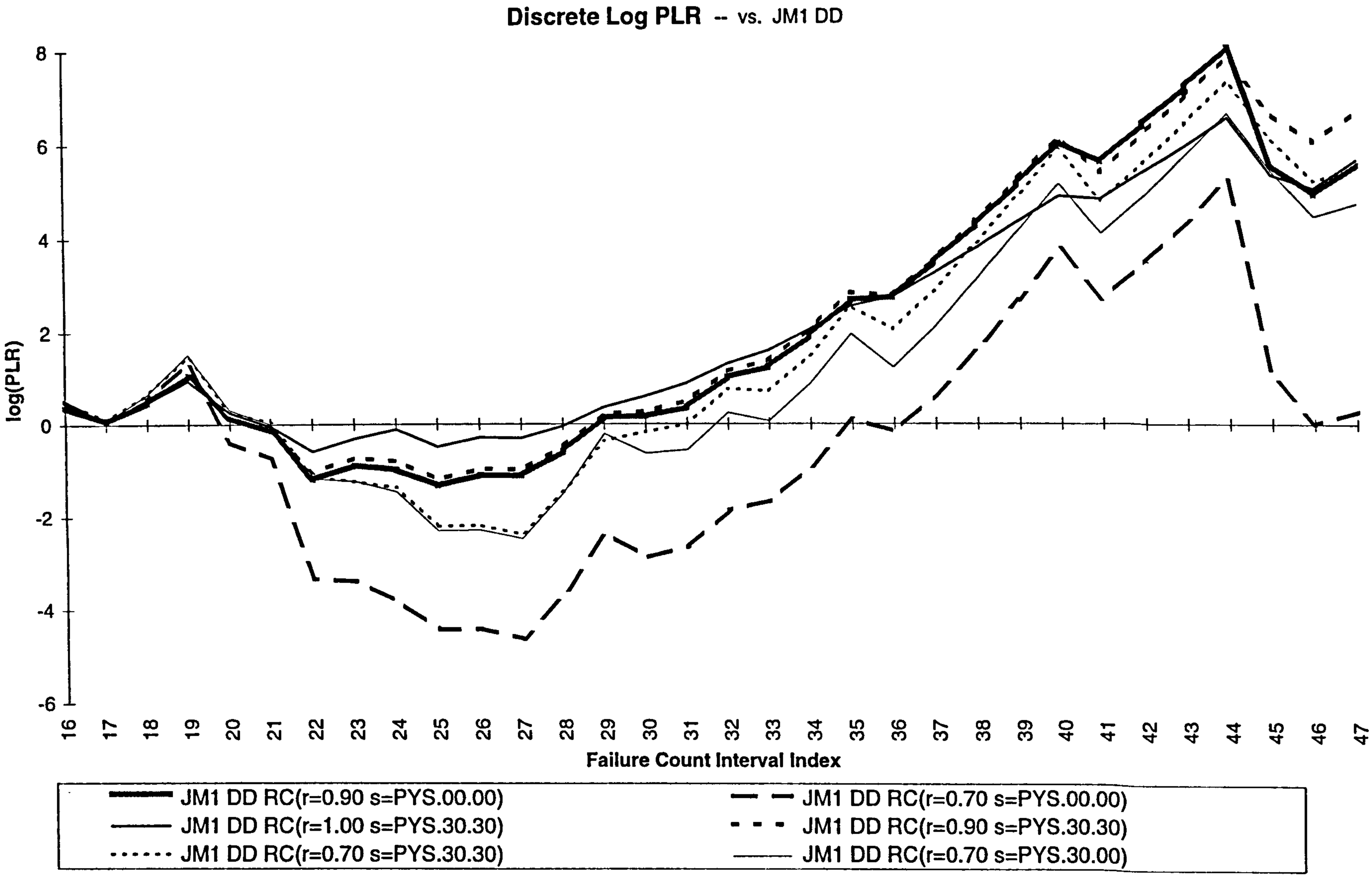

Discrete Log PLR -- vs. JM1 DD
log(PLR)
8
6
4
2
0
-2
-4
-6
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
Failure Count Interval Index
JM1 DD RC(r=0.90 s=PYS.00.00)
JM1 DD RC(r=1.00 s=PYS.30.30)
JM1 DD RC(r=0.70 s=PYS.30.30)
JM1 DD RC(r=0.70 s=PYS.00.00)
JM1 DD RC(r=0.90 s=PYS.30.30)
JM1 DD RC(r=0.70 s=PYS.30.00)

Figure 27b

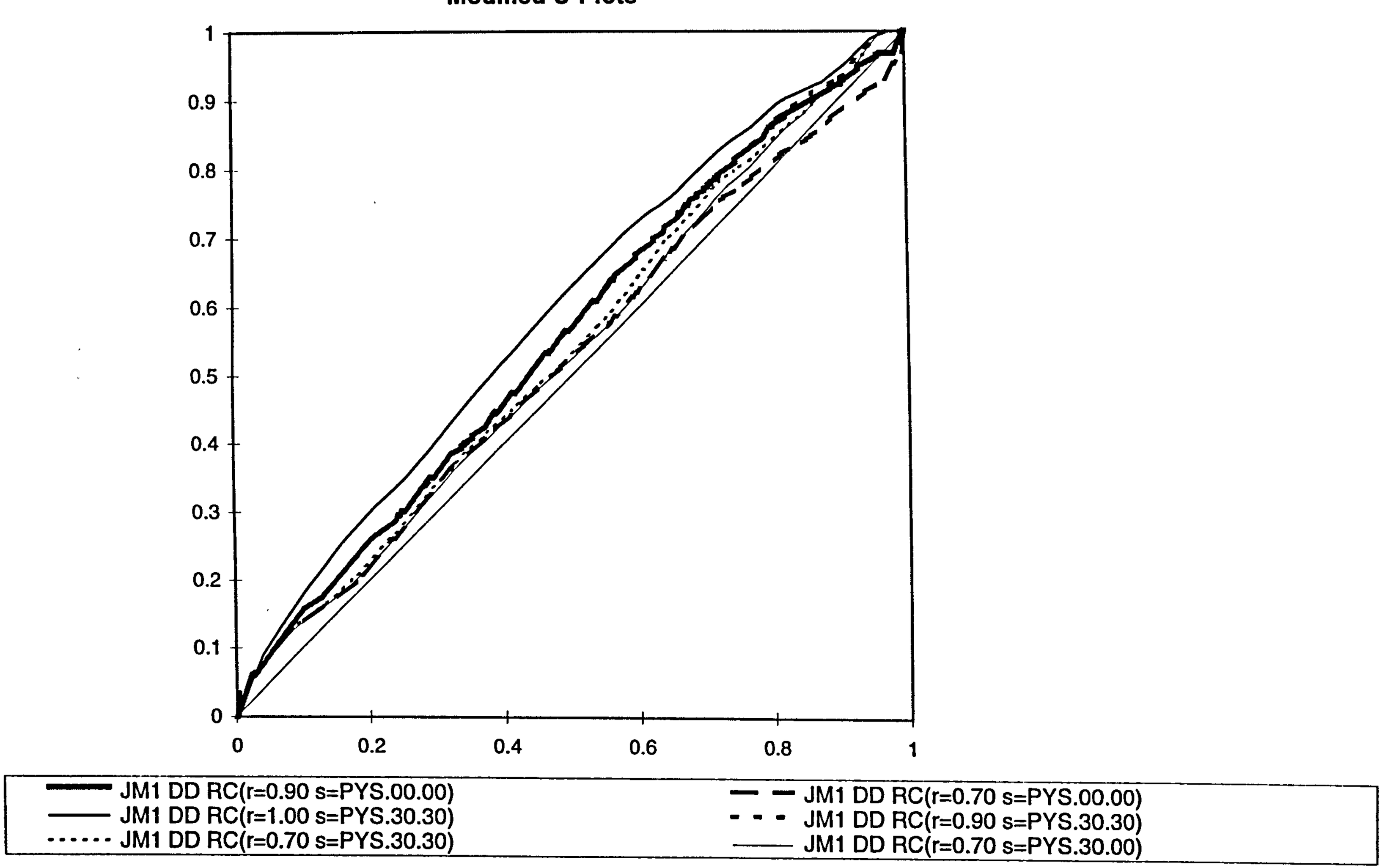

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
JM1 DD RC(r=0.90 s=PYS.00.00)
JM1 DD RC(r=1.00 s=PYS.30.30)
JM1 DD RC(r=0.70 s=PYS.30.30)
JM1 DD RC(r=0.70 s=PYS.00.00)
JM1 DD RC(r=0.90 s=PYS.30.30)
JM1 DD RC(r=0.70 s=PYS.30.00)

Figure 27c

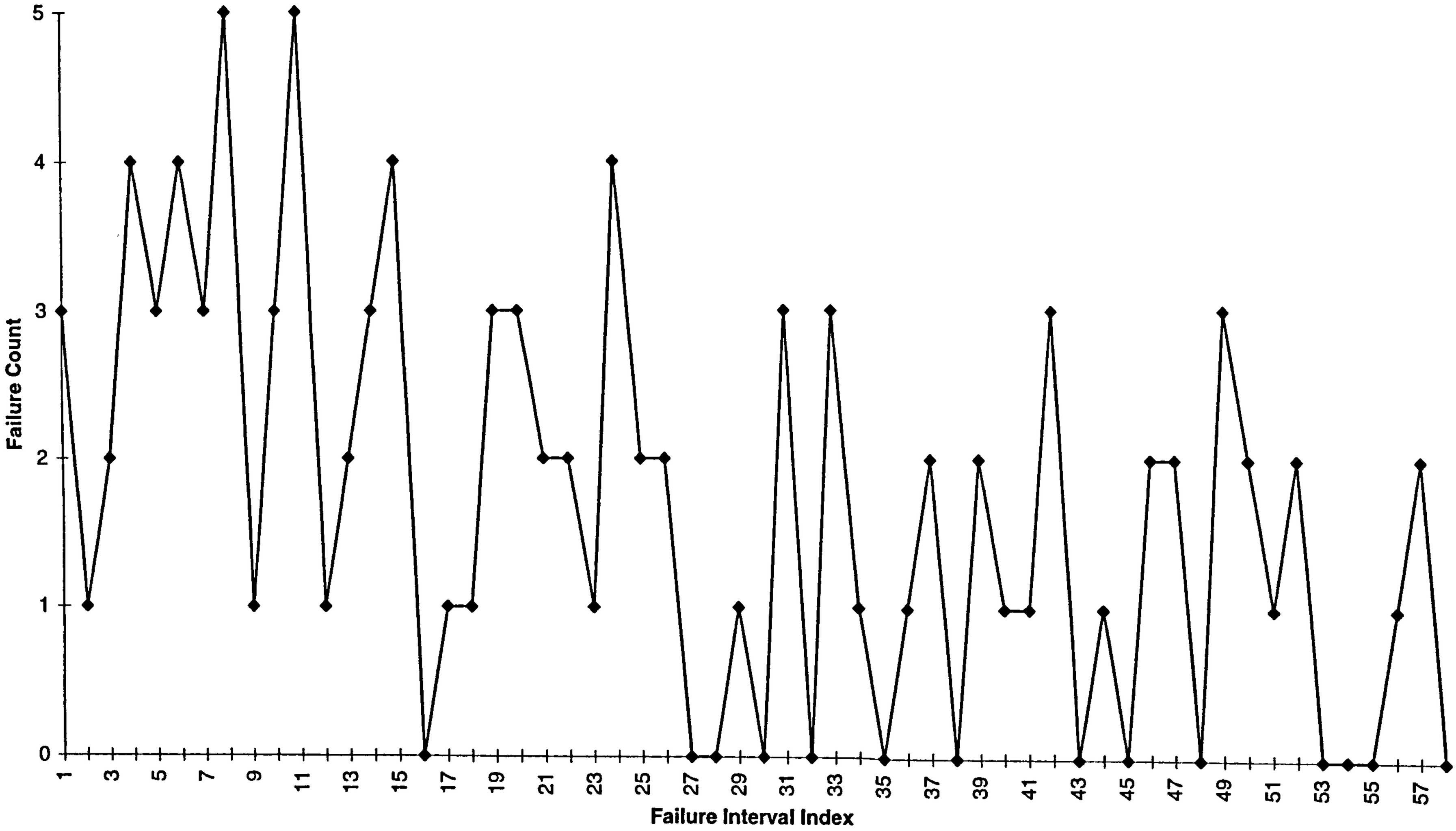

Failure Count Data: L1 - Simulated. alpha=1, beta=50,000, Interval Length 1,000
Failure Count
0
1
2
3
4
5
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
57
Failure Interval Index

Figure 28

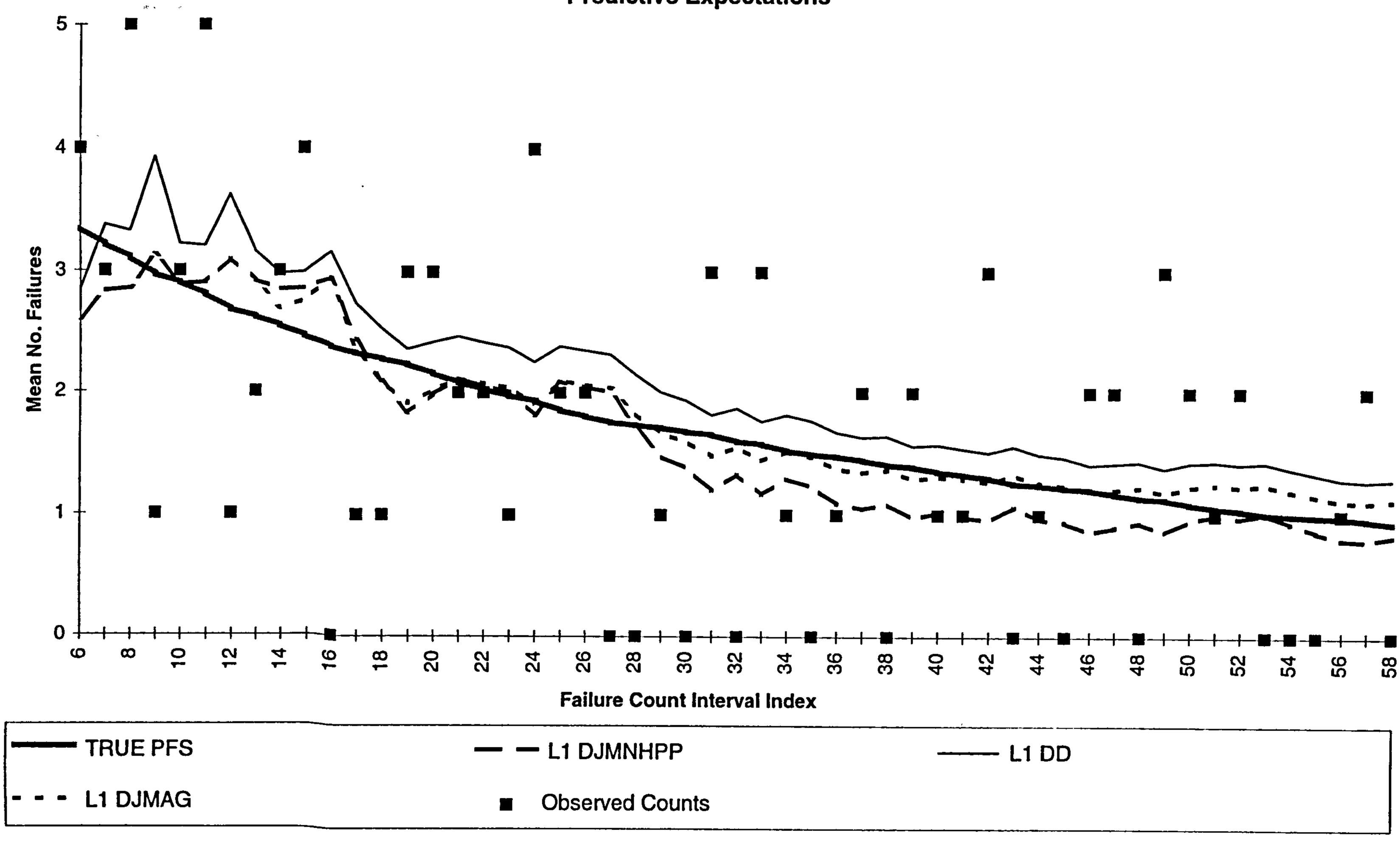

Predictive Expectations
Mean No. Failures
5
4
3
2
1
0
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
Failure Count Interval Index
TRUE PFS
L1 DJMNHPP
L1 DD
L1 DJMAG
Observed Counts

Figure 29a

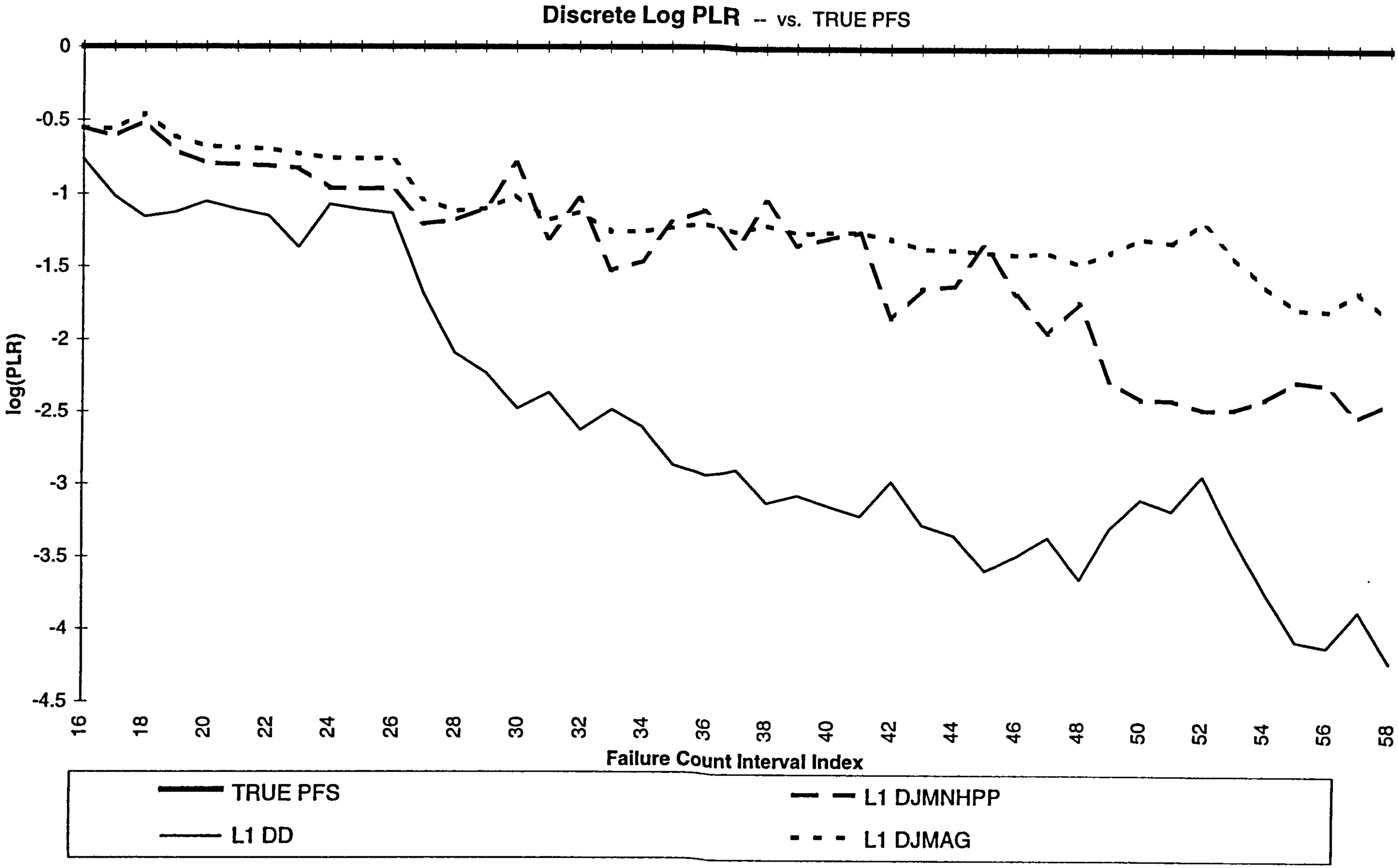

Discrete Log PLR -- vs. TRUE PFS
log(PLR)
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
Failure Count Interval Index
TRUE PFS
L1 DD
L1 DJMNHPP
L1 DJMAG

Figure 29b

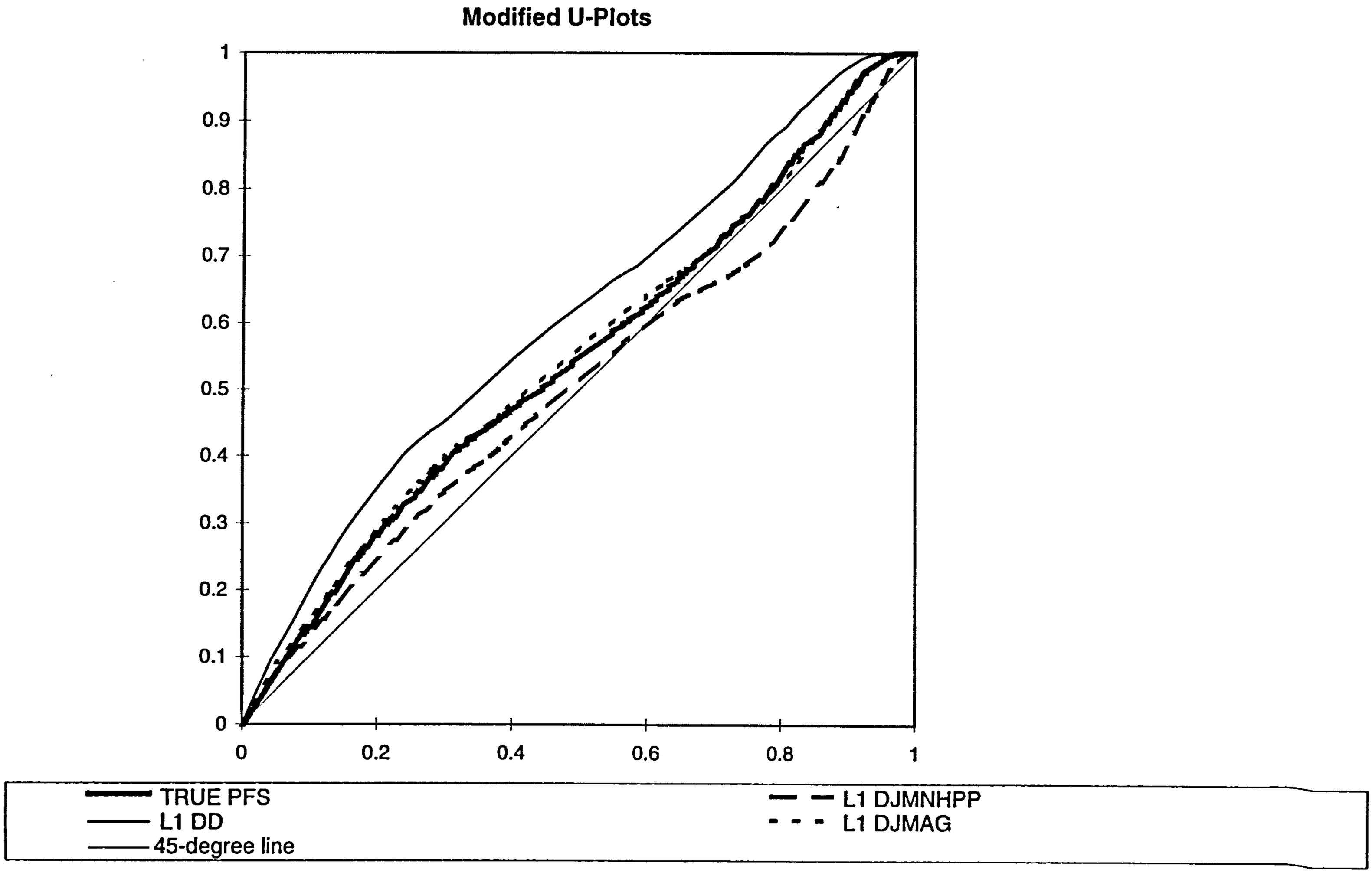

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
TRUE PFS
L1 DD
45-degree line
L1 DJMNHPP
L1 DJMAG

Figure 29c

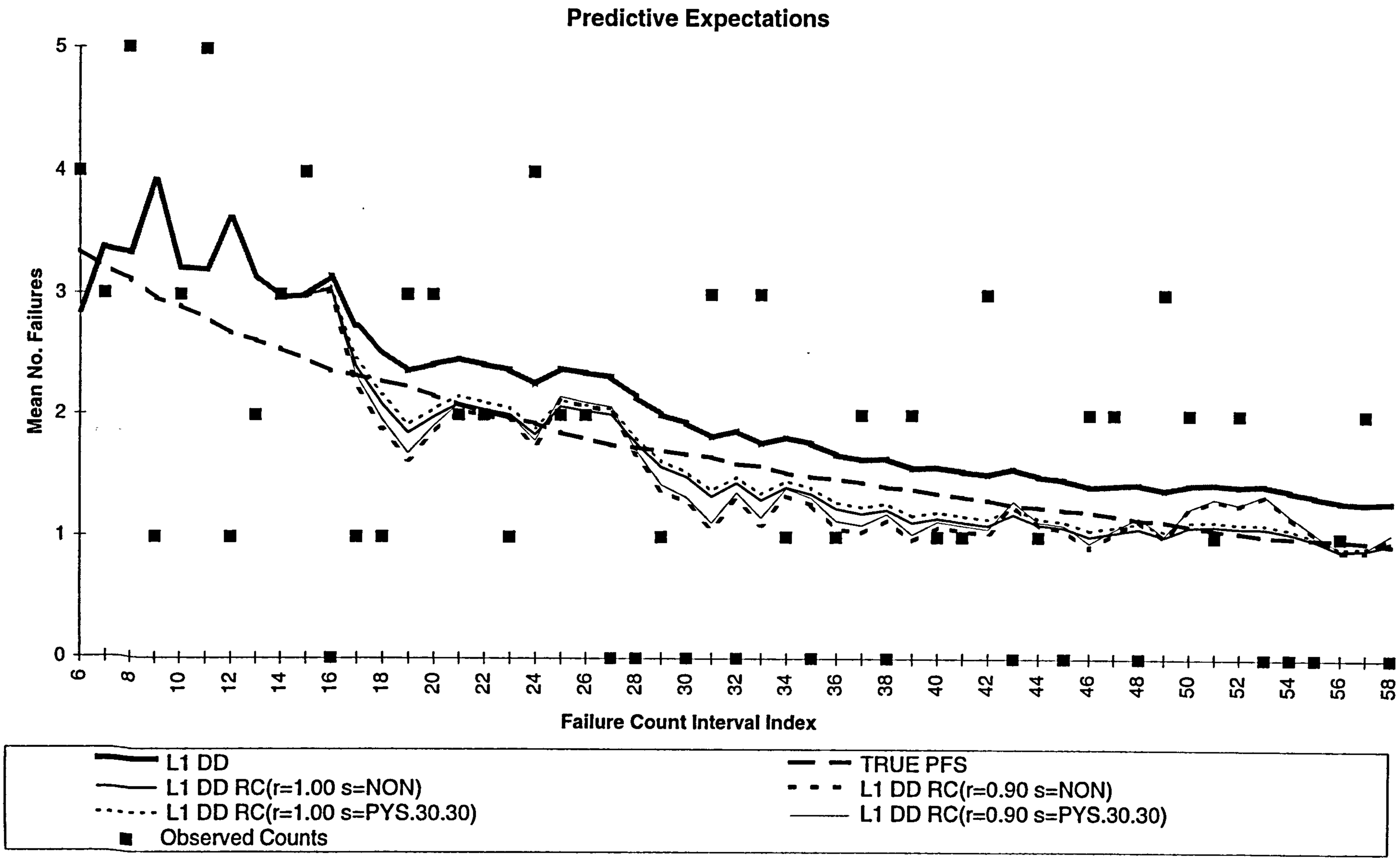

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
Failure Count Interval Index
L1 DD
L1 DD RC(r=1.00 s=NON)
L1 DD RC(r=1.00 s=PYS.30.30)
Observed Counts
TRUE PFS
L1 DD RC(r=0.90 s=NON)
L1 DD RC(r=0.90 s=PYS.30.30)

Figure 30a

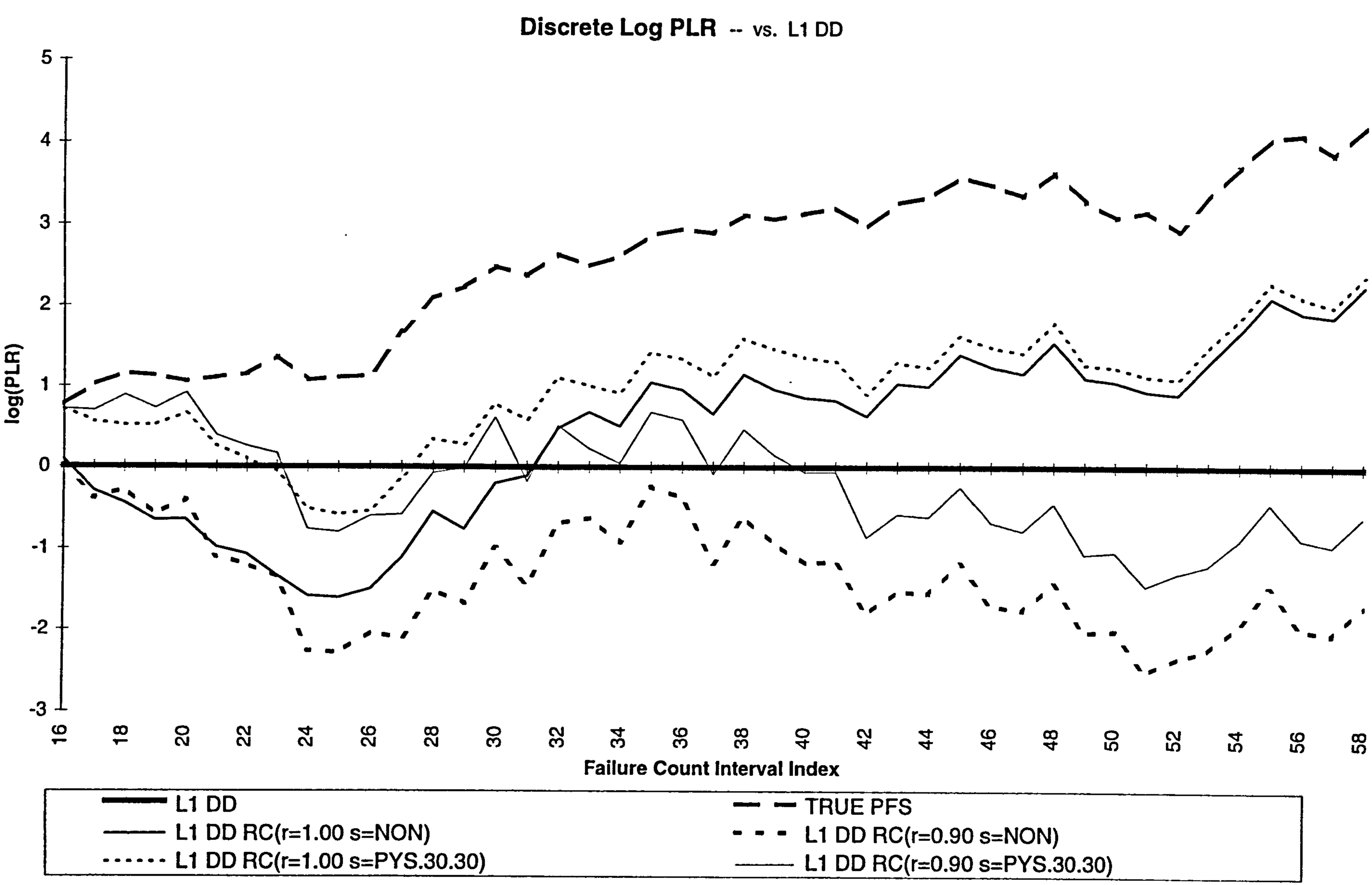

Discrete Log PLR -- vs. L1 DD
log(PLR)
5
4
3
2
1
0
-1
-2
-3
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
Failure Count Interval Index
L1 DD
L1 DD RC(r=1.00 s=NON)
L1 DD RC(r=1.00 s=PYS.30.30)
TRUE PFS
L1 DD RC(r=0.90 s=NON)
L1 DD RC(r=0.90 s=PYS.30.30)

Figure 30b

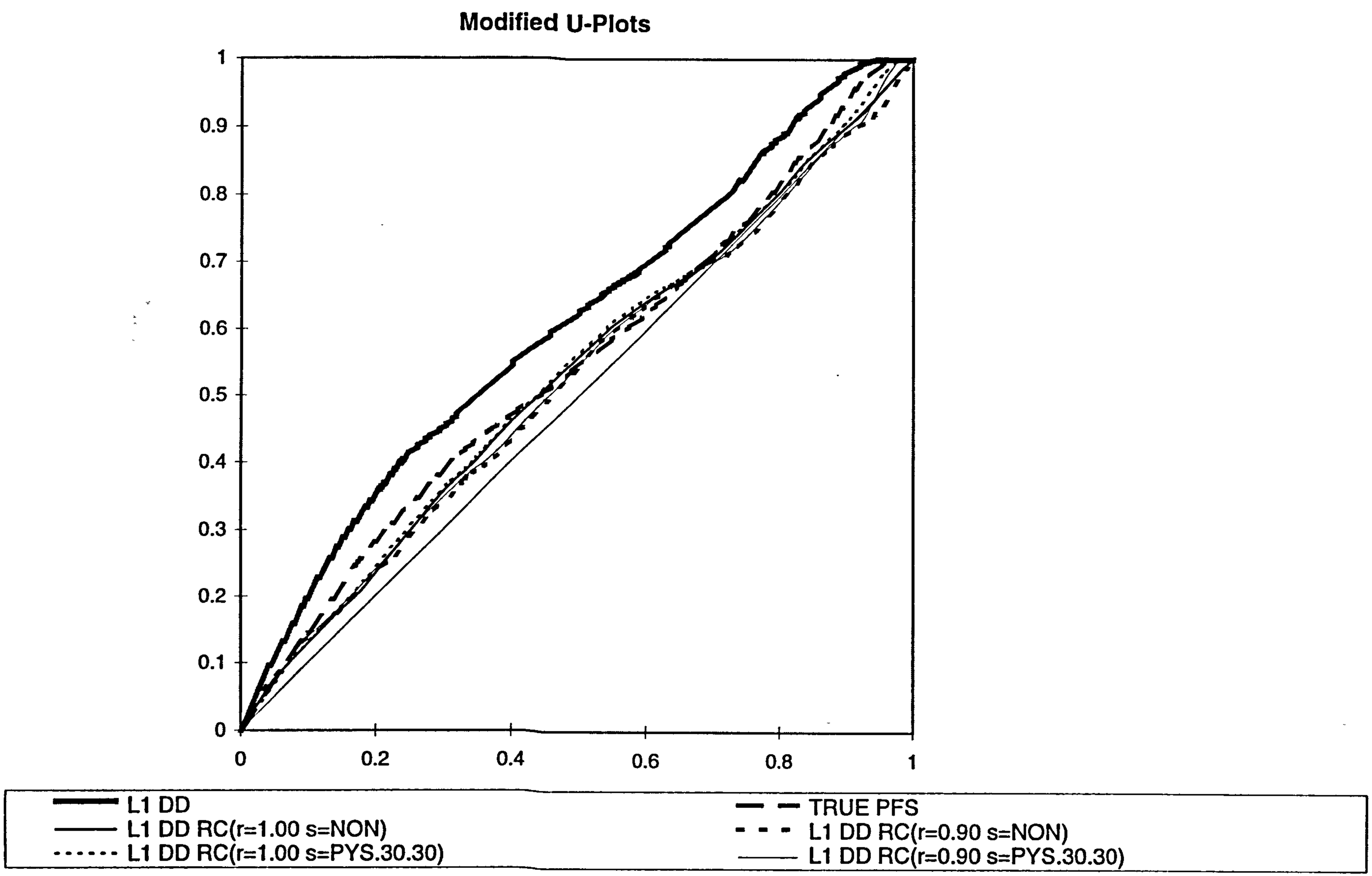

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
L1 DD
L1 DD RC(r=1.00 s=NON)
L1 DD RC(r=1.00 s=PYS.30.30)
TRUE PFS
L1 DD RC(r=0.90 s=NON)
L1 DD RC(r=0.90 s=PYS.30.30)

Figure 30c

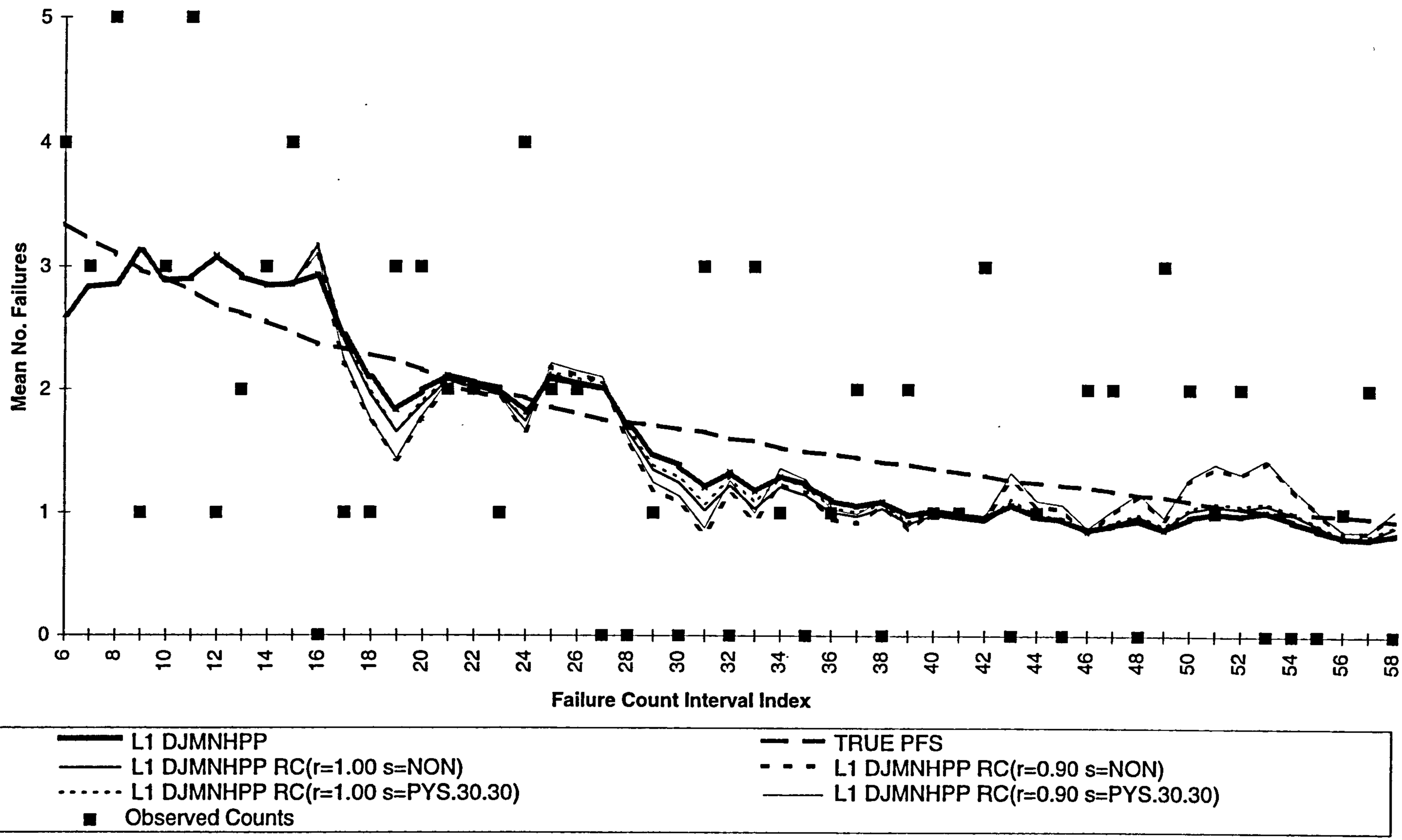

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
Failure Count Interval Index
L1 DJMNHPP
L1 DJMNHPP RC(r=1.00 s=NON)
L1 DJMNHPP RC(r=1.00 s=PYS.30.30)
Observed Counts
TRUE PFS
L1 DJMNHPP RC(r=0.90 s=NON)
L1 DJMNHPP RC(r=0.90 s=PYS.30.30)

Figure 31a

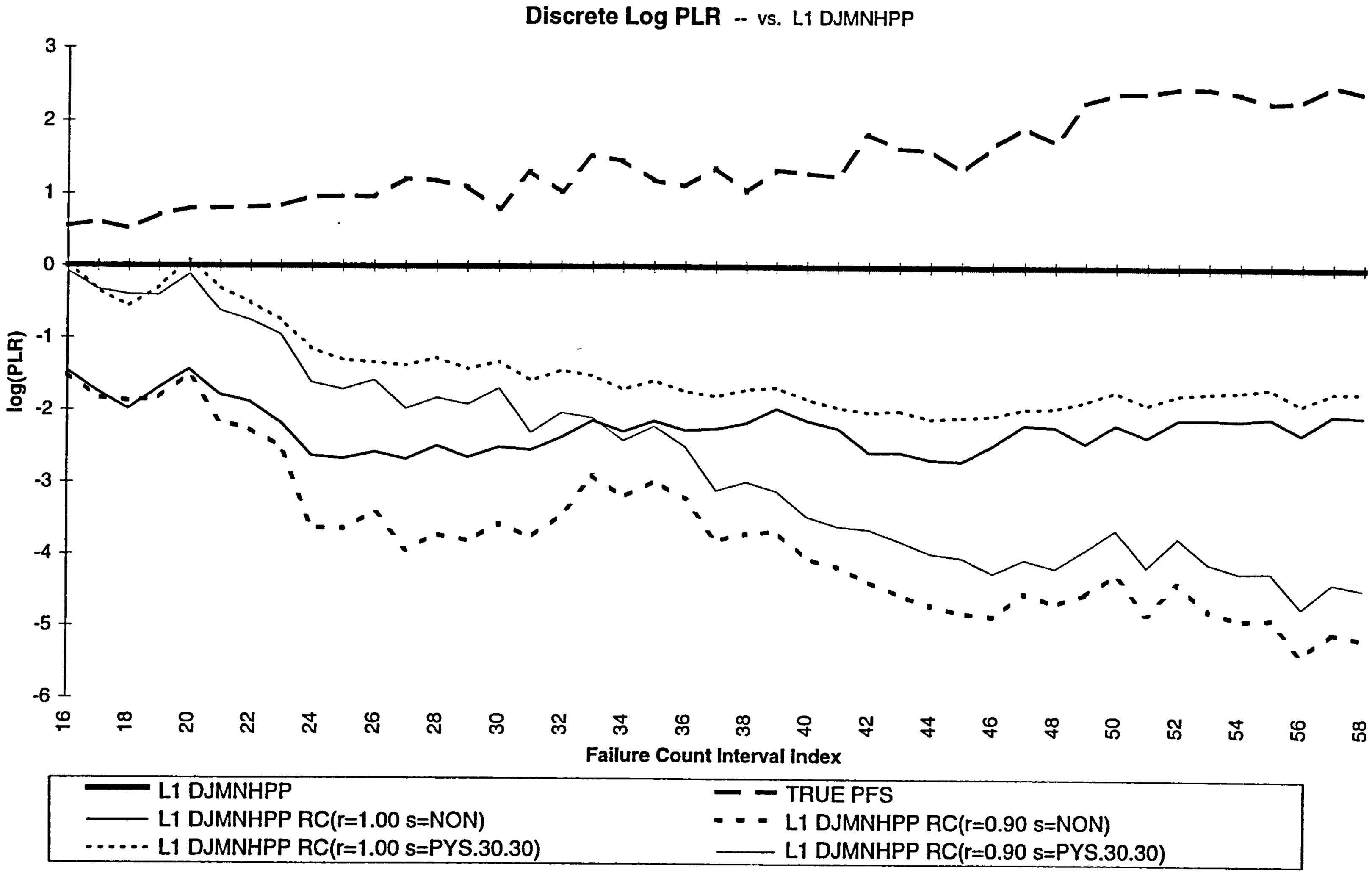

Discrete Log PLR -- vs. L1 DJMNHPP
log(PLR)
3
2
1
0
-1
-2
-3
-4
-5
-6
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
Failure Count Interval Index
L1 DJMNHPP
L1 DJMNHPP RC(r=1.00 s=NON)
L1 DJMNHPP RC(r=1.00 s=PYS.30.30)
TRUE PFS
L1 DJMNHPP RC(r=0.90 s=NON)
L1 DJMNHPP RC(r=0.90 s=PYS.30.30)

Figure 31b

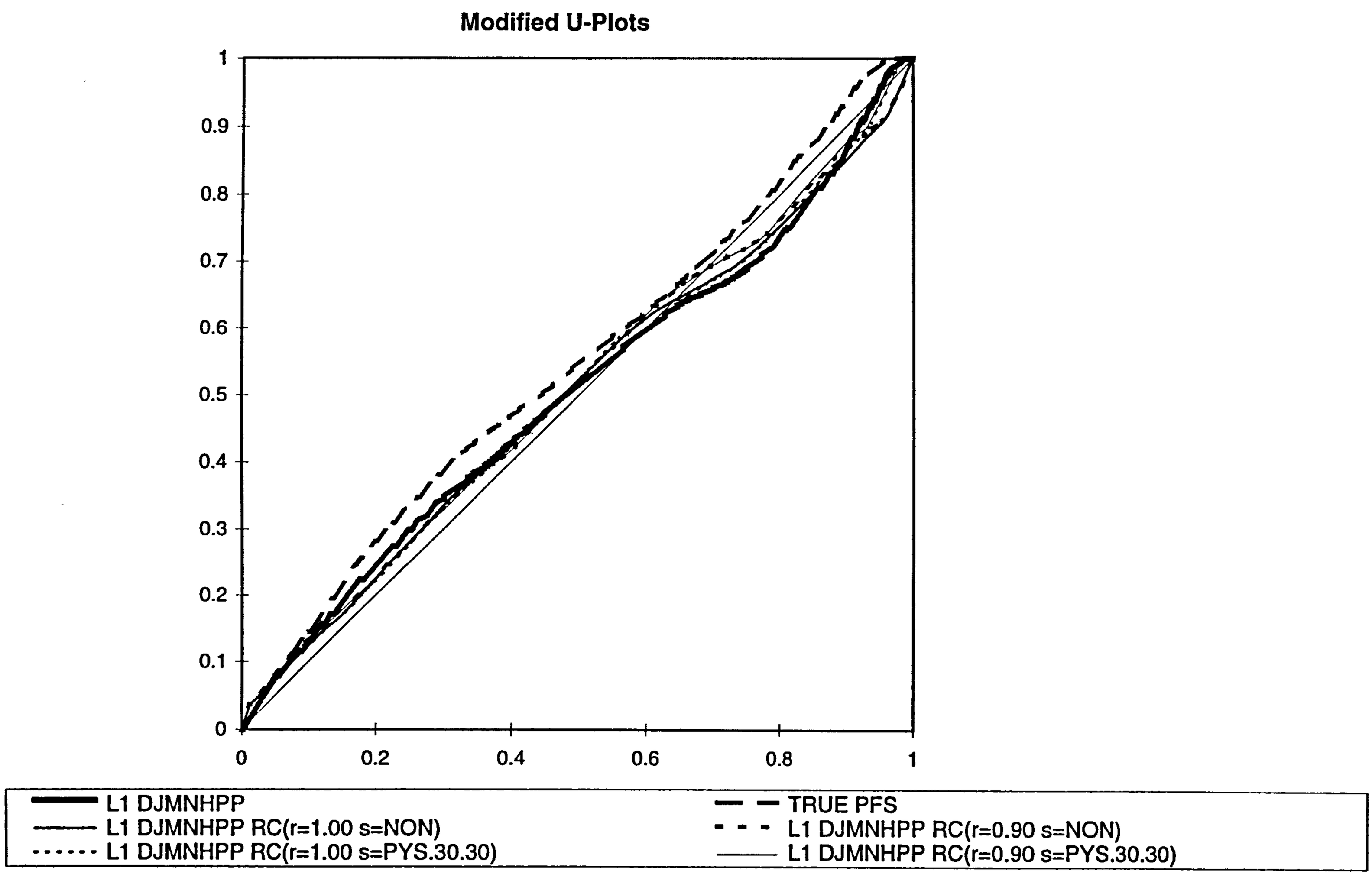

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
L1 DJMNHPP
TRUE PFS
L1 DJMNHPP RC(r=1.00 s=NON)
L1 DJMNHPP RC(r=0.90 s=NON)
L1 DJMNHPP RC(r=1.00 s=PYS.30.30)
L1 DJMNHPP RC(r=0.90 s=PYS.30.30)

Figure 31c

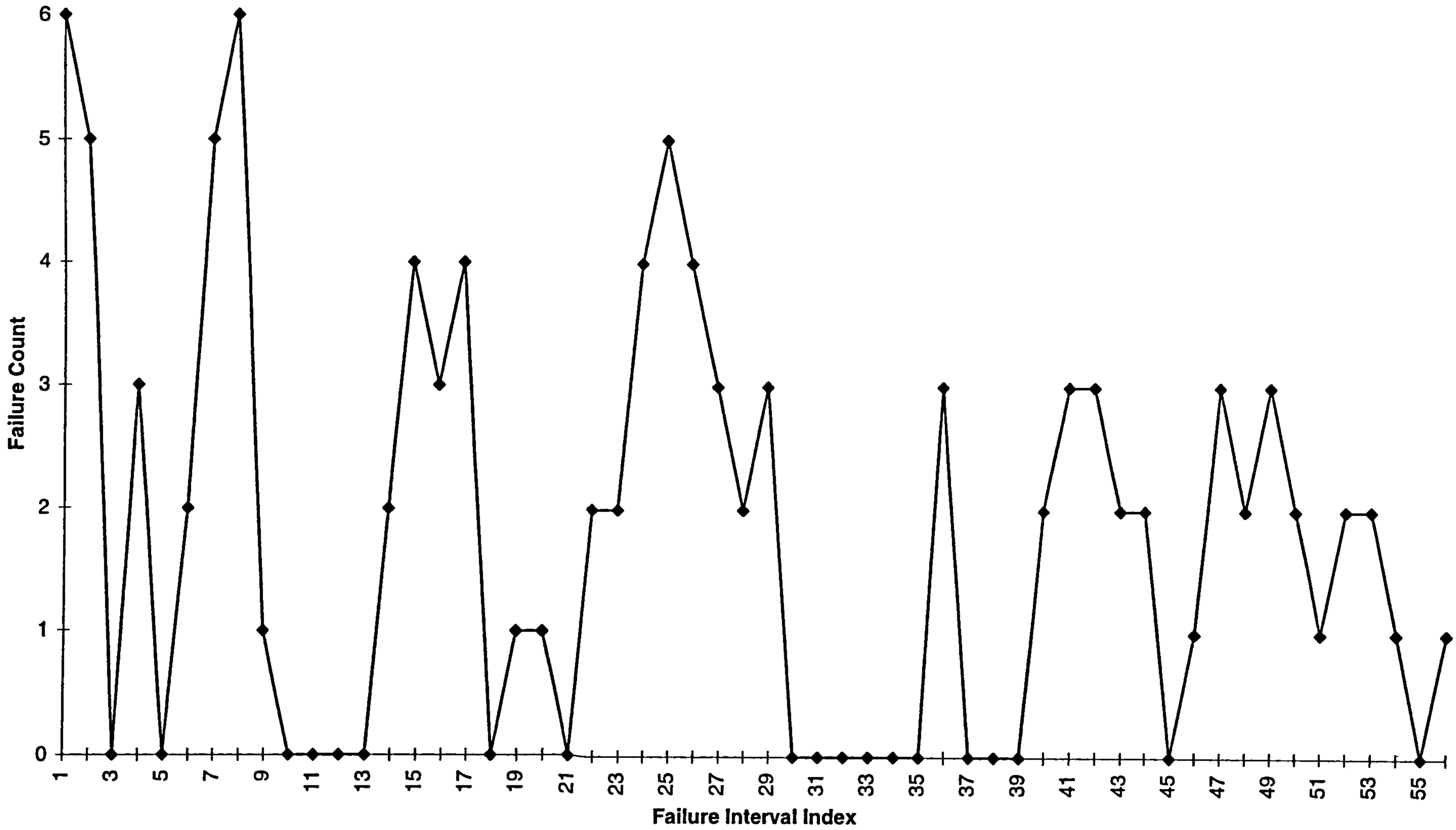

Failure Count Data: LV1 - Simulated. alpha=1.5, beta1=30, beta2=5, Interval Length 500
Failure Count
0
1
2
3
4
5
6
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
Failure Interval Index

Figure 32

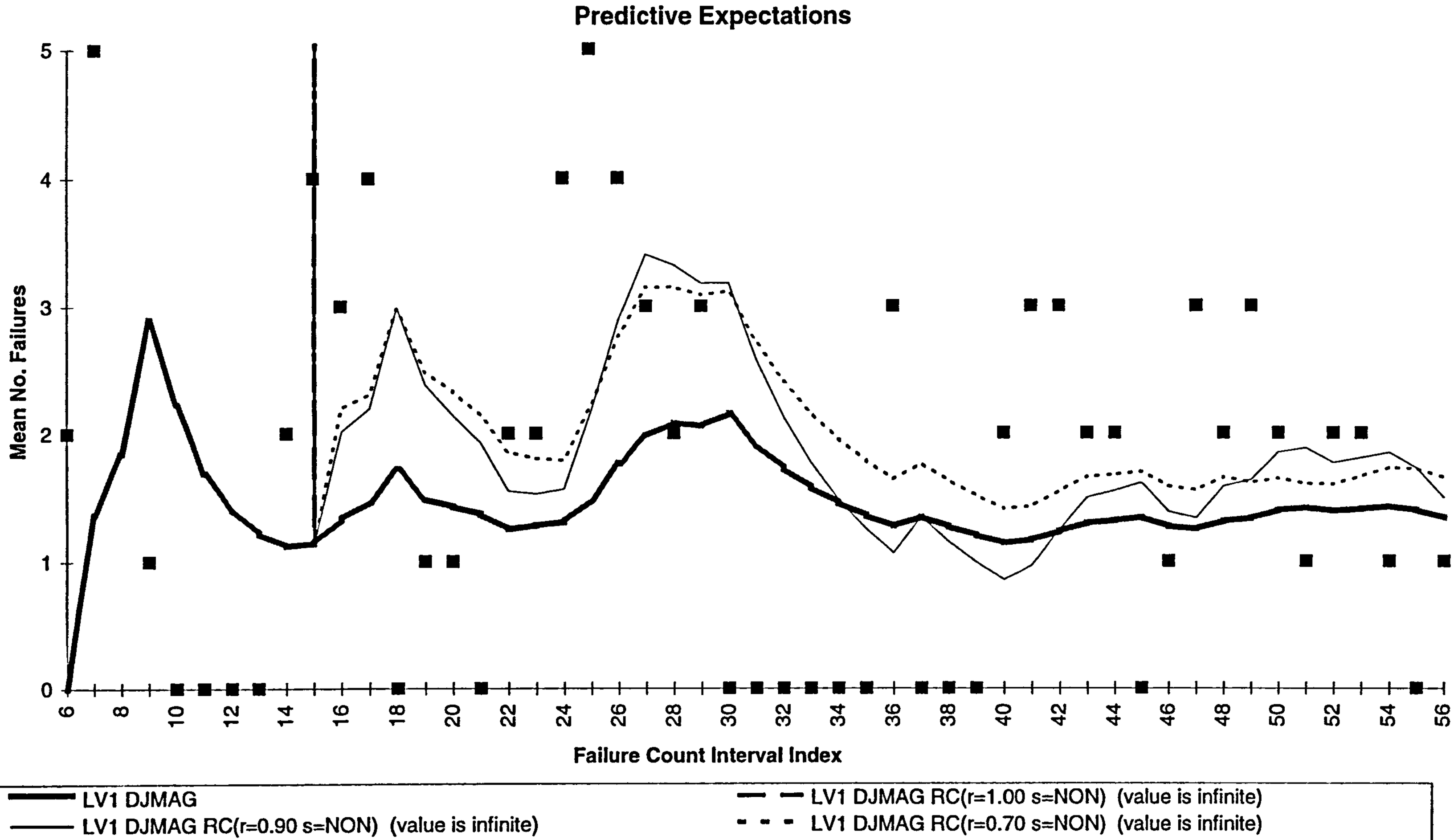

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
Failure Count Interval Index
LV1 DJMAG
LV1 DJMAG RC(r=0.90 s=NON) (value is infinite)
LV1 DJMAG RC(r=1.00 s=PYS.00.00)
Observed Counts (some off top of this scale)
LV1 DJMAG RC(r=1.00 s=NON) (value is infinite)
LV1 DJMAG RC(r=0.70 s=NON) (value is infinite)
LV1 DJMAG RC(r=0.90 s=PYS.00.00)

Figure 33a

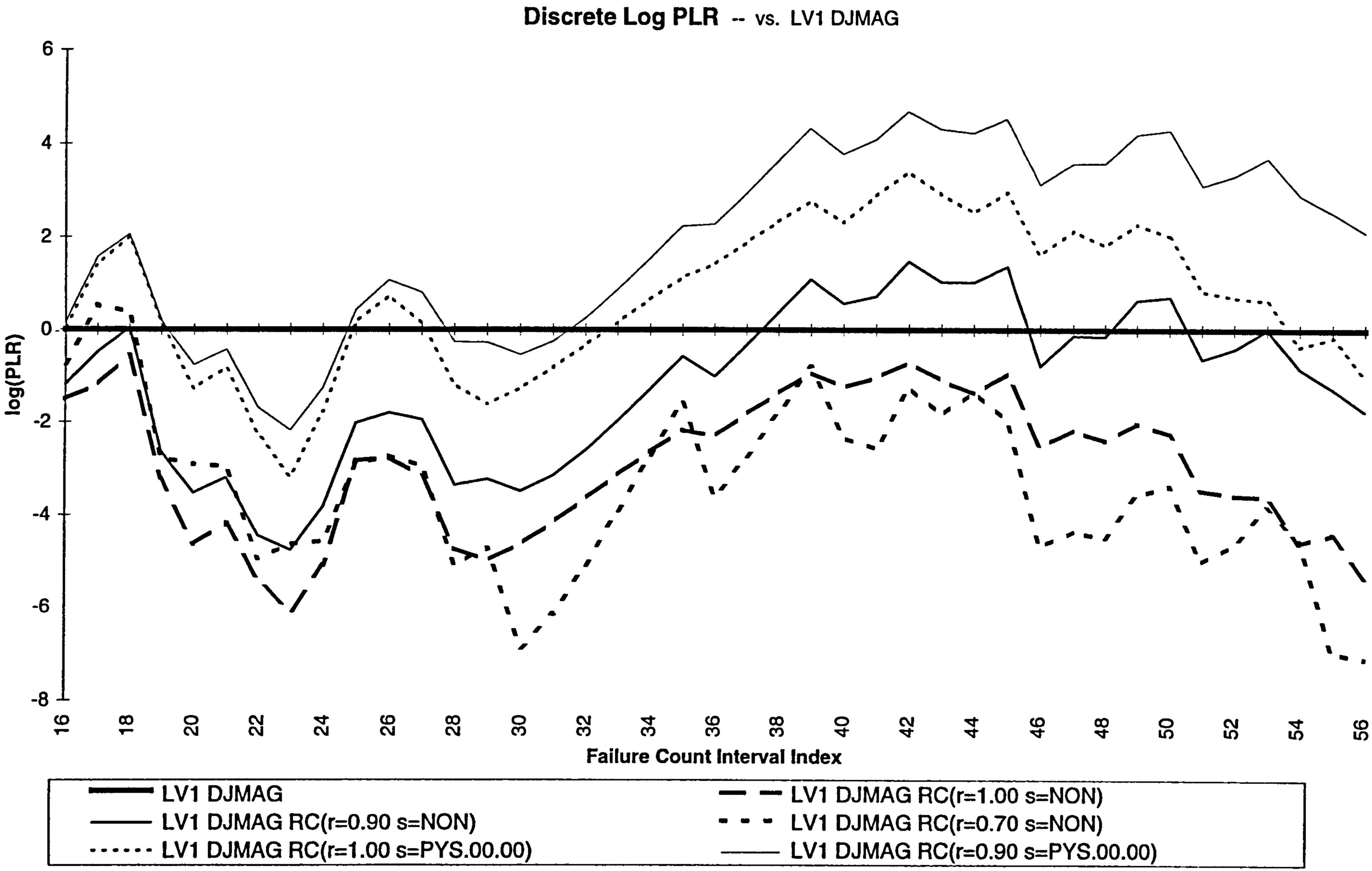

Discrete Log PLR -- vs. LV1 DJMAG
log(PLR)
6
4
2
0
-2
-4
-6
-8
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
Failure Count Interval Index
LV1 DJMAG
LV1 DJMAG RC(r=0.90 s=NON)
LV1 DJMAG RC(r=1.00 s=PYS.00.00)
LV1 DJMAG RC(r=1.00 s=NON)
LV1 DJMAG RC(r=0.70 s=NON)
LV1 DJMAG RC(r=0.90 s=PYS.00.00)

Figure 33b

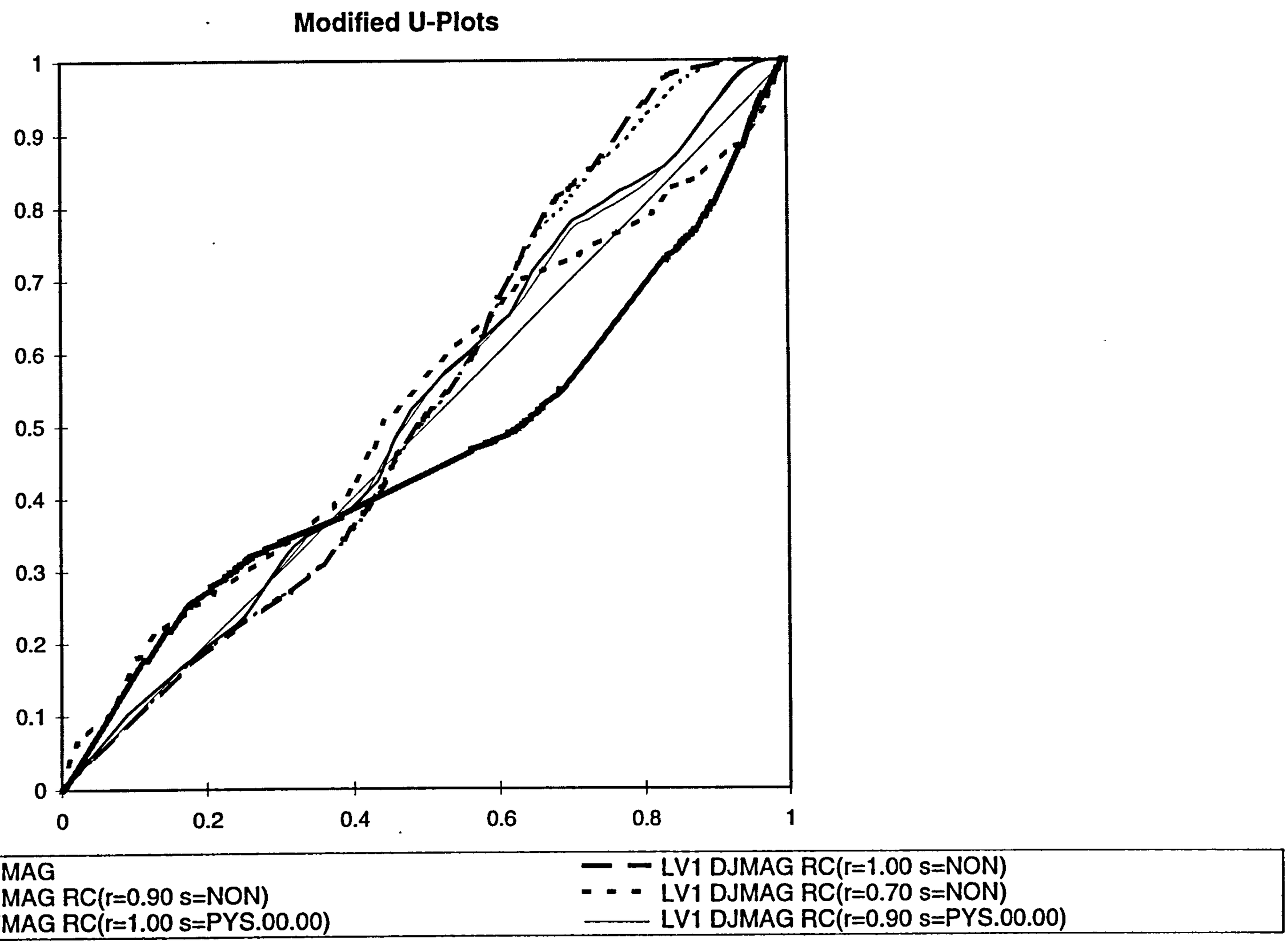

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
LV1 DJMAG
LV1 DJMAG RC(r=0.90 s=NON)
LV1 DJMAG RC(r=1.00 s=PYS.00.00)
LV1 DJMAG RC(r=1.00 s=NON)
LV1 DJMAG RC(r=0.70 s=NON)
LV1 DJMAG RC(r=0.90 s=PYS.00.00)

Figure 33c

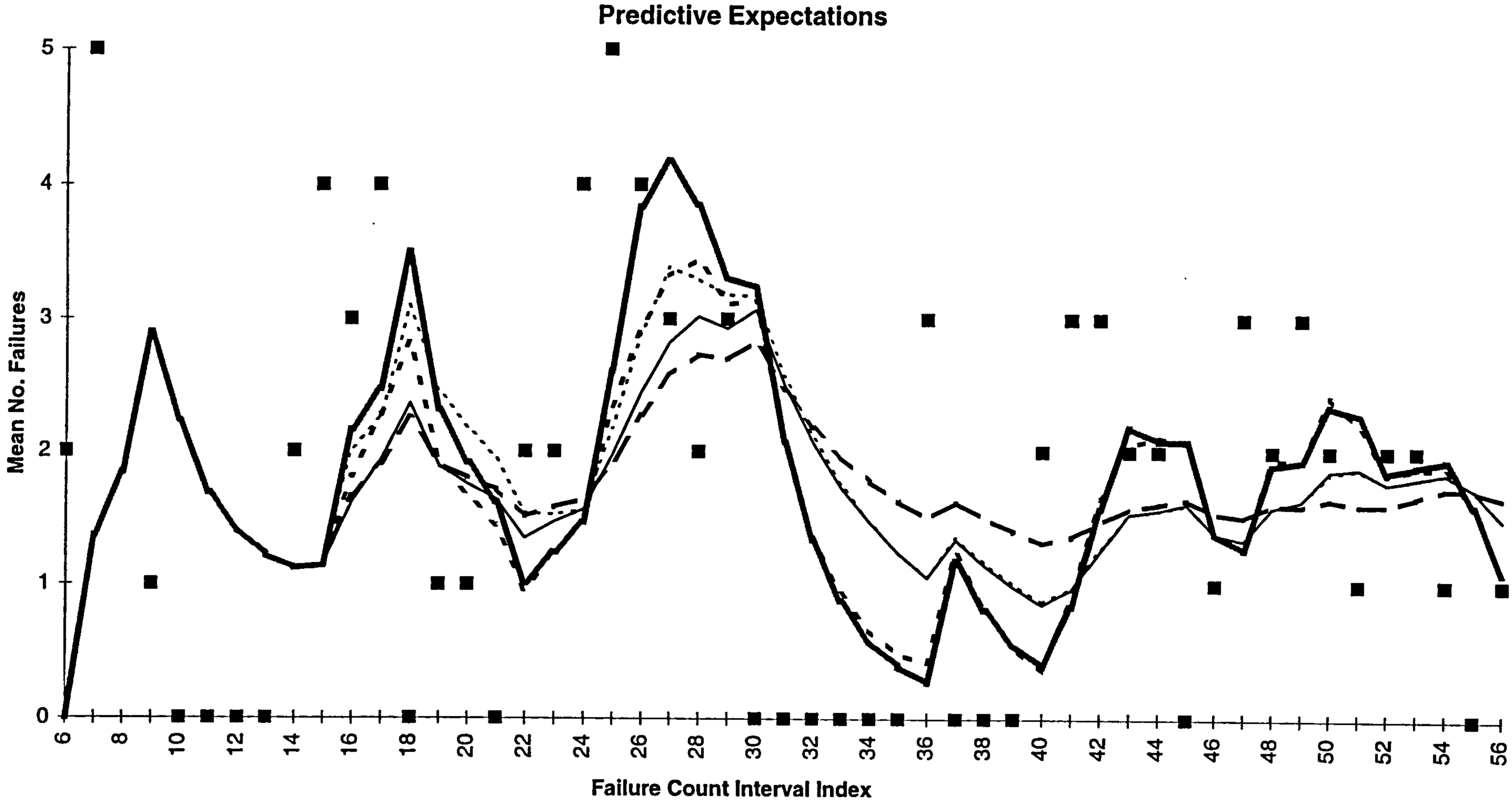

Predictive Expectations
Mean No. Failures
0
1
2
3
4
5
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
Failure Count Interval Index
LV1 DJMAG RC(r=0.70 s=PYS.00.00)
LV1 DJMAG RC(r=0.90 s=PYS.30.30)
LV1 DJMAG RC(r=0.90 s=PYS.30.00)
Observed Counts (some off top of this scale)
LV1 DJMAG RC(r=1.00 s=PYS.30.30)
LV1 DJMAG RC(r=0.70 s=PYS.30.30)
LV1 DJMAG RC(r=0.90 s=PYS.00.30)

Figure 34a

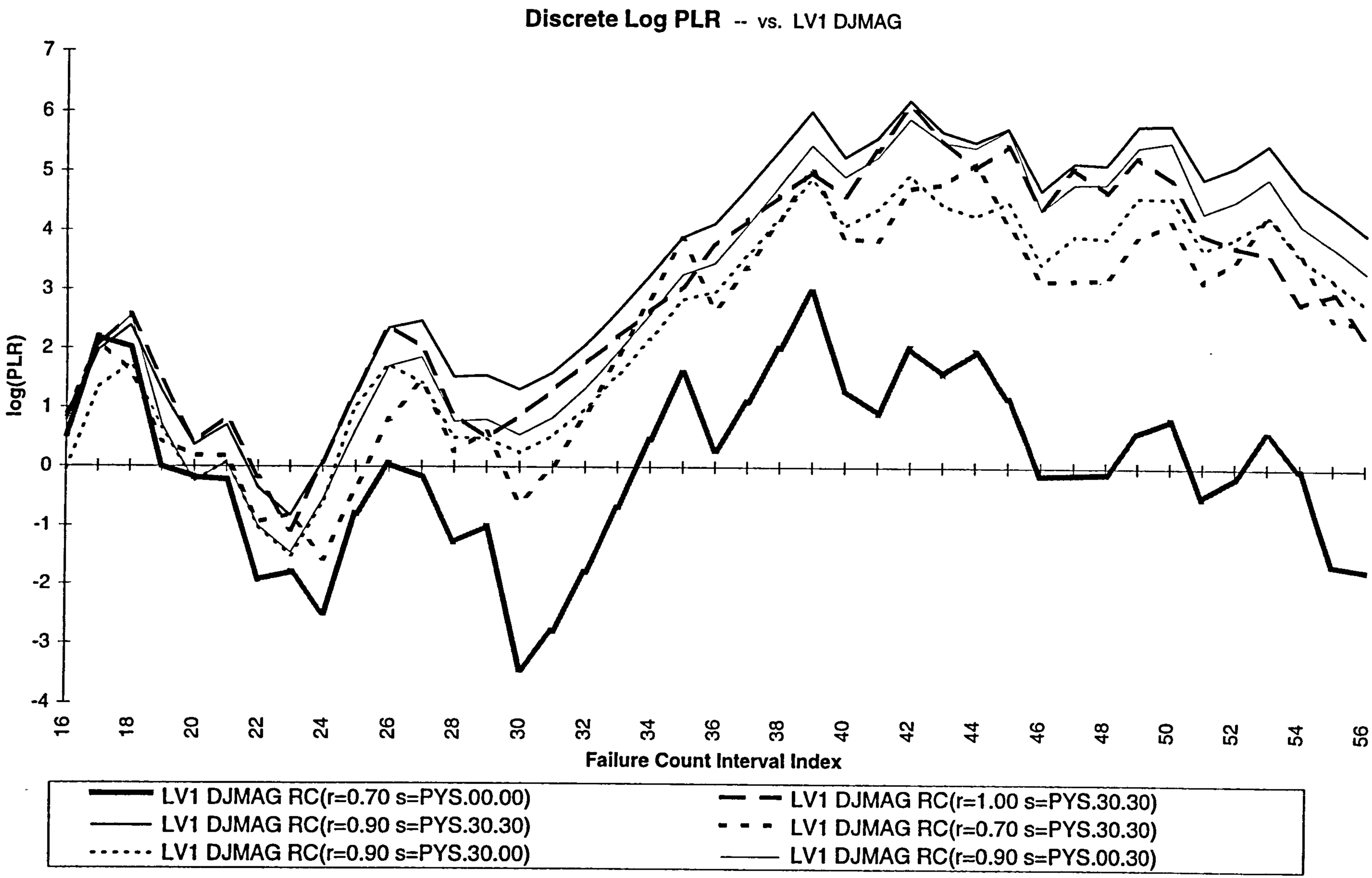

Discrete Log PLR -- vs. LV1 DJMAG
log(PLR)
7
6
5
4
3
2
1
0
-1
-2
-3
-4
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
Failure Count Interval Index
LV1 DJMAG RC(r=0.70 s=PYS.00.00)
LV1 DJMAG RC(r=0.90 s=PYS.30.30)
LV1 DJMAG RC(r=0.90 s=PYS.30.00)
LV1 DJMAG RC(r=1.00 s=PYS.30.30)
LV1 DJMAG RC(r=0.70 s=PYS.30.30)
LV1 DJMAG RC(r=0.90 s=PYS.00.30)

Figure 34b

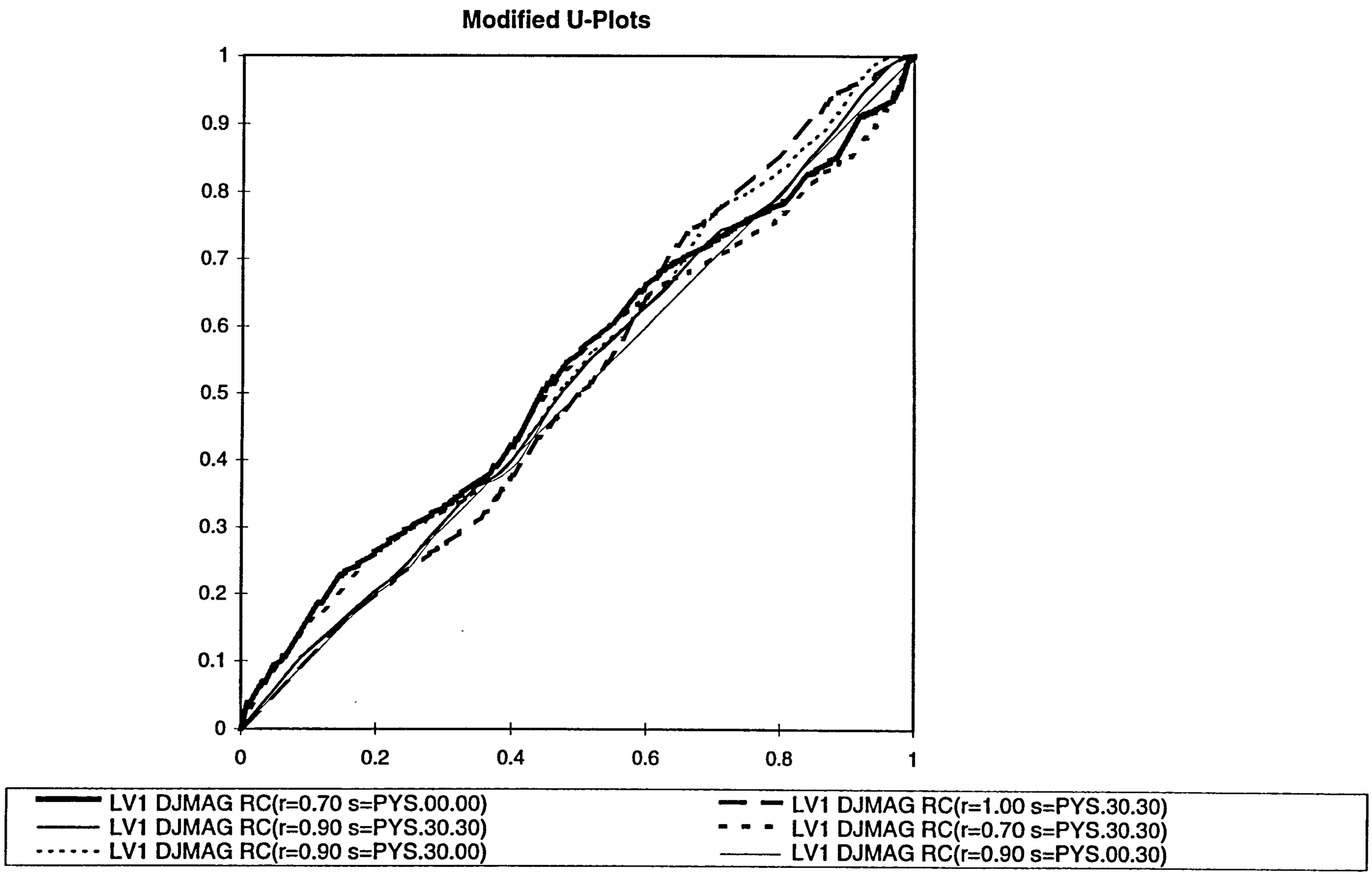

Modified U-Plots
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
LV1 DJMAG RC(r=0.70 s=PYS.00.00)
LV1 DJMAG RC(r=0.90 s=PYS.30.30)
LV1 DJMAG RC(r=0.90 s=PYS.30.00)
LV1 DJMAG RC(r=1.00 s=PYS.30.30)
LV1 DJMAG RC(r=0.70 s=PYS.30.30)
LV1 DJMAG RC(r=0.90 s=PYS.00.30)

Figure 34c

Appendix B

Mathematical Details

§B.1 contains details of the recalibration algorithm described in §3.6. More specifically this consists of some information about Chan & Snell's spline smoothing algorithm [10, Chapter 5], and the way that we have modified this here, so as to tighten the gradient constraints imposed. §B.2 concerns the calculation of 'recalibrated means', in the sense of means or expectations calculated with respect to one of the recalibrated predictive distributions $F_n^{*X}(x_n|x^{n-1})$ discussed in §3.6. (See in particular equation (34) on p65). We calculated some recalibrated expectations in order to produce the predictive expectation plots and the χ^2 predictive quality measures presented in Chapter 4, and Appendix A.

§§B.3–B.5 contain three detailed mathematical derivations required during the discussion of the *similar products* models (respectively on pages 141, 145, & 150 of §5.3).

B.1 The U-plot Spline-Fitting Algorithm used for Recalibration

B.1.1 Chan and Snell's Monotonic, Bi-cubic, Spline-Fitting Algorithm

In producing the recalibrators, the smoothing of the curve $u \mapsto S_{n-1}(u)$ defined by (29), (30), was done by re-using software written by Chan and Snell for the purpose of recalibrating inter-failure time predictions [11]. A brief description of the algorithm coded in this software now follows.

The smoothed fit is obtained from a set of data points $\{(x_j, y_j)'; j = 0, 1, \ldots, r\}$ by first parameterising in terms of the "normalised cumulative chord"

$$p_j = \frac{\sum_{i=1}^{j} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{\sum_{i=1}^{r} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}. \tag{107}$$

(Something will be said later about the selection and ordering of the points $(x_j, y_j)'$. The "prime" notation used here denotes "matrix transpose" since in describing the smoothing procedure, we choose to think in terms of column vectors representing points in the plane.) Cubic spline functions, $s_x(p)$ and $s_y(p)$, are then fitted separately by least-squares to the two data sets $\{(p_j, x_j)'\}$ and $\{(p_j, y_j)'\}$ to form a smooth parametric path in the plane, $c(p) = (s_x(p), s_y(p))'$, which approximates the data $(x_j, y_j)'$. The definition of the sum-of-squared-residuals objective function, the placement of knots, and the imposition of the constraints are all carried out independently for the two data sets, so that two quite separate constrained least-squares problems are solved in obtaining the two cubic spline functions s_x and s_y. The procedure involved is described for the $(p_j, x_j)'$ data set as follows—the method for the $(p_j, y_j)'$ data being exactly equivalent. The objective function is

$$\text{SSQR} = \sum_{i=1}^{r} (x_i - s_x(p_i))^2.$$

The spline-fitting algorithm is based on using a set of seven cubic B-splines as a basis for the space of cubic splines on the chosen knot set, (see [10] and [84]). The initial choice of knots satisfies $0 = \lambda_{-3} = \lambda_{-2} = \lambda_{-1} = \lambda_0 < \lambda_1 < \lambda_2 < \lambda_3 < \lambda_4 = \lambda_5 = \lambda_6 = \lambda_7 = 1$. Thus the spline fit is defined over the interval $0 \leq p \leq 1$ with three internal knots. The internal knots are chosen so that $\frac{1}{4}$ of the values p_j lie in each of the four subintervals created. The placement of four coincident knots at each end point is a numerical device to prevent the B-spline generating algorithm from automatically constraining s_x and its first two derivatives to be zero at these end-points. The insertion of these coincident knots causes the space of cubic splines spanned by the basis of B-splines to allow complete freedom for the values of s_x, s_x' and s_x'' at 0 and 1, although $s_x(0)$ and $s_x(1)$ *are* constrained during fitting by a different means as below. If the fit is particularly bad the algorithm will add extra knots until it improves. The constraints chosen can be represented as a finite set of simultaneous linear constraints on the seven B-spline coefficients, simplifying the computational problem. They are that $s_x(0) = 0$, $s_x(1) = 1$, and $s_x'(p)$, which is a *quadratic* spline on the same knot set $\{\lambda_i\}$, has no negative coefficients in its quadratic B-spline expansion. This last constraint implies that s_x is monotonic non-decreasing. (It is in fact a stronger constraint, easier to implement numerically. There are no problems with the fitted s_x being constant over an interval of positive length since it can be shown using basic properties of B-splines that such an interval would have to be of the form (λ_i, λ_j), which is never in practice the best fit to the data within the specified constraints. So

unless the data set is extremely unusual it can be assumed that the fitted s_x is *strictly* monotonic increasing.)

These constraints on s_x and s_y result in a fitted path $p \mapsto c(p)$ which defines a strictly monotonic smooth function $G : x \mapsto s_y(s_x^{-1}(x))$ with $G(0) = 0$, $G(1) = 1$. Then the "smoothed u-plot" is the graph of G on $[0, 1]$.

B.1.2 Method of Imposing Tighter Gradient Constraints

In order to make available the potential improvement mentioned at the end of §3.6, it is necessary to find a modification of this smoothing procedure which allows G to be further restricted so that its derivative satisfies a constraint of the form

$$0 < \epsilon \leq G' \leq \frac{1}{\delta} < \infty. \tag{108}$$

A straightforward extension of the previous method suggests solving for cubic spline functions s_x and s_y which minimise

$$\text{SSQR} = \sum_{i=1}^{r} (x_i - s_x(p_i))^2 + (y_i - s_y(p_i))^2. \tag{109}$$

subject to the constraints that s_x and s_y both assume the values 0 at 0, and 1 at 1. To ensure that the new derivative constraint (again strengthened for numerical reasons) is also satisfied, we could require that the two quadratic spline functions $-\epsilon s_x' + s_y'$ and $s_x' - \delta s_y'$ have no negative coefficients in their quadratic B-spline expansions. It would then follow that $s_y' \geq \epsilon s_x'$ and $s_x' \geq \delta s_y'$, which together are equivalent to the required constraints on G'.

It is perfectly feasible to efficiently solve such a problem by parameterising in terms of the coefficients in the cubic B-spline expansions of s_x and s_y, which is the approach used in Chan's and Snell's software for the previous problem. However this would require extensive modification to Chan's and Snell's software since what were previously two separate contrained least-squares problems in five dimensions would have to be replaced by a single problem in ten dimensions since the derivative constraints on s_x can no longer be separated from those on s_y. (Also both s_x and s_y would need to be defined using a common knot set, which currently does not always happen with Chan's and Snell's software.) For these reasons a shortcut is taken in obtaining the results presented in Chapter 4, allowing Chan's and Snell's existing software to be used with only minor modifications. The cost of this is that the norm in terms of which the objective function replacing (109) is defined is "distorted" by an amount depending on the size of ϵ and δ. Geometrically this shortcut works by "stretching" the data points (x_j, y_j) away from the 45°-line, fitting the smooth increasing curve as before, and then compressing the curve back towards the 45°-line, thereby tightening the bounds on

its derivative. The transformed data points are given by

$$\begin{pmatrix} X_j \\ Y_j \end{pmatrix} = A \begin{pmatrix} x_j \\ y_j \end{pmatrix}, \tag{110}$$

where

$$A = \begin{pmatrix} \frac{1}{1-\delta} & \frac{-\delta}{1-\delta} \\ \frac{-\epsilon}{1-\epsilon} & \frac{1}{1-\epsilon} \end{pmatrix}. \tag{111}$$

We restrict to $0 \leq \delta, \epsilon < 1$ so that the inverse transformation is

$$A^{-1} = \begin{pmatrix} 1-\vartheta & \vartheta \\ \eta & 1-\eta \end{pmatrix}$$

where $\delta = \frac{\vartheta}{1-\eta}$, $\epsilon = \frac{\eta}{1-\vartheta}$, and $\vartheta \geq 0$, $\eta \geq 0$, $\vartheta + \eta < 1$. Geometrically A moves a point along the direction $(\pm\vartheta, \mp\eta)'$ until its distance, in that direction, from the line "$y = x$" has been increased by a factor $\frac{1}{1-\vartheta-\eta}$. Cubic splines S_x and S_y are then fitted separately by the original algorithm to the data sets $\{(p_j, X_j)'\}$ and $\{(p_j, Y_j)'\}$, defining a strictly monotonic increasing function whose graph is the path $p \mapsto C(p) = (S_x(p), S_y(p))'$. Transforming back by

$$\begin{pmatrix} s_x(p) \\ s_y(p) \end{pmatrix} = A^{-1} \begin{pmatrix} S_x(p) \\ S_y(p) \end{pmatrix},$$

gives a function G defined as before by $G = s_y \circ s_x^{-1}$ satisfying the constraints (108).

The effect of this transformation on the original norm in (109) is as follows. In the "transformed space" of $(X_j, Y_j)'$ the objective function is

$$\text{SSQR} = \sum_{j=1}^{r} (X_j - S_x(p_j))^2 + (Y_j - S_y(p_j))^2$$

which, in terms of the original data $\{(x_j, y_j)'\}$, is a sum of the form

$$\begin{aligned} \text{SSQR} &= \sum_{j=1}^{r} (x_j - s_x(p_j), y_j - s_y(p_j)) A'A \begin{pmatrix} x_j - s_x(p_j) \\ y_j - s_y(p_j) \end{pmatrix} \\ &= \sum_{j=1}^{r} \left\| \begin{pmatrix} x_j \\ y_j \end{pmatrix} - \begin{pmatrix} s_x(p_j) \\ s_y(p_j) \end{pmatrix} \right\|_{A'A}^2, \end{aligned}$$

where this norm, $\| \cdot \|_{A'A}$, has elliptical contours, (each contour being the image under A^{-1} of a circle.) For the choices of ϵ and δ used in Chapter 4, the significance of the alteration in the norm is limited. This can be checked in any particular case since the direction of the major axis of a contour is

$$\begin{pmatrix} -\vartheta(1-\vartheta) + \eta(1-\eta) + \sqrt{(\vartheta^2+\eta^2)\{(1-\vartheta)^2 + (1-\eta)^2\}} \\ -2\vartheta\eta + \vartheta + \eta \end{pmatrix},$$

and the ratio of the major to the minor axis is

$$\sqrt{\frac{2\left\{1-\vartheta(1-\vartheta)-\eta(1-\eta)\right\}\left\{1-\vartheta(1-\vartheta)-\eta(1-\eta)+\sqrt{(\vartheta^2+\eta^2)\{(1-\vartheta)^2+(1-\eta)^2\}}\right\}}{(1-\vartheta-\eta)^2}} - 1.$$

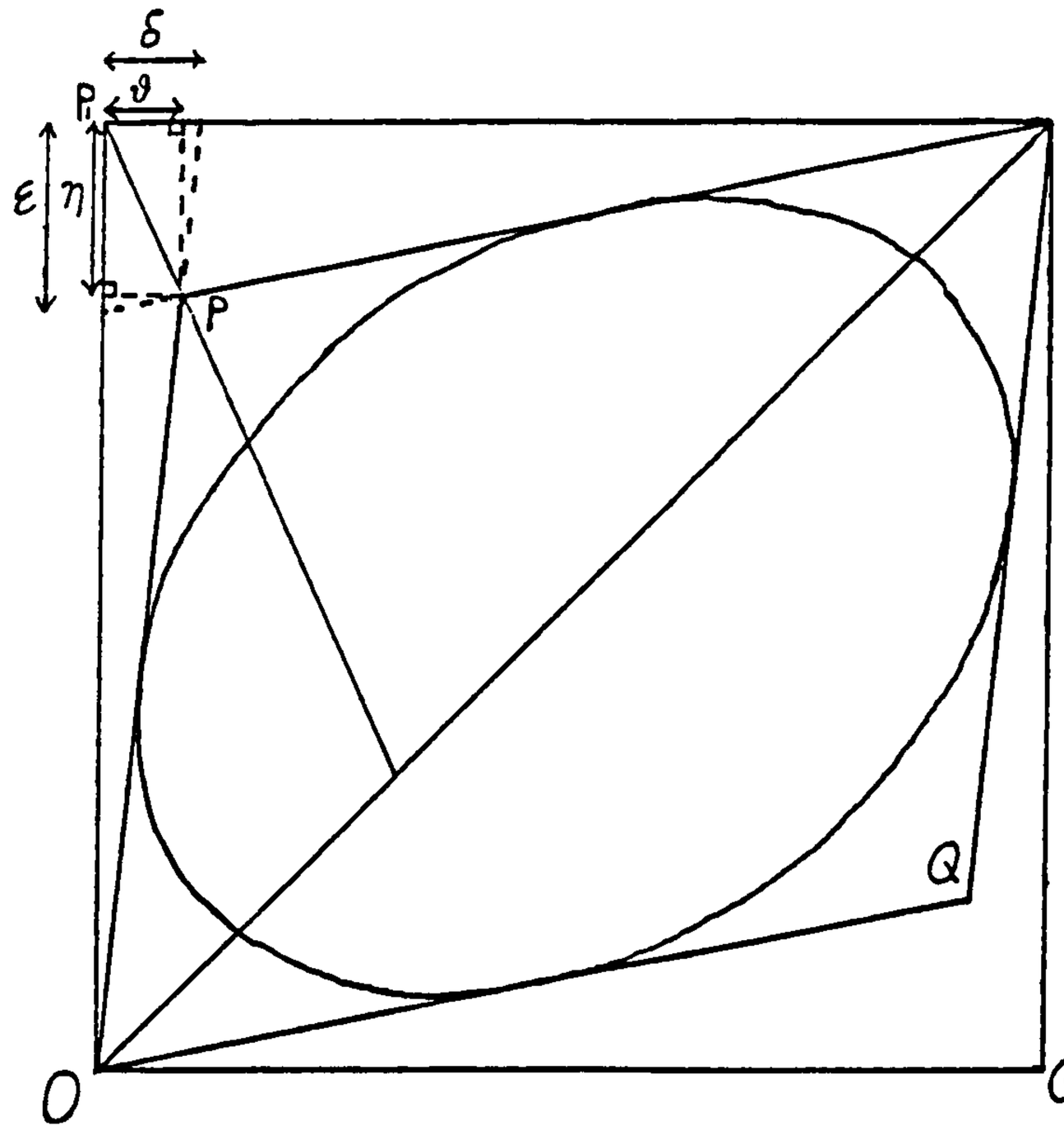


Figure 35: Geometrical description of linear transformation used in obtaining bounds on derivative of smoother

Figure 35 illustrates how the unit square and its inscribed circle are transformed under A^{-1}, and indicates geometrically the relationship between (δ, ϵ) and (ϑ, η). In this figure, the square $P_1 Q_1$ has sides of length 1. A maps P to P_1, Q to Q_1, and the ellipse shown onto the inscribed circle of the square.

B.1.3 Selection of Data Points for Spline-Fitting Algorithm

In obtaining the smoothed recalibrators, the data points $\{(x_j, y_j)'; j = 1, \ldots, r\}$ were obtained by dividing $[0, 1]$ into 50 equal intervals with mid-points m_j (indexed in increasing order) and either setting $x_j = \mathrm{m}_j$ or $y_j = \mathrm{m}_j$, and then reading off the other member of each pair from the unsmoothed function S_n defined in §§3.5 and 3.6, (with either constant or exponential weights, w_i, as discussed in §3.6). It appears that for the failure data and raw PFSs chosen, of these two possibilities for

obtaining the (x_j, y_j), equally spaced x_j give better PL performance of the recalibrated predictor than equally spaced y_j—the results presented in §4.2 were obtained with equally spaced x_j. In equation (107) $x_0 = y_0 = 0$ were used.

B.2 Moments and Expectations from a Recalibrated Predictive Distribution

The method of recalibration proposed in this thesis involves defining a recalibrated c.d.f. $F_n^{*X}(x_n|x^{n-1})$ by composing the corresponding raw (i.e. unrecalibrated) c.d.f. $F_n^X(x_n|x^{n-1})$, obtained at a given stage of one-step-ahead prediction, with a Modified U-Plot function $G_n^{S^*}(\cdot|u^{n-1}) = S_{n-1}$ based on earlier u-residuals of the raw PFS. Thus, the numerical calculation of a recalibrated c.d.f. value at a particular numerical argument requires only that we are able to evaluate individual numerical values of: (i) the raw c.f.d.; and (ii) the U-Plot function S_{n-1}. Likewise, the calculation of percentiles of the recalibrated predictive distribution involves only the calculation of values of the inverses of the same two functions at specified numerical arguments. We assume that these problems are easily solved for the raw c.d.f.[1]. It is clearly not difficult[2] to calculate values and inverse values of S_{n-1} implemented as a sample c.d.f. of past u_is. Neither the incorporation of weights w_i in the definition of S_{n-1}, as in equation (29), nor the replacement of Heaviside functions by the uniform c.d.f.s of (30), then pose any serious problems either. We still are limited to a non-decreasing polygonal S_{n-1} function, possibly with discontinuities at which it is defined to be right-continuous. When S_{n-1} is smoothed with cubic B-splines it merely becomes necessary to know how to find a root of a cubic polynomial[3], which is also straightforward. So there appear to be no significant problems with the numerical calculation of values of recalibrated c.d.f.s or percentiles.

The problem of the calculation of *moments* of recalibrated distributions is not quite so easy. We have proposed our recalibration procedure as quite generally applicable to arbitrary *discrete*, *continuous* or *mixed* one-step-ahead predictive distributions for a sequence of scalar quantities. In the numerical work here with failure-count prediction, however, only the first mentioned of these three cases has actually been needed. We will now briefly discuss the problem of numerical calculation of recalibrated expectations[4] in the general mixed case, and, in more detail, as we have used it in

---

[1]—always Poisson or Binomial for the raw failure-count PFSs explicitly discussed in this thesis.

[2]Of course, not all percentiles are necessarily defined for a distribution whose c.d.f. contains discontinuities; and it can be argued that some percentiles are not *uniquely* defined when the c.d.f. is constant on subranges of $\mathbb{R}$. But neither of these restrictions has any special connection with the use of recalibrated distributions.

[3]This continues to be the case when gradient-contraints ϵ and $\frac{1}{\delta}$ are imposed, because linear combinations of cubic polynomials are cubic polynomials.

[4]i.e., simply ordinary probabilistic expectations but starting from a probability distribution which contains a

Chapter 4 for discrete failure count prediction.

We can simplify the situation by forgetting about the predictive nature of our distribution, and the fact that it involves residuals from past terms in a sequence : The question addressed here is essentially simply that of how to calculate expectations with respect to a distribution, say F^*, obtained as $S \circ F$ where F is a given c.d.f., and S is a non-decreasing function satisfying $S(0) = 0$ and $S(1) = 1$. If our distribution is that of a non-negative RV X, then we can generally apply the formula

$$\mathrm{E}[X] = \int_0^\infty R(x)\,dx \qquad \text{where } R(x) = 1 - F(x) = \mathrm{P}[X{>}x].$$

The restriction to non-negative RVs is not an important limitation here since in other cases we can simply calculate the difference of expectations of the two non-negative RVs $XI_{[0,\infty)}(X)$ and $-XI_{(-\infty,0)}(X)$ (repectively $\max(X,0)$ and $\max(-X,0)$). Continuing to use the $*$ superscript to refer to recalibrated versions, it is easy to see that

$$R^* = l \circ R, \qquad \text{where } l(v) = 1 - S(1-v),\ 0 \le v \le 1.$$

The graph of l is just the graph of S rotated 180° about the point $(\frac{1}{2}, \frac{1}{2})$[5]. Thus the recalibrated expectation is

$$\mathrm{E}^*[X] = \int_0^\infty l(R(x))\,dx\,. \tag{112}$$

This integral could be evaluated using a numerical integration algorithm. However, the function l may be of a form such that we can locally approximate $l \circ R$ by a linear function of R. If this is the case, and if the raw distribution is one which is analytically tractable (as for us will often be the case, for example, if it is obtained by ML plug-in from a known model) then it may be easier, either over the entire range of integration or for just selected sub-intervals of this range, to approximate recalibrated expectations in terms of known raw expectations. Clearly the details of the best method of evaluating (112) for an *arbitrary* mixed-distribution of X will depend on all sorts of considerations such as the location and number of the points of concentrated probability mass in this distribution, and indeed, in the general case, the potential presence of accumulation points of this set of points. However, for the relatively simple discrete failure-count case, we found that we could divide the interval of integration into two sub-intervals, one requiring direct calculation of a sum of a small finite number of terms, and the other approximable in terms of the corresponding 'raw expectation' (the expectation of the distribution prior to the recalibration step). For any non-negative RV we can fix a number, a say, and express the expectation as the sum of a 'mean restricted to $[0,a]$' and

---

recalibration step within its definition.

[5] we will use independent variable v to denote a quantity corresponding to $1-u$, i.e. measuring a horizontal distance from the RHS of a u-plot.

a 'mean excess above a'.

$$\mathbf{E}[X] = \mathbf{E}[\min\{X, a\}] + \mathbf{E}[\max\{X - a, 0\}] \ .$$

Doing so exactly corresponds to dividing the interval of integration in (112) to give

$$\mathbf{E}^*[X] = \int_0^a l(R(x))\,dx + \int_a^\infty l(R(x))\,dx$$

In fact, for our purpose with failure-count predictions we can use an integer, say $a = m$, and define L_m, U_m by

$$L_m = \inf_{0<v\leq R(m)} \frac{l(v)}{v}, \qquad U_m = \sup_{0<v\leq R(m)} \frac{l(v)}{v}$$

so that for bounds on the recalibrated 'mean excess above m' in terms of the corresponding raw quantity we have

$$L_m \int_m^\infty R(x)\,dx \leq \int_m^\infty l(R(x))\,dx \leq U_m \int_m^\infty R(x)\,dx \tag{113}$$

where the raw term $\int_m^\infty R(x)\,dx$ is generally known analytically. For example if the raw distribution of X is Poisson with parameter λ, then we have

$$\int_m^\infty R(x)\,dx = \lambda - m + e^{-\lambda} \sum_{i=0}^{m-1} (m-i)\frac{\lambda^i}{i!}$$

As m is increased here, the sum

$$\int_0^m l(R(x))\,dx = \sum_{i=0}^{m-1} l(R(i))$$

has increasingly more terms requiring to be calculated numerically; but the bounds in (113) become, at the same time, tighter due to the accompanying reduction in the interval $[0, R(m)]$. For a polygonal recalibrator function, arising from the unsmoothed modified u-plot S, the bounds L_m and U_m can be obtained by checking only the values at vertices. For a spline-smoothed monotonic[6] S, extrema of the function $\frac{l(v)}{v}$ when reparameterized in terms of p (or better $1-p$ for a slight simplification) must be sought. This means checking the end point $p = 1$, and any internal knot locations at values of p that correspond to $v \leq R(m)$, and the stationary points of the ratios of cubics, if any of these fall within this search interval. (In practice, it is in the majority of cases easy to take m large enough that all B-spline internal knots correspond to $v > R(m)$, in which case – because $s_x(1) = s_y(1) = 1$ – we may then cancel a common factor $1 - p$ from the cubic ratio corresponding to the right-most polynomial component of each of the two splines and obtain instead a ratio of quadratics only, whose extrema must be obtained.) In either case, whether cubic or quadratic

[6] of the kind discussed in §B.1, denoted there by the function $G : x \mapsto s_y(s_x^{-1}(x))$ with $s_y(p)$, and $s_x(p)$ being the two cubic splines defined in terms of the normalised cumulative chord p of equation (107)

rational function, analytic differentiation easily yields formulas for the internal extrema, if any, lying within the search interval $s_x^{-1}(F(m)) \leq p \leq 1$. In the case of the more tightly gradient-constrained smoother (positive ϵ and/or δ on p227) the locations (in terms of the parameter p) of any such internal extrema are, conveniently, unchanged from the $\epsilon = \delta = 0$ case: Geometrically (see Figure 35), the linear transformation A of equation (111) maps straight lines through (1,1) onto other straight lines through (1,1), preserving the order of their gradients. Since $\frac{l(v)}{v}$ is simply the slope of such a line passing through the point $(1-v, S(1-v))$ on the smoothed u-plot, it is easy to see that the p-location of the extrema is unchanged by the act of executing the transformation A^{-1} (after using Chan's and Snell's least-squares spline fitting procedure, as indicated earlier in this appendix, on the A-transformed unsmoothed u-plot data-points).

A generally similar approach may work equally well for recalibrated higher moments and recalibrated expectations of many other quantities. In the general mixed case we might decompose the expectation into a sum of expectations restricted to (possibly more than two) sub-intervals that form a partition of the positive real line and use equations such as, for example

$$\begin{aligned} \mathbf{E}^*\left[g_-(X)I_{[a,b)}(X)\right] &= -\int_{[a,b)} g_-(x)\, d(l \circ R)(x) \\ &= \int_{[a,b)} l(R(x))\, dg(x) + l(R(a))g(a) - l(R_-(b))g_-(b) \end{aligned}$$

where l is a 180°-rotated u-plot of some kind (perhaps smoothed, and certainly a monotonic function on the unit interval with $l(0) = 0$ and $l(1) = 1$) and g must be of bounded variation on $[a, b)$. (See the integration-by-parts formula (25) on p54.) Here, the analogous unrecalibrated expectation would look the same with all the l's ommitted, so that bounds or approximations for l on each interval $[R(b-), R(a)]$ might again be used to bound or approximate the recalibrated expectation of $g_-(X)$ in terms of the analogous raw expectation.

B.3 Proof that the Best Attainable Improvement $\mathscr{R}$ of the Odds that P_k=0 which is Obtained by Incorporation of Previous Demand-Sequence Data is, for Fixed Prior Θ-Mean, an Increasing Function of our Prior Variance of Θ

This proof concerns the influence of the spread of our prior distribution for Θ on the improvement in $\mathbf{P}[P_k{=}0]$ which results from taking account of fixed observations of non-failure of previous sequences. See p141 for a more complete statement of the context. We will show this result specifically for the two-point P-distribution of §5.3.5.1 where, under our further restriction of §5.3.5.1 to a Beta assumption for Prior_θ, the result translates into the property that $\mathscr{R}$ of equation (94) is a decreasing function of the Beta parameter b, for $o = a/b$ fixed. So we work in terms of the reparameterised Beta Prior_θ

$$\text{Prior}_\theta(\theta) = \frac{\theta^{ob-1}(1-\theta)^{b-1}}{\beta(ob,b)}, \qquad \frac{\theta_1}{1-\theta_1} = o \quad \text{(see p137)}. \tag{114}$$

Questions remain about the precise form and extent of any generalisations of the monotonicity result beyond this case. On expanding the polynomial $l(\theta) = \sum_{i=0}^{k-1} c_i\theta^i$, the left-hand side of (94) becomes

$$\mathscr{R} = \frac{\sum_{i=0}^{k-1} c_i\beta(ob+i+1,b)}{o\sum_{i=0}^{k-1} c_i\beta(ob+i,b+1)}. \tag{115}$$

Since we are assuming that the $k{-}1$ previous products have all performed perfectly in their respective environments over the numbers of trials observed for each, $l(\theta)$ remains defined as on p137 so the coefficients $\langle c_i\rangle$ are all non-negative. Now, holding the odds parameter o constant and differentiating $\mathscr{R}$ with respect to b, we can verify a non-increasing function $\mathscr{R}(b)$ for all $b > 0$ as follows.

$$\frac{\partial\mathscr{R}}{\partial b} =$$
$$\left\{\sum_{i=0}^{k-1} c_i\left[o\psi(ob{+}i{+}1) + \psi(b) - (o{+}1)\psi(ob{+}b{+}i{+}1)\right]\beta(ob{+}i{+}1,b)\sum_{j=0}^{k-1} c_j\beta(ob{+}j,b{+}1)\right.$$
$$\left. -\sum_{j=0}^{k-1} c_j\beta(ob{+}j{+}1,b)\sum_{i=0}^{k-1} c_i\left[o\psi(ob{+}i) + \psi(b{+}1) - (o{+}1)\psi(ob{+}b{+}i{+}1)\right]\beta(ob{+}i,b{+}1)\right\} \tag{116}$$
$$\Big/\, o\left\{\sum_{i=0}^{k-1} c_i\beta(ob{+}i,b{+}1)\right\}^2$$

where ψ is the digamma function (put $r = 0$ in (134)) and where we have used the relations

$$\frac{\partial\beta}{\partial a}(a,b) = \beta(a,b)[\psi(a) - \psi(a+b)], \qquad \frac{\partial\beta}{\partial b}(a,b) = \beta(a,b)[\psi(b) - \psi(a+b)]. \tag{117}$$

The two double sums of the numerator of (116) can be expanded out and their corresponding terms subtracted to express the numerator as a single double sum of the form

$$\begin{aligned} \text{Numerator} &= \sum_{i=0}^{k-1}\sum_{j=0}^{k-1} t_{ij}c_ic_j \\ &= \sum_{i=0}^{k-1} t_{ii}c_i^2 + \sum_{i=1}^{k-1}\sum_{j<i}(t_{ij}+t_{ji})c_ic_j\,. \end{aligned} \tag{118}$$

We proceed to show that neither of the summands in the second form (118) can be greater than 0. In examining the terms of (118) it simplifies matters to first extract from t_{ij} a positive factor

$$s_{ij} = \frac{b}{(ob+b+i)(ob+b+j)}\beta(ob+i,b)\beta(ob+j,b) = \frac{\Gamma(ob+i)\Gamma(ob+j)\Gamma(b+1)\Gamma(b)}{\Gamma(ob+b+i+1)\Gamma(ob+b+j+1)} \tag{119}$$

which is symmetric in i,j. It can then be shown by selecting the relevant terms from the numerator of (116) and using the recurrence formula $\psi(z+1)=\psi(z)+1/z$ that $t_{ii}/s_{ii} = -i/b \le 0$. To do the same for the off-diagonal sums of pairs of terms from (118) is slightly more cumbersome. Selecting the relevant terms from (116) and simplifying in a similar way, we are left with (assuming without loss of generality that $i>j\ge 0$)

$$\frac{t_{ij}+t_{ji}}{s_{ij}} = (i-j)\Big\{o\big[\psi(ob+i)-\psi(ob+j)\big]-(o+1)\big[\psi(ob+b+i+1)-\psi(ob+b+j+1)\big]\Big\}-\frac{i+j}{b} \tag{120}$$

The problem here is the curly-bracketed term: ψ is monotonic increasing and the fact that $o+1>o$ suggests that this term might be negative, finishing our task rather easily. But perhaps the ratio $(o+1)/o$ *is insufficiently* large to compensate for the fact that the function ψ is *concave*. To verify that the whole expression (120) cannot be positive we manipulate it as follows, beginning by expanding the two ψ-differences

$$\begin{aligned} \frac{t_{ij}+t_{ji}}{s_{ij}} &= (i-j)\sum_{h=j}^{i-1}\left(\frac{o}{ob+h}-\frac{o+1}{(o+1)b+h+1}\right)-\frac{i+j}{b} \\ &= (i-j)\sum_{h=j}^{i}\left(\frac{o}{ob+h}-\frac{o+1}{(o+1)b+h}\right)+(i-j)\left(\frac{o+1}{(o+1)b+j}-\frac{o}{ob+i}\right)-\frac{i+j}{b} \\ &= (i-j)\sum_{h=j}^{i}\frac{-h}{(ob+h)\big((o+1)b+h\big)}-\frac{(i+j)o(o+1)b^2+ij(i+j+2b+4ob)}{b(ob+i)\big((o+1)b+j\big)} \le 0 \end{aligned}$$

We can therefore finally conclude that $\frac{\partial\mathscr{R}}{\partial b}\le 0$, making $\mathscr{R}$ a monotonic non-increasing function of $b>0$ for $o=a/b$ fixed. Examining the reasoning above, we see that $\mathscr{R}$ will almost always be a *strictly* decreasing function of b, the exceptional cases occurring only when we consider a few special or limiting conditions on the number $k-1$ of previous sequences, the Beta parameters o,b, and the possibility of zero values for the coefficients (c_i) in the expansion of the polynomial $l(\theta)$. For $\mathscr{R}(b)$ to be constant over some b-interval would require *all* terms in the sum (118) to be zero inside that interval.

B.4 Taylor Expansion of Numerator of Improvement $\mathscr{R}$ in Odds of $\mathcal{A}_k$-Perfection that Results From Observation of $\langle \mathcal{A}_1, \ldots, \mathcal{A}_{k-1} \rangle$

We wish to expand the two middle terms of the numerator of (104) on p144 as a Taylor series in powers of $\langle \nu_1, \nu_2, \nu_3 \rangle$, at the point $\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$. Our expansion is required to hold only within the plane $\nu_1+\nu_2+\nu_3 = 1$. It is clear from the development on p145 that what it remains to do, is to show that the Taylor expansion for the term

$$t = \sqrt{\frac{a+2}{b+1}} e^{\frac{z}{6}} \sum_{i=1}^{3} e^{(\frac{1}{3}-\nu_i)Z} + \sqrt{\frac{b+1}{a+2}} e^{-\frac{z}{6}} \sum_{i=1}^{3} e^{(\nu_i-\frac{1}{3})Z} \tag{121}$$

is given by the series

$$s = 6\cosh(\alpha) + \cosh(\alpha) Z^2 r^2 + \frac{Z^3}{3} \sum_{i=1}^{3} (\nu_i - \frac{1}{3})^3 \sinh\left[\alpha + u(\nu_i - \frac{1}{3})Z\right] \tag{122}$$

By (105) we can write t as

$$\begin{aligned} t &= e^{-\alpha} \sum_{i=1}^{3} e^{(\frac{1}{3}-\nu_i)Z} + e^{\alpha} \sum_{i=1}^{3} e^{(\nu_i-\frac{1}{3})Z} && (123) \\ &= 2\sum_{i=1}^{3} \cosh\left[\alpha + (\nu_i - \tfrac{1}{3})Z\right] && (124) \\ &= F(Z), \qquad \text{say.} && (125) \end{aligned}$$

We can now use a Taylor expansion for the function F

$$F(Z) = F(0) + F'(0)Z + F''(0)\frac{Z^2}{2} + F^{(3)}(uZ)\frac{Z^3}{6}, \qquad \text{where } 0<u<1$$

with

$$F^{(n)}(Z) = \begin{cases} 2\sum_{i=1}^{3} (\nu_i - \frac{1}{3})^n \cosh[\alpha + (\nu_i - \frac{1}{3})Z], & n = 0, 2, 4, \ldots \\ 2\sum_{i=1}^{3} (\nu_i - \frac{1}{3})^n \sinh[\alpha + (\nu_i - \frac{1}{3})Z], & n = 1, 3, 5, \ldots \end{cases}$$

to deduce the result that $t = s$ for some $0<u<1$ (with u depending on a, b, Z and $\langle \nu_i \rangle$). The first order term is zero because $\sum_{i=1}^{3} (\nu_i - \frac{1}{3}) = \left[\sum_{i=1}^{3} \nu_i\right] - 1 = 0$

B.5 Numerical Approximation to Very High Order Non-Central Moments of the Beta Distribution

We require for the purpose of numerical integration in §5.3.5.2 to have an efficient algorithm for calculating the expectation of $(1 - P)^n$ for large n when P is distributed with a Beta distribution

with parameters a, b. Thus we require an algorithm to calculate[7]

$$\begin{aligned}\mu_{0,n}(a,b) &= \frac{\beta(a,b+n)}{\beta(a,b)} \\ &= \frac{\Gamma(b+n)\,\Gamma(a+b)}{\Gamma(b)\,\Gamma(a+b+n)} \\ &= \frac{b(b+1)\ldots(b+n-1)}{(a+b)(a+b+1)\ldots(a+b+n-1)}\,.\end{aligned}$$

Problems with overflow, long computation times, and loss of precision due to subtraction of very similar numbers were experienced when attempting to compute using standard beta, gamma and log-gamma library functions in the obvious ways directly from the forms above. To avoid these problems some bounds are obtained below by directly working with the specific function $\mu_{0,n}(a,b)$.

Firstly, we note that

$$\log\left(\mu_{0,n}(a,b)\right) = \sum_{i=0}^{n-1} \log\left(1 - \frac{a}{a+b+i}\right)$$

so we can apply the 'integral test' approximation to sums of any strictly decreasing function f

$$\int_0^n f(t)\,dt < \sum_{i=0}^{n-1} f(i) < f(0) - f(n) + \int_0^n f(t)\,dt\,,$$

where $f(t) = -\log\left(1 - \frac{a}{a+b+t}\right)$, to give the interesting bounds

$$\mathcal{F}[t \mapsto (t-1)\log(t)] < \log\left(\mu_{0,n}(a,b)\right) < \mathcal{F}[t \mapsto t\log(t)] \tag{126}$$

where $\mathcal{F}$ (or, strictly, $\mathcal{F}_{a,b,n}$) is the linear functional given by

$$\mathcal{F}[g] = -g(a+b+n) + g(b+n) + g(a+b) - g(b)\,, \quad \text{for functions } g\,,$$

i.e., intuitively, $\mathcal{F}$ applies to its scalar function argument a difference operator the 'spacing' of whose differences is specified by a and n and the 'location' of application of which is specified by b. Note that, when $\mathcal{F}$'s argument g can be differentiated twice, the identity

$$\mathcal{F}[g] = -\int_0^a \int_0^n g''(t_1 + t_2 + b)\,dt_2\,dt_1 \tag{127}$$

will sometimes be used in what follows to demonstrate monotonicity of expressions which involve $\mathcal{F}$. Since we are interested in cases where n (and sometimes b also) are large compared to a, there are likely to be subtraction problems with numerical accuracy in calculating the bounds in (126) and some other bounds and approximations which also turn out to be defined by the application of $\mathcal{F}$ to

[7] Although, for our purposes we are only interested in integer n, we note in passing that moments of non-integer order are perfectly well defined and that for the Beta distribution we have the curious symmetry $\mu_{0,n}(a,b) = \mu_{0,a}(n,b)$, apparent from the Gamma-function representation here.

some function. These can be solved by rearrangement and perhaps also Taylor series approximations. E.g. for the upper bound, if n is much larger than a we can use

$$-(a+b+n)\log(a+b+n)+(b+n)\log(b+n)+(a+b)\log(a+b)-b\log(b)=$$
$$-a\,log(a+b+n)-(b+n)\log\left(\frac{a+b+n}{b+n}\right)+(a+b)\log(a+b)-b\log(b)$$

—unless b is also much larger than a, in which case

$$-(a+b+n)\log(a+b+n)+(b+n)\log(b+n)+(a+b)\log(a+b)-b\log(b)=$$
$$-a\log\left(\frac{a+b+n}{a+b}\right)$$
$$+b\left[-\frac{1}{2}\left(\frac{a}{b}\right)^2+\frac{1}{3}\left(\frac{a}{b}\right)^3-\ldots\right]-(b+n)\left[-\frac{1}{2}\left(\frac{a}{b+n}\right)^2+\frac{1}{3}\left(\frac{a}{b+n}\right)^3-\ldots\right]$$

should produce an accurate answer. Once these rather minor subtraction problems have been tackled, the resulting bounds[8]

$$\frac{(b+n)^{(b+n-1)}(a+b)^{(a+b-1)}}{b^{(b-1)}(a+b+n)^{(a+b+n-1)}}<\mu_{0,n}(a,b)<\frac{(b+n)^{(b+n)}(a+b)^{(a+b)}}{b^b(a+b+n)^{(a+b+n)}} \tag{128}$$

on $\mu_{0,n}(a,b)$ can be used for many values of n and ranges of $\langle a,b\rangle$ to produce quite tight bounds on the reliability predictions discussed in §5.3.5.2. These bounds are themselves in the ratio

$$1+\frac{an}{b(a+b+n)}$$

so that, for example, when a is small compared to b, we know that these bounds are at least correspondingly accurate approximations to $\mu_{0,n}(a,b)$. Returning to our application in §5.3.5.2, it is worth remarking that such values of a and b give very plausible distributions for P to characterise a family of (product, environment) pairs designed for very high reliability.

But, for those values, e.g. when $\frac{b}{a}$ is small, where these bounds are not known to give satisfactory accuracy, we can use the more general[9] and tighter bounds obtained using the Euler-Maclaurin summation formula, as follows.

Abramowitz and Stegun [3, p257] give bounds on the remainder $S_r(t)$ of the asymptotic expansion of the log-gamma function

$$\log(\Gamma(t))=(t-\tfrac{1}{2})\log(t)-t+\tfrac{1}{2}\log(2\pi)+\sum_{k=1}^{r}\frac{B_{2k}}{2k(2k-1)}\frac{1}{t^{2k-1}}+S_r(t)\,. \tag{129}$$

Here, $B_0,B_1,B_2,\ldots$ are the Bernoulli numbers $1,-\frac{1}{2},\frac{1}{6},\ldots$, see [3, pp804–10]. Note that if the terms in negative powers of t here, and the remainder term $S_r(t)$ are all neglected, and the remaining part of the right-hand side of (129) is then substituted in the definition of $\log(\mu_{0,n}(a,b))$,

[8] but note the improvement to the upper bound mentioned later on p241
[9] in that they are useful over a wider region of $\langle a,b,n\rangle$

we obtain the approximation $\mathcal{F}[t \mapsto (t - \frac{1}{2})\log(t)]$, where we have used the same linear functional notation $\mathcal{F}$ introduced above. It is straightforward to show[10] that this asymptotic approximation to $\log(\mu_{0,n}(a,b))$ lies between the two 'integration test' bounds obtained above. If, in place of (129), we consider instead the slightly easier problem of asymptotic approximation to *difference* of two values of the log-gamma function at arguments separated by an *integer*[11], then we can obtain information about the corresponding remainder term directly from the Euler-Maclaurin summation formula[12] [91, pp478–82]

$$\sum_{i=0}^{n-1} f(i) = \int_0^n f(t)dt + \frac{f(0)}{2} - \frac{f(n)}{2} + \sum_{k=1}^{r} \frac{B_{2k}}{(2k)!}\left(f^{(2k-1)}(n) - f^{(2k-1)}(0)\right) + Q_r \tag{130}$$

where[13]

$$Q_r = \int_0^1 \frac{B_{2r+1}(t)}{(2r+1)!} \sum_{i=0}^{n-1} f^{(2r+1)}(i+t)dt \tag{131}$$

which holds for any function f possessing the appropriate derivatives. (130) and (131) can be obtained from the integral form of the remainder terms for the ordinary Taylor series calculated for f and also for its derivatives using unit displacement from the series expansion point. (See [91] for details.) Applying this formula to the functions $f(x) = \log(b+x)$ and $f(x) = \log(a+b+x)$ and subtracting yields the formula

$$\begin{aligned}\log(\mu_{0,n}(a,b)) = \\ &\mathcal{F}[t \mapsto (t - \tfrac{1}{2})\log(t)] \\ &+ \sum_{k=1}^{r} \frac{B_{2k}}{2k(2k-1)}\left(-\frac{1}{(a+b+n)^{2k-1}} + \frac{1}{(b+n)^{2k-1}} + \frac{1}{(a+b)^{2k-1}} - \frac{1}{b^{2k-1}}\right) \\ &+ R_r\end{aligned} \tag{132}$$

with remainder

$$R_r = \int_0^1 \frac{B_{2r+1}(t)}{(2r+1)!}\left[-\psi^{(2r)}(t+a+b+n) + \psi^{(2r)}(t+b+n) + \psi^{(2r)}(t+a+b) - \psi^{(2r)}(t+b)\right]dt\,. \tag{133}$$

In this remainder term, $B_{2r+1}(t)$ is the Bernoulli polynomial [3, 804–6], and

$$\psi^{(2r)}(t) = \frac{d^{2r+1}}{dt^{2r+1}}\log(\Gamma(t)) \tag{134}$$

---

[10] either directly by subtraction, or by using (127) with $g(t) = (t+c)\log(t)$, $g''(t) = \frac{1}{t} - \frac{c}{t^2}$, monotonic decreasing in c

[11] This integer is the order of the $(1-P)$-moment. Does the E-M series which results (i.e. equations (132) and (133)) also hold exactly for non-integer n? For our purposes, this does not matter.

[12] Note that we do not use the E-M formula as given in [3, p806] since this contains errors.

[13] There are some alternative forms for the remainder in the Euler-Maclaurin summation formula. Note that, unlike some others which require $r \geq 1$, the form Q_r of remainder used here continues to be correct for $r = 0$ (provided that the 'empty sum' convention that $\sum_{k=1}^{0} \cdot = 0$ is used in (130)).

is the polygamma function [3, pp258–60]. The term in the integrand in square brackets is actually a shorthand, based on the basic polygamma recurrence relation

$$\psi^m(t+n) = \psi^m(t) + (-1)^m m! \left(\frac{1}{t^{m+1}} + \ldots + \frac{1}{(t+n-1)^{m+1}} \right) ,$$

for the expanded form (in which it was derived, as (131)).

The partial sum in (132) is a sum of alternating terms, since, for each term, the bracketed part $\mathcal{F}[t \mapsto t^{-(2k-1)}]$ is negative, and the even Bernoulli numbers (excluding B_0) are known to alternate in sign. We can show that, as one might hope to find, the sequence of *remainder terms* $\langle R_r \rangle$ of the approximation also alternates. To see this notice firstly that the square-bracketed term, $\xi(t)$, say, in (133) is a positive decreasing function of t. This fact is a consequence of putting $g = \psi^{(2r)}$ in (127) (and replacing b by $b+t$), since then we have $g'' = \psi^{(2r+2)}$ and the even polygamma functions[14] are known to be *increasing* (and negative) on the positive real axis. Alternatively, we obtain the same conclusion by leaving $\xi(t)$ in its original expanded form and rearranging the terms to give

$$\xi(t) = (2r)! \sum_{i=0}^{n-1} \left(\frac{1}{(t+b+i)^{2r+1}} - \frac{1}{(t+a+b+i)^{2r+1}} \right) ,$$

which is positive decreasing in t by the convexity of the inverse power function. The second requirement to deduce that $\langle R_r \rangle$ alternates in sign is the well-known property of the odd Bernoulli polynomials. [3, pp804–5] tells us that $B_{2r+1}(t)$ has a zero at $t = \frac{1}{2}$, has sign $(-1)^{r+1}$ on the interval $0 < t < \frac{1}{2}$ and satisfies the identity $B_{2r+1}(1-t) = -B_{2r+1}(t)$. Putting the above facts together we can deduce that the integrand in (133) has sign $(-1)^{r+1}$ on the interval $0 < t < \frac{1}{2}$, sign $(-1)^r$ on $\frac{1}{2} < t < 1$ and conclude that

$$\begin{aligned}
(-1)^{r+1} R_r &= \int_0^1 \frac{(-1)^{r+1} B_{2r+1}(t)}{(2r+1)!} \xi(t)\, dt \\
&> \int_0^{\frac{1}{2}} \frac{(-1)^{r+1} B_{2r+1}(t)}{(2r+1)!} \xi(\tfrac{1}{2})\, dt + \int_{\frac{1}{2}}^1 \frac{(-1)^{r+1} B_{2r+1}(t)}{(2r+1)!} \xi(\tfrac{1}{2})\, dt \\
&= \xi(\tfrac{1}{2}) \int_0^1 \frac{(-1)^{r+1} B_{2r+1}(t)}{(2r+1)!}\, dt \\
&= 0
\end{aligned}$$

Thus R_r has sign $(-1)^{r+1}$ and we can conclude that we have obtained bounds

$$\begin{aligned}
&\mathcal{F}[t \mapsto (t-\tfrac{1}{2}) \log(t)] \\
&\quad + \sum_{k=1}^{2s+1} \frac{B_{2k}}{2k(2k-1)} \left(-\frac{1}{(a+b+n)^{2k-1}} + \frac{1}{(b+n)^{2k-1}} + \frac{1}{(a+b)^{2k-1}} - \frac{1}{b^{2k-1}} \right) \\
&\quad < \log\left(\mu_{0,n}(a,b)\right)
\end{aligned}$$

[14] meaning the functions (134) with $r \geq 1$

$$< \mathcal{F}[t \mapsto (t - \tfrac{1}{2})\log(t)] + \sum_{k=1}^{2s} \frac{B_{2k}}{2k(2k-1)} \left(-\frac{1}{(a+b+n)^{2k-1}} + \frac{1}{(b+n)^{2k-1}} + \frac{1}{(a+b)^{2k-1}} - \frac{1}{b^{2k-1}} \right), \quad \text{for } s = 0, 1, 2, \ldots . \tag{135}$$

Note that the $s = 0$ case tells us that, in fact, $\mathcal{F}[t \mapsto (t - \frac{1}{2})\log(t)]$ is a strict *upper bound* for $\log(\mu_{0,n}(a,b))$. This enables a simple improvement to the right-hand side of (128).

Since we have now shown the sequence $R_0, R_1, R_2 \ldots$ to be alternating in sign, we have immediately

$$\begin{aligned} |R_r| &< |R_r - R_{r+1}| \\ &= \left| \frac{B_{2r+2}}{(2r+2)(2r+1)} \left(-\frac{1}{(a+b+n)^{2r+1}} + \frac{1}{(b+n)^{2r+1}} + \frac{1}{(a+b)^{2r+1}} - \frac{1}{b^{2r+1}} \right) \right| \\ &= \frac{(-1)^r B_{2r+2}}{(2r+2)(2r+1)} \left(\frac{1}{(a+b+n)^{2r+1}} - \frac{1}{(b+n)^{2r+1}} - \frac{1}{(a+b)^{2r+1}} + \frac{1}{b^{2r+1}} \right) \\ &< \frac{(-1)^r B_{2r+2}}{(2r+2)(2r+1)} \left(\frac{1}{b^{2r+1}} - \frac{1}{(a+b)^{2r+1}} \right) < \frac{(-1)^r B_{2r+2}}{(2r+2)(2r+1)b^{2r+1}} \end{aligned} \tag{136}$$

as a crude bound on the error of our approximation, so that, summarising our findings about the remainder term, we can say that

$$0 < (-1)^{r+1} R_r < \frac{(-1)^r B_{2r+2}}{(2r+2)(2r+1)b^{2r+1}} . \tag{137}$$

In the numerical results presented in §5.3.5.2, we chose to work with $r = 3$ and $r = 4$ to give us our lower and upper bounds (respectively) on $\mu_{0,n}(a,b)$. With these numbers of terms in the series, (137) becomes

$$-\frac{1}{1188b^9} < R_4 < 0 < R_3 < \frac{1}{1680b^7} \tag{138}$$

Two further problems remained to be addressed in order to implement an algorithm. The first problem is that of the size of the error bounds when b is small. Although for high reliability software families we would probably not expect an asymptote in the distribution of $P|(a,b)$ at $P = 1$, we might nevertheless in our prior distribution for $\langle a, b \rangle$ wish to assign a very small quantity of probability to such values. For this reason we prefer to use a numerical algorithm for $\mu_{0,n}(a,b)$ which is able to cope well with values of b close to or even less than 1. Fortunately, there is a relatively painless solution to this requirement. Examination of the remainder term (133) leads one to conclude that for small b the remainder is largely accounted for by the Euler-Maclaurin series' comparative inability to approximate the first few terms of the original series $\sum_{i=0}^{n-1}(\log(b+i) - \log(a+b+i))$. This suggests removing these few terms from the sum, say remove the first j terms.

$$\log(\mu_{0,n}(a,b)) = \left(\sum_{i=0}^{j-1} \log(b+i) - \log(a+b+i) \right) + \log(\mu_{0,n-j}(a, b+j))$$

Then these removed terms can be calculated directly, and the Euler-Maclaurin approximation used only for the later part of the sum which is equal to $\log(\mu_{0,n-j}(a, b+j))$. For j large enough so that we have $b+j$ greater than about 5 or 6, the new Euler-Maclaurin remainder will be very small. To be precise, for $b+j > 5$ as we in fact used in §5.3.5.2, we have from (138) an Euler-Maclaurin remainder satisfying

$$-4.310\times10^{-10} < R_4 < 0 < R_3 < 7.619\times10^{-9}\,.$$

Using this approach we avoided ever having to use the expansion (132) with any value of $b \leq 5$.

The second of the two problems mentioned is purely computational and has to do with the avoidance of a loss of precision on subtraction of very similar numbers, which could occur in several places due to the multiple occurrences of differences of the form $\mathcal{F}$, and to the fact that a, b and n may differ by quite large orders of magnitude. Our approach to avoiding such problems for the $\mathcal{F}[t \mapsto (t-\frac{1}{2})\log(t)]$ term in (132) is to write it as

$$\mathcal{F}[t \mapsto (t-\tfrac{1}{2})\log(t)] = -a\log\left(\frac{a+b+n}{a+b}\right) + \left(b-\frac{1}{2}\right)\log\left(\frac{a+b}{b}\right) - \left(b+n-\frac{1}{2}\right)\log\left(\frac{a+b+n}{b+n}\right)$$

except when $b \geq 100a$ in which case we note that the second and third of these three terms will begin to become very similar. (They are both asymptotic to a as $\frac{a}{b} \to 0$.) Therefore, under this condition we replace each by a Taylor series approximation with a subtracted to give

$$\mathcal{F}[t \mapsto (t-\tfrac{1}{2})\log(t)] = -a\log\left(\frac{a+b+n}{a+b}\right) + h\left(a, \frac{a}{b}\right) - h\left(a, \frac{a}{b+n}\right)$$

where

$$h(x,y) = -y\left(\left(\left(\left(\left(\frac{y}{6}-\frac{1}{5}\right)y+\frac{1}{4}\right)y-\frac{1}{3}\right)y+\frac{1}{2}\right)\left(x-\frac{y}{2}\right)+\frac{1}{2}\right).$$

Note also that the terms in the Euler-Maclaurin sum itself are *also* of the $\mathcal{F}$ form and so could likewise give rise to imprecision via subtraction. We avoid this problem by removing a factor a, rewriting (also for computational efficiency reasons) $\frac{1}{a}\left(1/(a+b)^{2k-1} - 1/b^{2k-1}\right)$ as a function of a^2, and $b(a+b)$, and similarly rewriting $\frac{1}{a}\left(-1/(a+b+n)^{2k-1} + 1/(b+n)^{2k-1}\right)$ as a function of a^2, and $(b+n)(a+b+n)$, to give, for $r=4$

$$\sum_{k=1}^{4}\frac{B_{2k}}{2k(2k-1)}\left(-\frac{1}{(a+b+n)^{2k-1}}+\frac{1}{(b+n)^{2k-1}}+\frac{1}{(a+b)^{2k-1}}-\frac{1}{b^{2k-1}}\right)$$
$$= \quad a\left[l\left((b+n)(a+b+n),\,a^2\right) - l\left(b(a+b),\,a^2\right)\right],$$

where

$$l(x,y) = \frac{1}{12x} - \frac{3x+y}{360x^3} + \frac{5x(x+y)+y^2}{1260x^5} - \frac{7x(x+y)^2+y^3}{1680x^7}\,.$$

Bibliography

[1] A. A. Abdel-Ghaly. *Analysis of Predictive Quality of Software Reliability Models.* PhD thesis, The City University, London, Center for Software Reliability, 1986.

[2] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood. Evaluation of competing software reliability predictions. *IEEE Transactions on Software Engineering*, 12:950–67, 1986.

[3] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions.* Dover, New York, 1970.

[4] A. W. Andrew. *A Class of Software Reliability Growth Models Based on Exercise Frequencies.* PhD thesis, Glasgow College, 1990.

[5] H. E. Ascher and H. Feingold. *Repairable Systems Reliability: Modelling, Inference, Misconceptions and Their Causes.* Marcel Dekker, New York, 1984.

[6] S. Brocklehurst, P. Y. Chan, and B. Littlewood. Adaptive software reliability modelling. In *Proc. Conference on Measurement for Software Control & Assurance*, 1987.

[7] Sarah Brocklehurst. *Software Reliability Prediction: A Multi-Modelling Approach.* PhD thesis, City University, London, Center for Software Reliability, February 1995.

[8] R. W. Butler and G. B. Finelli. The infeasibility of quantifying the reliability of life-critical real-time software. *IEEE Transactions on Software Engineering*, 19(1):3–12, 1993.

[9] X. Castillo and D. P. Siewiorek. Workload, performance and reliability of digital computing systems. In *Proc.* 11^{th} *Fault-Tolerant Computing Symposium*, pages 84–89. IEEE, 1981.

[10] P. Y. Chan. *Software Reliability Prediction.* PhD thesis, The City University, London, Center for Software Reliability, 1986.

[11] P. Y. Chan and B. Littlewood. Parametric spline approach to adaptive reliability modelling. Technical report, CSR, The City University, London, 1986.

[12] C. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society, Series B*, 34:187–220, 1972.

[13] C. R. Cox. Partial likelihood. *Biometrika*, 62(2):269–276, 1975.

[14] C. R. Cox and P. A. W. Lewis. *The Statistical Analysis of Series of Events.* Statistical Monographs. Methuen, London, 1966.

[15] C. R. Cox and D. Oakes. *Analysis of Survival Data.* Chapman & Hall, London, 1984.

[16] Attila Csenki. On Bayesian software reliability predictions. Alvey software reliability modelling project deliverable, CSR, The City University, London, March 1988.

[17] C. Dale. Software reliability models. In Mellor and Bendell [72], pages 31–44,244–7.

[18] C. Dale. Software reliability issues. In Rook [87], pages 1–19.

[19] A. P. Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–13, 1982.

[20] A. P. Dawid. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society, Series A*, 147, 1984.

[21] G. de Barra. *Measure Theory and Integration.* Mathematics and Its Applications. Ellis Horwood, Chichester, 1981.

[22] B. de Finetti. *Theory of Probability*, volume 1. Wiley, London, 1974.

[23] B. de Finetti. *Theory of Probability*, volume 2. Wiley, London, 1975.

[24] J. T. Duane. Learning curve approach to reliability monitoring. *IEEE Transactions on Aerospace*, 2:563–6, 1964.

[25] W. K. Ehrlich and J. P. Stampfel. Applications of software reliability modelling to product quality and test process. In *Proc.* 12^{th} *ICSE*, pages 108–16. IEEE, 1990.

[26] W. H. Farr. A survey of software reliability modelling and estimation. Technical Report NSWC TR 82-171, Naval Surface Weapons Centre, Dahlgren, Va., 1983.

[27] W. Feller. *An Inroduction to Probability Theory and Its Applications*, volume 2. Wiley, New York, 1966.

[28] W. Feller. *An Inroduction to Probability Theory and Its Applications*, volume 1. Wiley, New York, 3 edition, 1968.

[29] N. Fenton, B. Littlewood, and P. Mellor. *Software Reliability & Measurement.* Three Video Tapes & Accompanying Notes. City University, London, London, 1991.

[30] Norman Fenton. *Software Metrics: A Rigorous Approach.* Chapman & Hall, London, first edition, 1991.

[31] S. French, R. M. Cooke, and M. Wiper. The use of expert judgement in risk assessment. *Journal of the Institute of Mathematics and Its Applications*, 27:36–40, 1991.

[32] A. L. Goel and K. Okumoto. Time-dependent error-detection rate model for software and other performance measures. *IEEE Transactions on Reliability*, 28(3):206–11, August 1979.

[33] M. Henrion and B. Fischhoff. Assessing uncertainty in physical constants. *American Journal of Physics*, 54(9):791–8, 1986.

[34] D. Hunns and Wainwright. Software-based protection for Sizewell B: the regulator's perspective, September 1991.

[35] R. K. Iyer and D. J. Rossetti. Effect of system workload on operating system reliability: A study on IBM 3081. *IEEE Transactions on Software Engineering*, 11:1438–48, 1985.

[36] Z. Jelinski and P. Moranda. Software reliability research. In W. Friedberger, editor, *Computer Performance Evaluation*, pages 465–84. Academic Press, 1972.

[37] Z. Jelinski and P. Moranda. Estimation and prediction for a simple software reliability model. *The Statistician*, 37:319–25, 1988.

[38] J. D. Kalbfleisch and R. L. Prentice. *The Statistical Analysis of Failure Time Data.* Wiley, 1980.

[39] R. Kay. Proportional hazard regression models and the analysis of censored survival data. *Journal of the Royal Statistical Society, Series C*, 26:227–37, 1977.

[40] P. A. Keiller, B. Littlewood, D. R. Miller, and A. Sofer. On the quality of software reliability prediction. In J. K. Skwirzynski, editor, *Electronic System Effectiveness and Life Cycle Costing*, Nato ASI Series, Volume F3. Springer-Verlag, Heidelberg, 1983. See p451.

[41] M. G. Kendall and A. Stuart. *Inference and Relationship.* Volume 2 of [42], 1979.

[42] M. G. Kendall and A. Stuart. *The Advanced Theory of Statistics.* Three Volumes. Griffin, London, .

[43] M. G. Kendall, A. Stuart, and J. K. Ord. *Design and Analysis, and Time Series.* Volume 3 of [42], 1983.

[44] M. G. Kendall, A. Stuart, and J. K. Ord. *Distribution Theory.* Volume 1 of [42], fifth edition, 1987.

[45] J. F. C. Kingman and S. J. Taylor. *Introduction to Measure Theory and Probability.* Cambridge University Press, London, 1966.

[46] D. E. Knuth. Literate programming. *The Computer Journal*, 27(2):97–111, 1984.

[47] J. C. Laprie. For a product-in-a-process approach to software reliability evaluation. In *Proc. 3rd International Symposium on Software Reliability Engineering (ISSRE92)*, pages 134–9, Research-Triangle Park, USA, 1992. Invited Paper.

[48] J. C. Laprie, C. Beounes, M. Kaâniche, and K. Kanoun. The transformation approach to the modelling and evaluation of the reliability growth. In *Proc.* 20^{th} *Fault-Tolerant Computing Symposium*, pages 364–71. IEEE, 1990.

[49] J. F. Lawless. Regression models for Poisson process data. *Journal of the American Statistical Association*, 82:808–15, 1987.

[50] B. Littlewood. How to measure software reliability and how not to. *IEEE Transactions on Reliability*, 28(2):103–10, June 1979.

[51] B. Littlewood. Software reliability model for modular program structure. *IEEE Transactions on Reliability*, 28:241–6, 1979.

[52] B. Littlewood. Theories of software reliability: How good are they and how can they be improved? *IEEE Transactions on Software Engineering*, 6(5):489–500, 1980.

[53] B. Littlewood. Stochastic reliability-growth: A model for fault-removal in computer-programs and hardware designs. *IEEE Transactions on Reliability*, 30:313–320, 1981.

[54] B. Littlewood. Predicting software reliability. *Philosophical Transactions of the Royal Society of London*, A 327:513–527, 1989.

[55] B. Littlewood. Modelling growth in software reliability. In Rook [87], pages 137–53.

[56] B. Littlewood. Forecasting software reliability. In P. Sander and R. Badoux, editors, *Proc. ESRA Workshop on Bayesian Methods in Reliability, (Eindhoven)*. Kluwer Academic Publishers, 1991.

[57] B. Littlewood. *Software Reliability Measurement.* Volume 2 of [29], 1991.

[58] B. Littlewood, A. A. Abdel-Ghaly, and P. Y. Chan. Tools for the analysis of software reliability predictions. In J. K. Skwirzynski, editor, *Software System Design Methods.* Springer-Verlag, 1986.

[59] B. Littlewood and P. A. Keiller. Adaptive software reliability modelling. In *Proc.* 14^{th} *Fault-Tolerant Computing Symposium*, pages 108–13. IEEE, 1984.

[60] B. Littlewood and L. Strigini. Validation of ultra-high dependability for software-based systems. *Comm. Assoc. Computing Machinery*, 36(11):69–80, November 1993.

[61] B. Littlewood and L. Strigini. Software reliability and dependability: a roadmap. In A. Finkelstein, editor, *The Future of Software Engineering.* State of the Art Report given at 22^{nd} *Int. Conf. on Software Engineering*, pages 177–88, Limerick, June 2000. ACM, ACM Press.

[62] B. Littlewood and J. L. Verrall. A Baysian reliability growth model for computer software. *Journal of the Royal Statistical Society, Series C*, 22(3):332–46, 1973.

[63] B. Littlewood and D. Wright. Stopping rules for the operational testing of safety-critical software. In 25^{th} *Fault-Tolerant Computing Symposium (Pasadena, California), Digest of Papers*, pages 444–51. IEEE, IEEE Computer Society Press, 1995.

[64] B. Littlewood and D. Wright. Some conservative stopping rules for the operational testing of safety-critical software. *IEEE Transactions on Software Engineering*, 23(11):673–83, 1997.

[65] M. R. Lyu, editor. *Handbook of Software Reliability Engineering.* IEEE Computer Society Press, 1996. with enclosed CD containing software failure data sets.

[66] P. McCullagh and J. A. Nelder. *Generalised Linear Models.* Monographs on Statistics and Applied Probability. Chapman & Hall, London, 1983.

[67] P. Mellor. Analysis of software failure data (1): Adaptation of the Littlewood stochastic reliability growth model for coarse data. *ICL Technical Journal*, 1984.

[68] P. Mellor. Software reliability data collection: Problems and standards. In Mellor and Bendell [72], pages 165–81,256–7.

[69] P. Mellor. Software reliability modelling: The state of the art. *Information and Software Technology*, 29(2):81–98, March 1987. Tutorial.

[70] P. Mellor. *Data Collection for Software Reliability & Measurement.* Volume 3 of [29], 1991.

[71] P. Mellor and A. Bendell. Software reliability models. In *Software Reliability: State of the Art Report* [72], pages 319–46. Chapter 3 of 'Analysis' section.

[72] P. Mellor and A. Bendell, editors. *Software Reliability: State of the Art Report.* Pergamon Infotech, 1986.

[73] J. F. Meyer, B. Littlewood, and D. R. Wright. Dependability of modular software in a multiuser operational environment. In *Proc. 6th Int'l Symp. Software Reliability Engineering*, pages 170–9, Toulouse, France, October 1995.

[74] D. R. Miller. Exponential order statistic models of software reliability growth. *IEEE Transactions on Software Engineering*, 12:12–24, 1986.

[75] L. H. Miller. Table of percentage points of Kolmogorov statistics. *Journal of the American Statistical Association*, 51:111–21, 1956.

[76] W. M. Miller, L. J. Morell, R. E. Noonan, S. K. Park, D. M. Nicol, and B. W. Murrill. Estimating the probability of failure when testing reveals no failures. *IEEE Transactions on Software Engineering*, 18(1), 1992.

[77] J. D. Musa. Software reliability data. Technical report, Data and Analysis Center for Software, Rome Air Development Center, Rome, New York, 1980.

[78] J. D. Musa, A. Iannino, and K. Okumoto. *Software Reliability: Measurement, Prediction, Application.* McGraw-Hill International Editions, 1987.

[79] P. M. Nagel and J. A. Skrivan. Software reliability: Repetitive run experimentation and modelling. Technical Report BCS-40366, Boeing Computer Services Company, Seattle, Washington, 1982.

[80] H. R. Neave. *Statistics Tables.* Routledge, an imprint of Taylor & Francis Books Ltd, 1993.

[81] A. N. Pettit and I. Bin Daud. Investigating time-dependence in Cox's proportional hazards model. *Journal of the Royal Statistical Society, Series C*, 39:313–29, 1990.

[82] H. R. Pitt. *Measure and Integration for Use.* IMA Monograph Series. Oxford University Press, 1985.

[83] O. Pons and E. de Turckheim. Cox's periodic regression model. *Annals of Statistics*, 16(2):678–93, 1988.

[84] M. J. D. Powell. *Approximation Theory and Methods.* Cambridge University Press, Cambridge, 1981.

[85] R. L. Prentice, R. J. Williams, and A. V. Peterson. On the regression analysis of multivariate failure time data. *Biometrika*, 68:373–9, 1981.

[86] Requirements and Technical Concepts for Aeronautics. Software considerations in airborn systems and equipment certification. Technical Report DO-178B, Department of Defense, 1992.

[87] P. Rook, editor. *Software Reliability Handbook.* Elsevier Applied Science, London and New York, 1990.

[88] D. J. Rossetti and R. K. Iyer. Software related failures on the IBM 3081: A relationship with system utilisation. Technical Report 82-8, Center for Reliable Computing, Computer Systems Laboratory, Stanford University, 1982.

[89] J. C. Rouquet and Z. Z. Traverse. Safe and reliable computing on board the Airbus and ATR aircraft. In W. J. Quirk, editor, *Proc. Fifth IFAC Worshop on Safety of Computer Control Systems*, pages 93–97, Oxford, 1986. Pergamon Press.

[90] D. Schoenfield. Partial residuals for the proportional hazards regression model. *Biometrika*, 69:239–41, 1982.

[91] M. Spivak. *Calculus.* World Student Series Edition. Addison-Wesley, Menlo Park, California, 1973.

[92] V. G. Vovk. A logic of probability with applications to the foundations of statistics (with discussion). *Journal of the Royal Statistical Society, Series B*, 55:317–51, 1993. Read at RSS meeting on 17/9/92.

[93] D. W. Wightman and A. Bendell. Proportional hazards modelling of software failure data. In Mellor and Bendell [72], pages 229–42,261–3.

[94] D. R. Wright. Recalibrated prediction of some software failure-count sequences. Project deliverable, CSR, City University, London, September 1993. In First Year Report of ESPRIT-funded 'Predictably Dependable Computing Systems' Project (6362 - PDCS2).