Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis

NASA/SP-2009-569

Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis

Dr. Homayoon Dezfuli
NASA Project Manger, NASA Headquarters, Washington, DC

Dana Kelly
Idaho National Laboratory, Idaho Falls, ID

Dr. Curtis Smith
Idaho National Laboratory, Idaho Falls, ID

Kurt Vedros
Idaho National Laboratory, Idaho Falls, ID

William Galyean
Idaho National Laboratory, Idaho Falls, ID

National Aeronautics and
Space Administration

June 2009

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at 443-757-5803

- Phone the NASA STI Help Desk at 443-757-5802

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

Foreword

This NASA-HANDBOOK is published by the National Aeronautics and Space Administration (NASA) to provide a Bayesian foundation for framing probabilistic problems and performing inference on these problems. It is aimed at scientists and engineers and provides an analytical structure for combining data and information from various sources to generate estimates of the parameters of uncertainty distributions used in risk and reliability models. The overall approach taken in this document is to give both a broad perspective on data analysis issues and a narrow focus on the methods required to implement a comprehensive database repository. It is intended for use across NASA.

Recent years have seen significant advances in the use of risk analysis at NASA. These advances are reflected both in the state of practice of risk analysis within projects, and in the status of several NASA requirements and procedural documents. Because risk and reliability models are intended to support decision processes, it is critical that inference methods used in these models be robust and technically sound. To this end, the Office of Safety and Mission Assurance (OSMA) undertook the development of this document. This activity, along with other ongoing OSMA-sponsored projects related to risk and reliability, supports the attainment of the holistic and risk-informed decision-making environment that NASA intends to adopt.

This document is not intended to prescribe any technical procedure and/or software tool. The coverage of the technical topics is also limited with respect to (1) the historical genesis of Bayesian methods; (2) comparisons of "classical statistics" approaches with Bayesian ones; (3) the detailed mathematics of a particular method (unless needed to apply the method); and (5) a source of actual reliability or risk data/information. Additionally, this document is focused on hardware failures; excluded from the current scope are specific inference approaches for phenomenological, software, and human failures. As with many disciplines, there are bound to be differences in technical and implementation approaches. The authors acknowledge that these differences exist and to the extent practical these instances have been identified.

The *Bayesian Inference* handbook assumes that probabilistic inference problems range from simple, well-supported cases to complex, multi-dimensional problems. Consequently, the approaches provided to evaluate these diverse sets of issues range from single-line spreadsheet formula approaches to Monte Carlo-based sampling methods. As such, the level of analyst sophistication should be commensurate with the issue complexity and the selected computational engine. To assist analysts in applying the inference principles, the document provides "call out" boxes to provide definitions, warnings, and tips. In addition, a hypothetical (but representative) system analysis and multiple examples are provided, as are methods to extend the analysis to accommodate real-world complications such as uncertain, censored, and disparate data.

For most of the example problems, the *Bayesian Inference* handbook uses a modern computational approach known as Markov chain Monte Carlo (MCMC). Salient references provide the technical basis and mechanics of MCMC approaches. MCMC methods work for simple cases, but more importantly, they work efficiently on very complex cases. Bayesian inference tends to become computationally intensive when the analysis involves multiple parameters and correspondingly high-dimensional integration.

MCMC methods were described in the early 1950s in research into Monte Carlo sampling at Los Alamos. Recently, with the advance of computing power and improved analysis algorithms, MCMC is increasingly being used for a variety of Bayesian inference problems. MCMC is effectively (although not literally) numerical (Monte Carlo) integration by way of Markov chains. Inference is performed by sampling from a target distribution (i.e., a specially constructed Markov chain, based upon the inference problem) until convergence (to the posterior distribution) is achieved. The MCMC approach may be implemented using

custom-written routines or existing general purpose commercial or open-source software. In the *Bayesian Inference* document, an open-source program called OpenBUGS (commonly referred to as WinBUGS) is used to solve the inference problems that are described. A powerful feature of OpenBUGS is its automatic selection of an appropriate MCMC sampling scheme for a given problem. The approach that is taken in the document is to provide analysis "building blocks" that can be modified, combined, or used as-is to solve a variety of challenging problems. Not just for risk and reliability inference, MCMC methods are also being used for other U.S. Government activities (e.g., at the Food and Drug Administration, the Nuclear Regulatory Commission, National Institute of Standards and Technology, and the National Atmospheric Release Advisory Center).

The MCMC approach used in the document is implemented via textual scripts similar to a macro-type programming language. Accompanying each script is a graphical diagram illustrating the elements of the script and the overall inference problem being solved. In a production environment, analysis could take place by running a script (with modifications keyed to the problem-specific information). Alternatively, other implementation approaches could include: (1) using an interface-driven front-end to automate an applicable script, (2) encapsulating an applicable script into a spreadsheet function call, or (3) creating an automated script-based inference engine as part of an information management system. In lieu of using the suggested MCMC-based analysis approach, some of the document's inference problems could be solved using the underlying mathematics. However, this alternative numerical approach is limited because several of the inference problems are difficult to solve either analytically or via traditional numerical methods.

As a companion activity to this handbook, a data repository and information management system is being planned by OSMA. The approaches described in the report will be considered in the development of the inference engine for this system. While not a risk or reliability data source itself, the *Bayesian Inference* document describes typical sources of generic and NASA-specific data and information. Further, examples of failure taxonomies and associated hierarchies of information are discussed. Some of the examples and case studies use real data – nonetheless the inference *results* produced in this document should not be used for analysis.

Follow-up activities to this document include developing a stand-alone supporting document describing the technical detail of the methods described herein. It is important to note that having the detailed mathematics is not required to run or understand the approaches described in the report. A limited set of the examples provided in this report have been validated, but the results and associated tools will be further checked for correctness via a more formal approach. Additional examples will be introduced in future revisions.

Comments and questions concerning the contents of this publication should be referred to the National Aeronautics and Space Administration, Director, Safety and Assurance Requirements Division, Office of Safety and Mission Assurance, Washington, DC 20546.

Dr. Homayoon Dezfuli
Project Manager, NASA Headquarters
June 2009

Contents

Tables

Figures

VIII

Scripts

XI

Examples

XIII

Acknowledgements

The following individuals are recognized for their contributions, comments, and reviews to this handbook:

Acronyms

ASME	American Society of Mechanical Engineers
ATCS	active thermal control system
BGR	Brooks, Gelman and Rubin convergence statistic
BUGS	Bayesian updating using Gibbs sampling
CCF	common cause failure
DIC	deviance information criterion
DAG	directed acyclic graph
EPIX/RADS	Equipment Performance Information Exchange/Reliability and Availability Database System
FOIA	Freedom of Information Act
GIDEP	Government-Industry Data Exchange Program
GSFC	Goddard Space Flight Center
IEEE	Institute of Electrical and Electronics Engineers
INL	Idaho National Laboratory
ISS	International Space Station
ITAR	International Traffic in Arms Regulations
MCMC	Markov chain Monte Carlo
MGL	multiple Greek letter
MLE	maximum likelihood estimation
MMOD	micro meteoroid orbital debris
MTBF	mean time between failure
MTTF	mean time to failure
NASA	National Aeronautics and Space Administration
NPR	NASA Procedural Requirements
NPRD	Non-Electronic Parts Reliability Data
NUCLARR	Nuclear Computerized Library for Assessing Reactor Reliability
OREDA	Offshore Reliability Data
OSMA	Office of Safety and Mission Assurance
PLC	programmable logic computer
PM	performance measures
PRA	probabilistic risk assessment
PRACA	Problem and Corrective Action
PVC	population variability curve
RIAC	Reliability Information Analysis Center
ROCOF	rate of occurrence of failures
RTG	radioisotope thermoelectric generator
STRATCOM	Strategic Command

This page left blank

1. Introduction

1.1 Risk and Reliability Data Analysis Objectives

This document, Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis, is intended to provide guidelines for the collection and evaluation of risk and reliability-related data. It is aimed at scientists and engineers familiar with risk and reliability methods and provides a hands-on approach to the investigation and application of a variety of risk and reliability data assessment methods, tools, and techniques. This document provides both:

- A broad perspective on data analysis collection and evaluation issues.
- A narrow focus on the methods to implement a comprehensive information repository.

The topics addressed herein cover the fundamentals of how data and information are to be used in risk and reliability analysis models and their potential role in decision making. Understanding these topics is essential to attaining a risk informed decision making environment that is being sought by NASA requirements and procedures such as 8000.4 (Agency Risk Management Procedural Requirements), NPR 8705.05 (Probabilistic Risk Assessment Procedures for NASA Programs and Projects), and the System Safety requirements of NPR 8715.3 (NASA General Safety Program Requirements).

1.2 Taxonomy of Uncertainty

The word "uncertainty" is, itself, ironic in that it has a variety of meanings and has a variety of synonyms: error, information, vagueness, scatter, unknown, discord, undefined, ambiguous, probability, stochastic, distribution, confidence, chance, and so forth. This vagueness in terminology needs to be minimized – to the extent possible – to institutionalize the methods described herein. In this report, definitions are provided in shaded boxes. A list of definitions is also provided in Appendix A.

Data	Distinct **observed** (e.g., measured) values of a physical process. Data may be factual or not, for example they may be subject to uncertainties, such as imprecision In measurement, truncation, and interpretation errors.
Information	The result of evaluating, processing, or organizing data/information in a way that adds to knowledge.
Knowledge[1]	What is known from gathered information.
Inference	The process of obtaining a conclusion based on what one knows.

Examples of data include the **number** of failures during system testing, the **pressure** pulse recorded during hydrogen detonations, the **times** at which a component has failed and been repaired, and the **time** it takes until a heating element fails. In these examples, the measured or observed item is bolded to emphasize that data are observable. Note, however, that information is not necessarily observable; only the *subset* of information that is called data is observable. The availability of data/information, like other types of resources, is crucial to analysis and decision-making. Furthermore, the process of collecting, storing, evaluating, and retrieving data/information affects its organizational value.

Alone, that is, outside of a structured inference process, data has little utility. For example, recording the pressure during a hydrogen detonation test as a function of distance from the detonation point provides a relationship of pressure and distance (as seen in Figure 1-1). This data records a deterministic relationship between pressure and distance, but the data does not convey much information that would be of use in risk evaluations, it only records a set of ordered pairs of pressure and distance. The data does **not** indicate the potential variability in hydrogen detonation, the consistency (were the data

---

[1] "Science is organized knowledge. Wisdom is organized life." - Immanuel Kant

recorded correctly?), why the pressure decreases the further away one is from the detonation, or whether the overpressure will be sufficiently large to fail structures of interest. Data alone cannot answer such questions. Instead, one must **analyze** the data to produce information, information that will ultimately be used for making inferences, whereby such questions may be answered.

Information is the foundation for knowledge. Examples of information include an empirical estimate of system failure probability, the probability that a hydrogen detonation deforms a heat shield, an expert's opinion of a component's failure probability, and the failure rate for a heating element. Like data,

Figure 1-1. Example observed data from a hydrogen detonation (hydrogen/oxygen mix, 0.258 lb hydrogen).
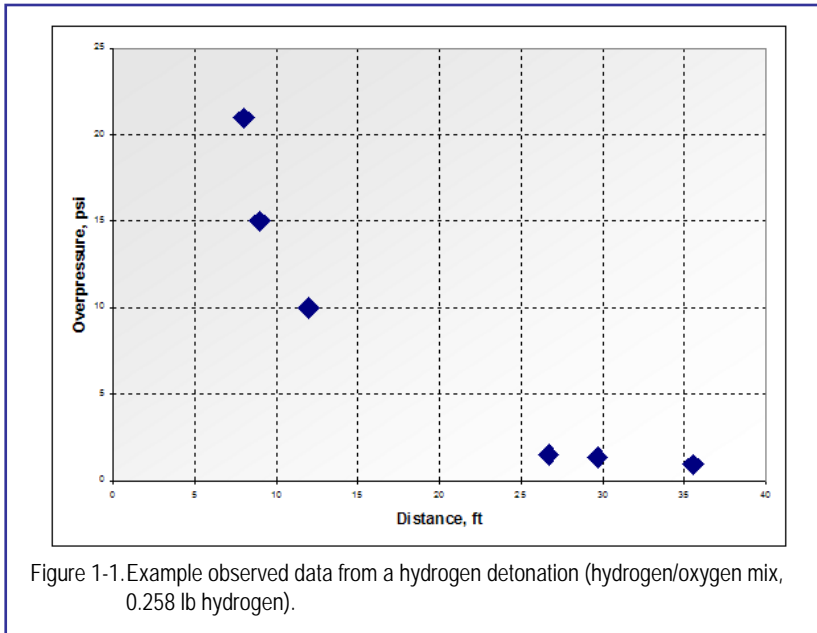
information has organizational value. Key to this value is that information should be scientifically defensible and support required inference processes. Since processes of concern cover a variety of activities, any data/information collection activity and repository must support not only changing needs but modifications to requirements and applications.

Since data are the subset of information that is observable or measurable, there is an implied contextual structure to the information conditional on the data context. However, as described later in this report, knowledge about specific context may be generalized to other situations. This generalization concept is an integral part of any inference process.

> Context   The set of environmental conditions and circumstances, both helpful and adverse, that surround a reliability or risk-relevant event. In the analysis of human performance, context includes both system-manifested impacts (e.g., available time, ergonomics, and task complexity) and human-manifested impacts (e.g., stress level, degree of training, fatigue). In the analysis of hardware or software performance, context includes the operational environments and interactions with other hardware, software, and human elements.

To evaluate or manipulate data, we must have a "model of the world" (or simply "model") that allows us to translate real-world observables into information. [Winkler, 1972; Apostolakis, 1994] Within this model of the world, there are two fundamental types of model abstractions, aleatory and deterministic. The term "aleatory" when used as a modifier implies an inherent "randomness" in the outcome of a process. For example, flipping a coin[1] is modeled as an aleatory process, as is rolling a die. When flipping a coin, the "random" but observable data are the outcomes of the coin flip, that is, heads or tails. Note that since

---

[1] Flipping a coin is deterministic in principle, but the solution of the "coin-flipping dynamics" model, including knowledge of the relevant boundary conditions, is too difficult to determine or use in practice. Hence, we abstract the flipping process via an aleatory model of the world.

probabilities are **not** observable quantities, we do not have a model of the world directly for probabilities. Instead, we rely on aleatory models (e.g., a Bernoulli[1] model) to predict probabilities for observable outcomes (e.g., two heads out of three tosses of the coin).

Model (of the world)	A mathematical construct that converts information (including data as a subset of information) into knowledge. Two types of models are used for risk analysis purposes, aleatory and deterministic.
Aleatory	Pertaining to stochastic (non-deterministic) events, the outcome of which is described by a probability. From the Latin *alea* (game of chance, die).
Deterministic	Pertaining to exactly predictable (or precise) events, the outcome of which is known with certainty if the inputs are known with certainty. As the antitheses of aleatory, this is the type of model most familiar to scientists and engineers and include relationships such as $E=mc^2$, $F=ma$, $F=G\,m_1\,m_2/r^2$, etc.

The models that will be described herein are parametric, and most of the model parameters are themselves imprecisely known, and therefore uncertain. Consequently, to describe this second layer of uncertainty, we introduce the notion of epistemic uncertainty. **Epistemic** uncertainty represents how precise our state of knowledge is about the model (including its parameters), regardless of the type of model. [Eerola, 1994] Whether we employ an aleatory model (e.g., Bernoulli model) or a deterministic model (e.g., applied stress equation), **if** any parameter in the model is imprecisely known, then there is epistemic uncertainty associated with the model (for examples of epistemic distributions, see Appendix B or Figure 4-2). Stated another way, if there is epistemic uncertainty associated with the parametric inputs to a model, then there is epistemic uncertainty associated with the output of the model, as well.

Epistemic	Pertaining to the degree of knowledge of models and their parameters. From the Greek episteme (knowledge).

It is claimed that models have epistemic uncertainty, but is there epistemic uncertainty associated with other elements of our uncertainty taxonomy? The answer is yes, and in fact almost all parts of our taxonomy have a layer of epistemic uncertainty, including the data, context, model information, knowledge, and inference.

In summary,
- We employ mathematical models of reality, both deterministic and aleatory.
- These models contain parameters – whose values are estimated from information – of which data are a subset.
- Uncertain parameters (in the epistemic sense) are inputs to the models used to infer the values of future observables, leading to an increase in scientific knowledge. Further, these parameters may be known to high precision and thus have little associated epistemic uncertainty (e.g., the speed of light, the gravitational constant), or they may be imprecisely known and therefore subject to large epistemic uncertainties (e.g., frequency of lethal solar flares on the Moon, probability of failure of a component).

Visually, our taxonomy appears as shown in Figure 1-2.

---

[1] A Bernoulli trial is an experiment outcome that can be assigned to one of two possible states (e.g., success/failure, heads/tails, yes/no). The outcomes are assigned to two values, 0 and 1. A Bernoulli process is obtained by repeating the same Bernoulli trial, where each trial is independent. If the outcome assigned to the value 1 has probability p, it can be shown that the summation of n Bernoulli trials is binomial distributed ~ Binomial(p, n).

Knowledge

Knowledge is used as a part of future inference

Model (Aleatory)

Information (Via Inference)

Data (Observable)

Engineering Information

Information

Reality
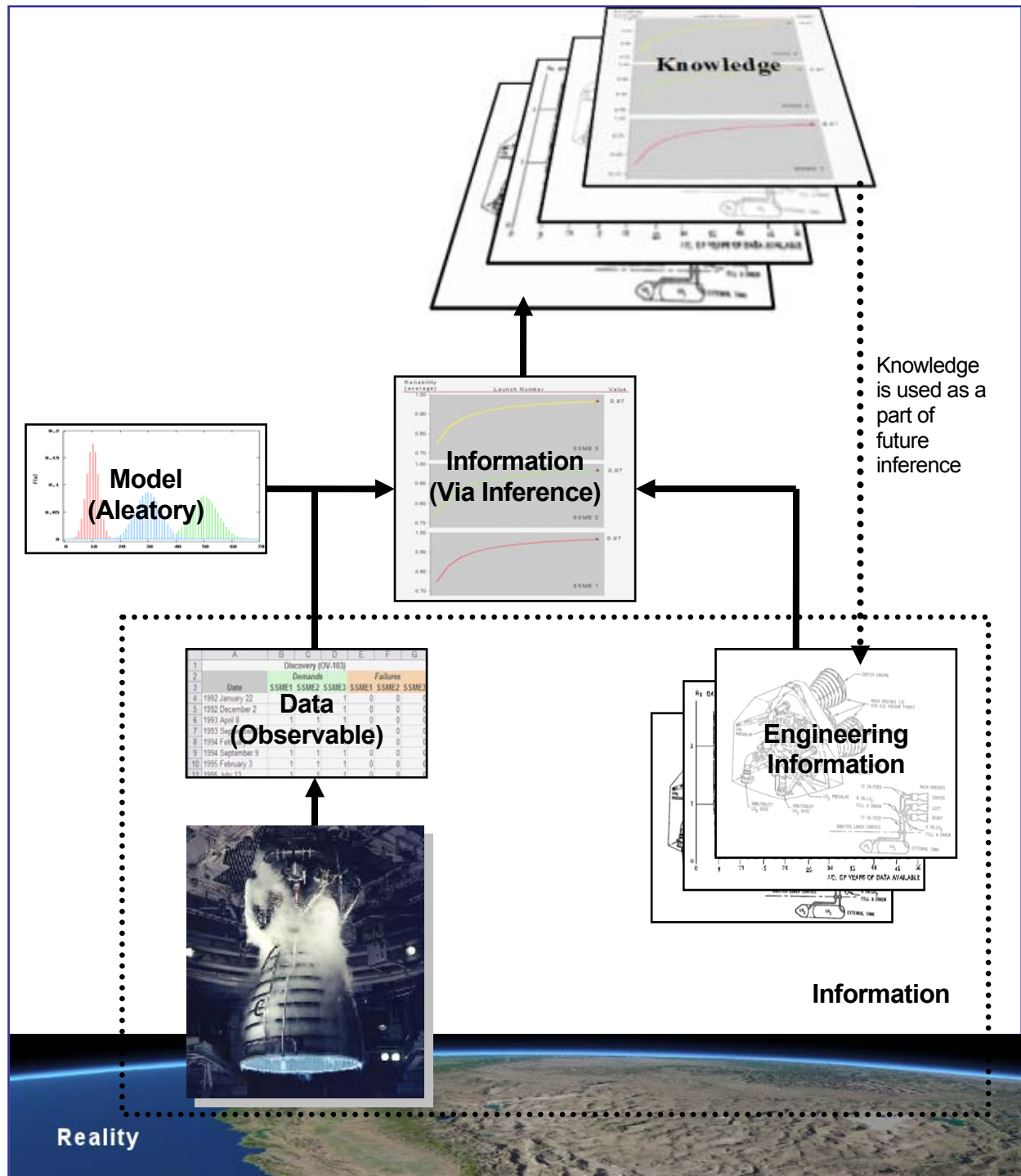
• Figure 1-2. Representation of the data, modeling, analysis, and inference approach that is the basis for taxonomy of uncertainty used in this document (for example, in estimating the Space Shuttle Main Engine demand failure probability).

1.3 Data-Model Relationship

1.3.1 Modeling system performance

A multitude of complex factors affect system performance. For example, operating a trajectory console while fatigued increases the likelihood of an error, a caustic environment increases the likelihood of corrosion-induced failure, and large overpressure during an explosion may cause structural deformation, leading to structural failure. In each of these scenarios, there is at least one causal mechanism linking the event to the resulting outcome. For example, in the case of an error at the trajectory console, fatigue influences the probability of experiencing the negative outcome. If one evaluates statistics for the rate of console-based errors, the **correlation** between drowsiness and the chance of an error may be high. In general though, one must infer carefully, because correlation does not automatically imply causation. The time-worn example of the high positive correlation between crime rate and the density of churches illustrates this point. Pearl, in his text on causation, notes, "Causation is much harder to measure than correlation." [Pearl, 2000] If correlation is the throw rug purchased at the local department store, causation is the Persian tapestry, much more rich, intricate, and interwoven.

For the types of inference described in this report, we will need to determine how to make inference in relation to associated causal factors related to risk. In many cases, we may not have (or choose not to use) a model describing the causal factors at work due to the complexity of the scenario. In these cases, we rely for inference on aleatory models.

> **Hill, in his work on causation related to disease [Hill, 1965] suggests that causal associations are themselves affected by considerations such as (in order of importance, with most important first): strength, consistency, specificity, temporality, biological gradient, plausibility, coherence, experiment, and analogy.**

We have already mentioned aleatory models such as the Bernoulli. We call these models **parametric** because the key feature of these approaches is that a complex, unpredictable behavior is described not via causal mechanisms but by a simple, high-level set of mathematical rules, typically expressed in terms of only one or two parameters. In the case of a heating element fails-to-operate scenario, a PRA analyst typically would use a Poisson[1] process model and would be concerned with only two parameters, the heating element's failure rate, , which is to be estimated, and the time required to operate, t (or mission time), which is specified. Absent from these types of models are parameters for the explicit causes of failures – no breakage of the element due to vibration, no short due to current surge, no failures due to low-temperature embrittlement, etc. This abstraction of a complex failure environment to a simple aleatory model (the Poisson process, with the key parameter "failure rate") is illustrated in Figure 1-3. Note that in this Poisson model, we do not explicitly model impact due to the environment, the heating element itself, or the operation. Instead these mechanisms are included implicitly in the failure rate (a model parameter), which determines the probability of observed outcomes such as failure.

---

[1] The Poisson model is defined in Section 4.2.2.
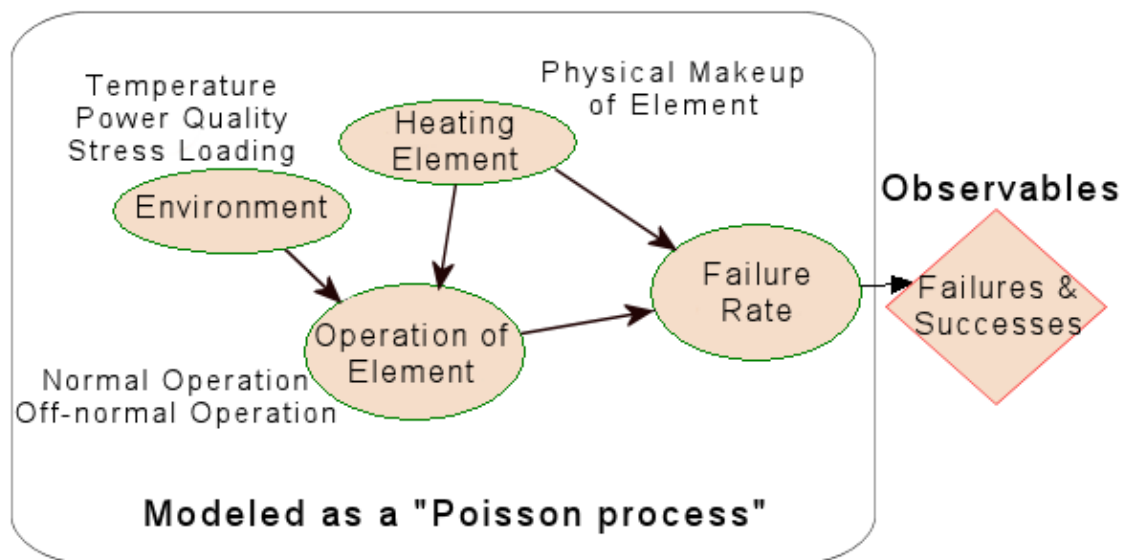
- Figure 1-3. Abstracting a complex failure environment into a simple aleatory (parametric) model.

A more complex type of aleatory model, in which causes of failure are modeled more explicitly, but which is less frequently encountered, is sometimes termed a **reliability-physics** model. In general, this is a model that attempts to predict the reliability of a complex process by using detailed, physics-based deterministic models associated with observable outcomes which themselves contribute to failure. For example, in the aging study described in [Smith et al, 2001], the researchers used a flow-accelerated corrosion model to predict pipe wall thinning. Within this aleatory model of the corrosion process, the researchers concerned themselves with 13 environmental variables that affected corrosion, including the operational time, the pipe material, the pipe geometry, the fluid velocity, the water chemistry, the water temperature, and the initial pipe thickness. The rate of pipe corrosion, via a causation mechanism, then affected the **probability** of a pipe failure.

Reliability	The degree of plausibility that an item (e.g., component or system) would not fail during a specified time (or mission).

To a large degree, current PRAs do not use such causal-based aleatory models, with some exceptions seen in phenomenological, structural, and human failure modeling. Analysts generally do not use reliability-physics aleatory models for initiating events or hardware failures even though, in theory, they certainly could. For example, if one reviewed hardware failure "causes," one could surmise a variety of items that do impact hardware reliability. A short list of these hardware performance shaping factors (to borrow a term from human reliability analysis) include:

"Structural failure, explosions, blockage, dust, metal fragments, thermal stresses, bending moments, lightning, short circuits, fabrication errors, erosion, fatigue, embrittlement..." [Andrews and Moss, 2002]

While the measurement of correlation is not an exact science, neither is that of causation. Dodge, in discussing uncertainty in his experimental human psychology research in the 1920s, noted, "Variability is quite as real as the central value." [Dodge, 1924] Consequently, both correlation and causation may be described by inference methods since both can be described by different types of probabilistic models. However, one common element between these two models is the data used to perform the analysis.

1.3.2 How simple is tossing a coin?

We have explained that tossing a coin is typically evaluated using an aleatory model rather than a deterministic one. However, since coin-tossing is such a commonplace process, we have many opportunities (at least in principle) to study the deterministic details of the coin-tossing process. Instead, we choose to abstract the complex physics of the toss into a simple aleatory model called the Bernoulli model – this is a **modeling choice**. By using this Bernoulli model, we can make predictions about the probability of seeing a certain outcome (e.g., H H T, in three tosses of the coin, where H=heads and T=tails). In reality though, the physics surrounding the coin toss are quite complex. Concepts such as air friction of the coin, the center of gravity (centroid) of the coin, the initial velocity (angular and rectilinear), etc., all come into play for the coin physics. Many different causal factors are involved such as the local environment of the toss and the person doing the tossing. A question one might ask is where do these factors go when we use the simpler Bernoulli model?

We can identify a variety of physical manifests (the person providing the coin, the way the coin is tossed, the environment surrounding the coin, etc.) that play a role in the process of tossing a coin. However, when we turn to the Bernoulli model, these physical manifests are summarily hidden. In place of these complex physical influences, we use the simple model[1] known as a Bernoulli model and its associated aleatory likelihood function, the binomial formula for multiple tosses. Note that, within this likelihood, the only **parameter** of interest is the coin toss outcome probability (i.e., the probability of obtaining a head, p). In other words, we acknowledge that the procedure of tossing a coin is quite involved and has a variety of real factors that influence the toss. Our predictive ability to explicitly take these influences into account and determine observable outcomes – in our case the number of heads versus tails – is quite limited.

Rather than relying on a deterministic model that represents the process of interest, we instead abstract the process to a higher level (a Bernoulli model) which, in this case, is at the highest level possible, that of utilizing the outcome probability directly in the model. Consequently, the **data** that is required for this coin-toss model is also at the highest level possible, that of an observed outcome (heads or tails) on a toss of the coin.

> The "bi" in binomial indicates two possible outcomes, head/tails, pass/fail, failure/success, etc.
>
> If more than two outcomes are possible (e.g., failure, degraded failure, success), then a multinomial model may be used. It may also be possible to use the multinomial in a decomposition of failures, say failure due to stress, failure due to training, failure due to complexity, but in these cases, we are still abstracting a complex process by a high-level model.

If we chose to model the coin toss by including the causal factors in our inference process, we might develop a graphical model similar to that shown in Figure 1-4. This type of model is known as an influence diagram and is an acyclic graph wherein nodes represent influencing factors and the arcs (or arrows) between nodes indicate causality among these factors. In our coin-tossing model, we are ultimately trying to make inference as to the coin toss outcome probability. This probability might be that of tossing a head, for example. Factors that affect this probability include (but are not necessarily limited to) items such as the coin itself (an asymmetrically weighted coin would increase the chance of seeing a head), the environment (it might be more difficult to influence the outcome when tossing a coin in a strong wind versus the still confines of a room), and the tossing mechanism. While we can make this graphical model as complex as needed **and** we may have the computational tools (software/hardware) to attempt to analyze this model, we usually choose not to use such a detailed model. Instead, we abstract this incredibly rich and complex process into the much simpler Bernoulli model. It is this abstraction procedure that is a **key** element to defining the **data requirements** since the type of model chosen determines the type of data needed to make inference about the model parameters and to check the validity of the model.

---

[1] "To choose a simple model because data are scarce is like searching for the key under the light in the parking lot." - Pierre Baldi

However, as we will describe later, it is possible within the Bayesian framework to infer low-level information when only high-level information or data are available just as easily as inferring high-level information when low-level information or data are available.
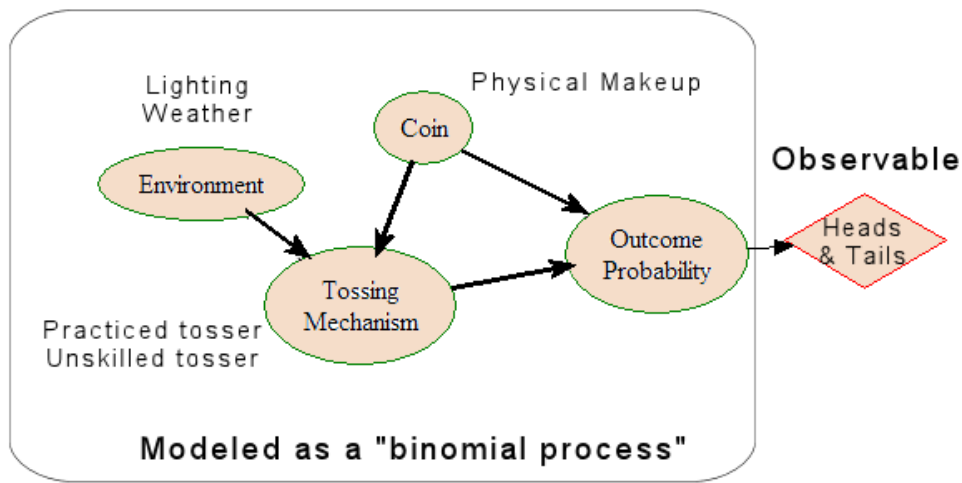
• Figure 1-4. A causal model representing the physics of tossing a coin as a function of causal factors.

1.3.3 What is data?

Earlier, data was defined as distinct observed (i.e., measured) outcomes of a physical process. It was noted that data may be factual or not and that data may be known with complete certainty or not. Later in this document, we will describe processes that are available to quantitatively evaluate both situations.

In our definition, we stated that data are an observable quantity. This implies a temporal constraint on the collection of data: data must be collected prior to the present moment (defined as t_0). While it is possible to talk about data being collected in the future, such data does not exist until it is actually observed. If we ask experts to predict what might happen in the future – for example, to predict how many U. S. launch vehicle failures we can expect to see over the next 10 years – this prediction is not a data collection activity. Instead, these opinions are information, not data, since they are not observed.[1] However, it would be **incorrect** to then assume either:

- This information has less organizational value than available data.
- This information cannot be used quantitatively.

Information, in all its various forms, is just **as useful** as data and may have more organizational value than data, depending on the type, quality, and quantity of the data/information. And when we describe the Bayesian approach to inference, we will see that information is as useful as data, it is just used in a different manner.

When we use data (which, by definition, occurs after t_0), we are transforming the data into information. This transformation takes place in conjunction with a model. Note that it is possible to construct a model, collect data, and then produce information about an event that has occurred in the past – this approach occurs frequently in science because the past (especially the distant past) may be just as uncertain as the future. An illustration of this for past events is shown in Figure 1-5 where temperature deviations (in the Northern Hemisphere) over the last 1,000 years are shown, along with uncertainty estimates derived via data and models. In this figure, measured values of temperature, which we call "data," are shown in red (called "direct temperatures" on the chart) while inferred information (not measured) are shown in blue (called "proxy data"). In this figure, the uncertainties are shown in grey.

---

[1] Such information about future outcomes of a physical process is sometimes referred to as "predicted data."
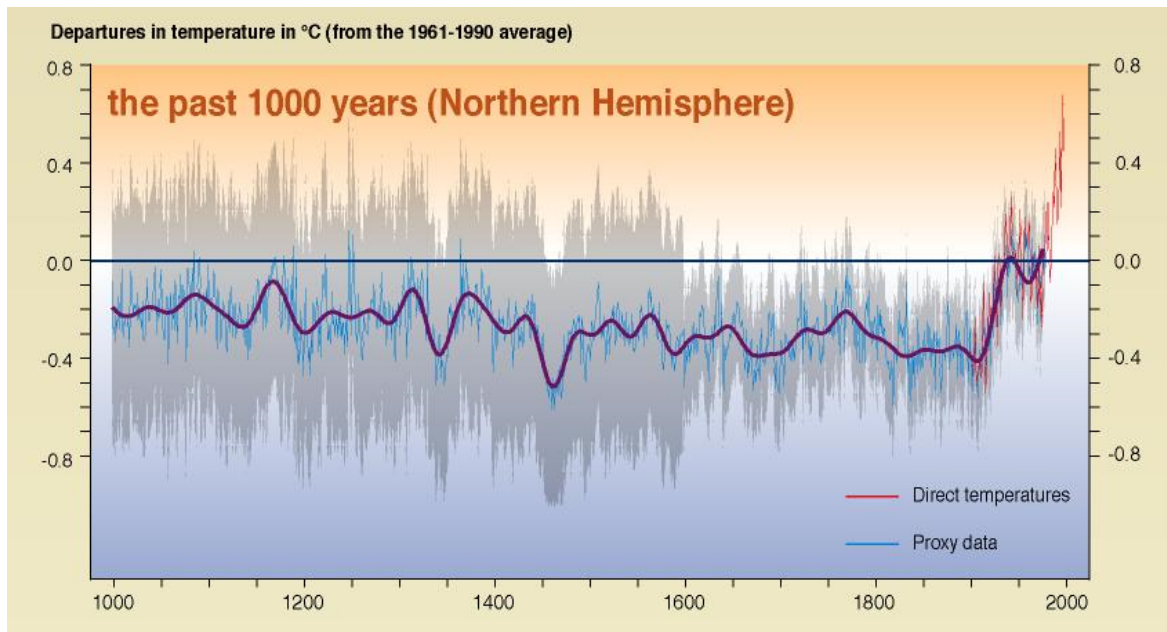
- Figure 1-5. Variations in the Earth's surface temperature over the last millennia. (Source: http://www.ipcc.ch/graphics/graphics/2001syr/large/05.16.jpg)

In summary,

- We model complex processes using either aleatory or deterministic models, where we have divided the aleatory classification into simple parametric models, such as the Bernoulli model for coin-tossing and more complex reliability-physics models, such as the one used for flow-accelerated corrosion by [Smith et al, 2001].
- These models require information for support, but the type of information varies from one model to the next.
- What is known about these models and information, including data, may be uncertain, leading to a layer of epistemic uncertainty.

If our model is as simple as $Z = \text{data}_1 + \text{data}_2$, we still end up with information (Z), but this is not a very useful model since it does not predict anything. For example, if $\text{data}_1 = 57$ marbles and $\text{data}_2 = 42$ marbles, then the model returns the information 99 marbles (spread across two measurements). This model simply returns the aggregation of what we already saw. Real organizational value from data collection and analysis comes from the "processing, manipulating, and organizing" process when we obtain information.

What if, however, instead of data we do not know how many marbles are in our containers? We can still use the same model ("+"), but in this case we would have epistemic uncertainty on the parameters of the model and the model would need to be modified slightly $Z = d_1 + d_2$, where each d_i is an uncertain quantity. In this case, we will still use the data we have (if any) but we will need to **infer** the possible values of d_i. Since the parameters of this model are uncertain, the information (Z) is uncertain. To use this information in decision-making for NASA processes, we need to identify, describe, and understand this uncertainty.

Uncertainty	A state of knowledge – measured by probability – represented by aleatory and epistemic elements.

This page left intentionally blank.

2. Bayesian Inference

2.1 Introduction

In Chapter 1, we defined inference as the process of obtaining a conclusion based on the information available, which includes observed data as a subset. From this, we can define Bayesian inference.

Bayesian Inference	A process of inference using Bayes' Theorem in which information is used to newly infer the plausibility of a hypothesis. This process (Bayesian inference) produces information that adds to organizational knowledge.

Information about a hypothesis beyond the observable empirical data about that hypothesis is included in the inference. In this view, probability quantifies our state of knowledge and represents the plausibility of an event, where "plausibility" implies apparent validity. In other words, Bayesian inference is a mechanism to **encode** information, where the encoding metric is a value (on an absolute scale from 0 to 1) known as a **probability**.

Probability	A measure of the degree of plausibility of a hypothesis. Probability is evaluated on a 0 to 1 scale.

The types of hypotheses potentially considered as a part of risk analyses can be quite varied and include items such as: the ability of an astronaut to carry out an action, the temperature on the leading edge of a heat shield during Earth reentry, the likelihood of ground personnel incorrectly performing testing, the potential for redundant thrusters to fail simultaneously, the stress loading on near-by components during a hydrazine deflagration, and so on.

Key points related to Bayesian inference include:
- Bayesian inference produces **information**, specifically, probabilities related to a hypothesis. Bayesian Inference = Information, where Information = Models + Data + Other Information.
- Probability[1] is a **measure** of the degree of plausibility of a hypothesis. Probability is evaluated on a 0 to 1 scale.
- Unlike temperature though, probability – in the objective sense – **does not** exist (it is not measured, therefore it is never considered data).
- Since probability is subjective, for any hypothesis there is **no true** value for its associated probability. Furthermore, because model validity is described probabilistically, there is no such thing as a true, perfect, or correct model.

First used by Laplace, the Bayesian method of inference or inductive reasoning has existed since the late 1700s. [Bayes, 1763; Laplace, 1814]. Inductive reasoning relies on the fundamental concept of a probability as illustrated in Figure 2-1. In general, events described as part of a risk analysis represent necessary elements of a scenario-based model. These events are generally uncertain – we do not know exactly whether an event will occur (true) or not occur (false) and, therefore, events are described via probabilities. Probability in this context corresponds to the "absolute" scale described by Sträter [2003] and is simply measured on a zero-to-one cardinal scale.

---

[1] In this document, like in risk and reliability practice, we denote probabilities via different nomenclature, including f(x), π(x), P(x), and Pr(x).
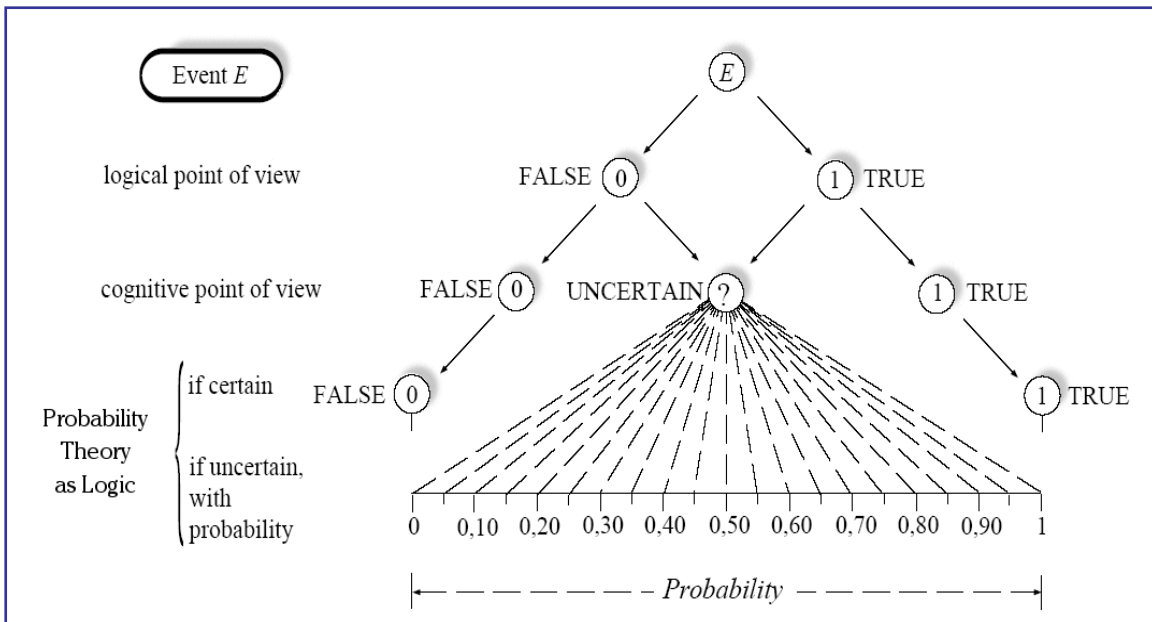
- Figure 2-1. Graphical abstraction of probability as a measure of information (adapted from "Probability and Measurement Uncertainty in Physics" by D'Agostini, [1995]).

The use of Bayesian inference methods, as noted by risk researchers (for example, Kelly [1998]), uses "all of the (available) information" and "leads to better parameter estimates and to better decisions." These aspects are important since organizations are often asked to make inference using sparse data. Consequently, waiting to obtain the "long run" objective frequency before making a decision is in many cases simply not possible. Furthermore, additional information (beyond data) usable in the Bayesian inference framework should not be ignored. For example, a great deal of engineering design and experience has gone into current NASA missions such as the Space Station and Space Shuttle – ignoring this bank of information based upon a desire to let the data "speak for themselves" is inefficient and misguided.

To describe the Bayesian method of inference (or simply the "Bayes" method), we first note that logic theory uses a variety of deductive and inductive methods [Jaynes; 2003]. Logic theory relies on two plausibility statements, the so-called "strong" and "weak" statements:

Strong: If **A** occurs, then **B** will occur   **B** is false   thus, **A** is false
Weak: If **A** occurs, then **B** will occur   **B** is true   thus, **A** is more plausible

where the observed evidence that we speak of is contained in event **B**.

In probabilistic failure models such as those used in PRA, we generally obtain information in the form of the weak logic statement. The observations we track, categorize, and analyze are failures – failures of hardware, software, and humans. In terms of the logic statements above, when we record failure events, we have observations where B is true. Examples of these observations include:

- If it is raining then the sidewalk is wet. The sidewalk is wet. Thus, it is more plausible that it is raining.
- If an engine gimble demand failure probability is 0.01, then the gimble will, on average, fail once in 100 starts. The gimble failed 3 times in the last 100 tests. Thus, it is more plausible that the failure probability is larger than 0.01.

The more failures we record, the more information we have related to the associated event **A**, which may represent hardware, software, or human performance. As described, it is the weak logic statement that is used in predicting performance, where this relationship between observable events and performance can be described mathematically by:

$$P(A \mid B) = P(A) \frac{P(B \mid A)}{P(B)}.$$ **(2-1)**

This equation of logical **inference** is known as Bayes' Theorem. If we dissect Equation (2-1), we will see there are four parts, as listed in Figure 2-2:

where   D =   the **D**ata

              H =   our **H**ypothesis

              E =   general information known prior to having updated information (or data) specific to problem at hand in other words, our **E**xperience).
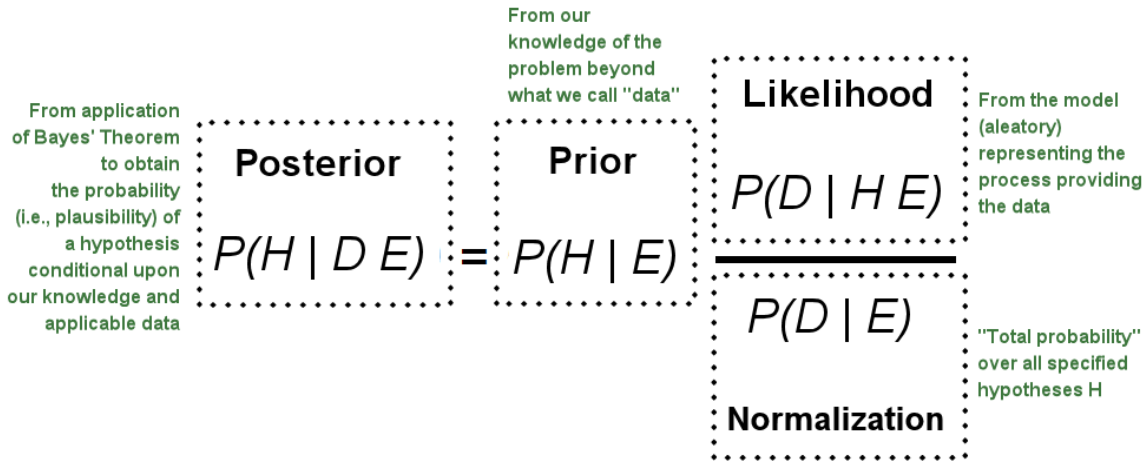
- Figure 2-2. The equation for Bayes' Theorem dissected into four parts.

In the context of PRA, where we use probability distributions to represent our state of knowledge regarding parameter values in the models, Bayes' Theorem gives the **posterior** (or updated) distribution for the parameter (or multiple parameters) of interest, in terms of the prior distribution, failure model, and the observed data, which in the general continuous form is written as:

$$\pi_1(\theta \mid x) = \frac{f(x \mid \theta)\pi(\theta)}{\int f(x \mid \theta)\pi(\theta)d\theta}.$$ **(2-2)**

In this equation, $\pi_1(\theta|x)$ is the posterior distribution for the parameter of interest, denoted as θ (note that θ can be vector-valued). The posterior distribution is the basis for all inferential statements about θ, and will also form the basis for model validation approaches to be discussed later. The observed data enters via the likelihood function, $f(x|\theta)$, and $\pi(\theta)$ is the prior distribution of θ.

The denominator of Bayes' Theorem is sometimes denoted f(x), and is called the marginal or unconditional distribution of x. The range of integration is over all possible values of θ. Note that it is a weighted average distribution, with the prior distribution for θ acting as the weighting function. In cases where X is a discrete random variable (e.g., number of events in some period of time), f(x) is the probability of seeing x events, unconditional upon a value of θ. In this context, which will become useful for model validation, f(x) will be referred to as the **predictive** distribution for X.

> **The likelihood function, or just likelihood, is also known by another name in PRA applications – it is the aleatory model describing an observed physical process. For example, a battery failure may be modeled in an electric power system fault tree as a Poisson process. Consequently, there is a fundamental modeling tie from the PRA to the data collection and evaluation process.**

The **likelihood function**, $f(x|\theta)$, is most often binomial, Poisson, or exponential in traditional PRA applications.[1] This function represents the observed failure generation mechanism (the aleatory failure model) and is also a function of the data collection process. Note that the symbol "|" represents a conditionality, which in the case of the likelihood function is described as "the probability we see the observed data **given** the parameter θ takes on a certain value."[2]

Priors can be classified broadly as either informative or noninformative. Informative priors, as the name suggests, contain substantive information about the possible values of the unknown parameter θ. Noninformative priors, on the other hand, are intended to let the data dominate the posterior distribution; thus, they contain little substantive information about the parameter of interest. Other terms for noninformative priors are diffuse priors, vague priors, flat priors, formal priors, and reference priors. They are often mathematically derived, and there are numerous types in use (for example, see Appendix B).

2.2 A simple example — Tossing a coin to collect data

If we toss a coin, can we tell if it is an unfair coin? Specifically, what can the Bayesian method do to assist in this inference task? The issue that we are concerned with is the possibility of an unfair coin (e.g., a two-headed coin; for now, we will ignore the possibility of a two-tailed coin only to simplify the presentation) being used. Let us jump directly into the Bayes analysis to see how straightforward this type of analysis can be in practice.

First, we note that Bayesian methods rely on three items, the prior, the likelihood, and data. The prior is a mechanism for encoding a state of knowledge about a hypothesis. In this example, we have (at a minimum) two hypotheses (**H**):

> **H1** = fair coin
> **H2** = unfair coin

Recall in this example, that an unfair coin implies a two-headed coin. Thus, the probability of heads associated with **H2** would be 1.0 (since we cannot obtain a tail if in fact we have two heads). At this point, we are ready to assign the prior.

Step 1: The prior
Knowledge of the person tossing the coin might lead us to believe there is a significant chance that an unfair coin will be used in the toss (perhaps this person has such a coin in their possession). Thus, for the sake of discussion, let us assume that we assign the following prior probabilities to the two hypotheses:

> P(**H1**)= 0.75　　　　　　P(**H2**) = 0.25

This means that we think there is a 25% chance that an unfair coin will be used for the next toss. Expressed another way, this prior belief corresponds to odds of 3:1 that the coin is fair.

Step 2: The data

---

[1] When dealing with repair or recovery times, the likelihood function may be lognormal, Weibull, or gamma. If the observed variable is unavailability, the likelihood may be a beta distribution. We will describe these, and other, examples in Chapter 4.
[2] When the observed random variable is discrete, the likelihood function is a probability. For continuous random variables, the likelihood is a density function, which is proportional to a probability.

The coin is tossed once, and it comes up heads.

Step 3: Bayesian calculation to estimate probability of a fair coin, $P(H_1)$

The likelihood function (or aleatory model) representing "random" outcomes (head/tail) for tossing a coin is given by the Bernoulli model:

$$P(D \mid H\,E) = p_i \tag{3}$$

where

p_i = P(getting a head on a single toss conditional upon the ith hypothesis)

The normalization constant in Bayes, $P(D \mid E)$, is found by summing the prior multiplied by the likelihood over all possible hypotheses, or in our case:

$$P(D \mid E) = P(H_1)p_1 + P(H_2)p_2 = 0.75(0.5) + 0.25(1.0) = 0.625 \tag{4}$$

where for hypothesis **H1**, $p_1 = 0.5$ while for **H2**, $p_2 = 1.0$. At this point, we know the prior, the likelihood (as a function of our one data point), and the normalization constant. Thus, we can compute the posterior probability for our two hypotheses. When we do that calculation, we find:

P(**H1** | one toss, data are "heads") = 0.6
P(**H2** | one toss, data are "heads") = 0.4

The results after one toss are presented in Table 1 and show that the posterior probability is the normalized product of the prior probability and the likelihood (e.g., H_1 posterior is 0.375/0.625 = 0.60).

• Table 1. Bayesian inference of one toss of a coin on an experiment to test for a fair coin.

Hypothesis	Prior Probability	Likelihood	(prior) x (likelihood)	Posterior Probability
H1: fair coin (i.e., the probability of a heads is 0.5)	0.75	0.5	0.375	0.60
H2: two-headed coin (i.e., the probability of a heads is 1.0)	0.25	1.0	0.250	0.40
	Sum: 1.00		Sum: 0.625	Sum: 1.00

What has happened in this case is that the probability of the second hypothesis (two-headed coin) being true has increased by almost a factor of two simply by tossing the coin once and getting a head. If it is possible to continue to collect data (which always has a beneficial impact in Bayesian methods), we can evaluate the data sequentially via Bayes' Theorem as it arrives. For example, let us assume that we toss the coin j times and want to make inference on the hypotheses (if a head comes up) each time. Thus, we toss (x = 1, 2, …, j) the coin again and again, and each time the estimate of the probability that the coin is fair changes. We see this probability plotted in Figure 2-3, where initially (before any tosses) the prior probability of a fair coin (H_1) was 0.75. However, after five tosses where a head appears each time, the probability that we have a fair coin is small, less than ten percent – a Bayesian would claim the odds are about 10:1 **against** having a fair coin.
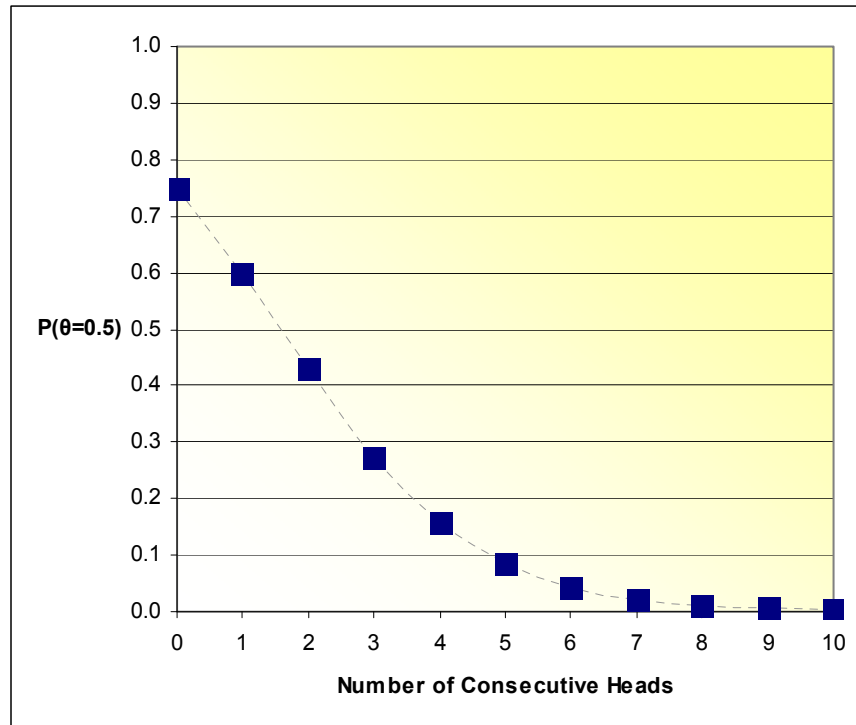
- Figure 2-3. Plot of the probability of a fair coin as a function of the number of tosses.

| Odds | The odds of an event is determined by taking the ratio of the event probability and one minus the event probability, or: Odds = Event Probability / (1 – Event Probability). |

At this point, one may be wondering what tossing a coin has to do with PRA or data analysis. The short answer to that query is that, just as in the simple coin-tossing example, Bayesian methods provide a structured and consistent way (logically and mathematically) to perform quantitative inference in situations where we have experience (i.e., prior information) and subsequent data. The same aleatory models (Bernoulli model for failures on demand, Poisson model for failures in time) that are used for coin-tossing and radioactive decay are applied to the basic events in traditional PRAs.

3. Data Collection Methods

3.1 Introduction

3.1.1 Key Topical Areas

This section provides an overview of the types of information and data of interest, the sources of these, and type classifications. The concept of a taxonomy specific to risk and reliability analysis is described. Lastly, an example case study is presented – this case study will be used in the next Chapter to demonstrate the inference methods.

3.1.1 Types of Information and Data

The focus of a data collection process is to inform future risk/reliability assessments, which themselves inform decision-making processes. As noted by Vesely, "the data analysis results serve as an input to the risk analysis…risk analysis, however, also affects the data collection and data analysis…" [ASME, 1977] The implication is that data "collection" and "analysis" are not performed in isolation – an understanding of the intended use and application of the process results should be present during the design and implementation of the analysis methods.

The analysis methods described in this document are applicable to a wide variety of inference situations, including those outside the risk and reliability community. Consequently, we could generalize the methods herein very broadly. However, the focus of this document is to demonstrate these methods as they pertain to risk and reliability practices. Within these boundaries, the types of information and data to be described are found in a relatively narrow set.

A general description of the type of information and data dealt with in this document is that we are looking at failure, success, or degradation for systems or components. As we will describe in the next Chapter, information or data that is vague, uncertain, or only somewhat applicable is still useful when inferring the performance of systems or components. We illustrate this concept of ancillary information, in a qualitative fashion, in Figure 3-1, where two key attributes, applicability and precision, of data and information are shown. Within this figure, the term "applicability" implies the degree of specificity between the set of information and the inference being made. For example, if we are inferring the failure probability of a pulsed plasma thruster, "complete applicability" information would need to be specific to that unique thruster type. Conversely, "minimal applicability" information could be failures for the general class of "thrusters."

In Figure 3-1, the term "precision" implies the amount of quantitative information available for the inference process. In the pulsed plasma thruster, an example of "high precision" information would be when the thruster has been designed, constructed, tested, and operated under real conditions, where this history is available for analysis. The converse, "low precision" information would be when no test or operational data are available for the thruster.

Data and Information

A: **Generic data for a class of components (e.g., electrical components)**

B: **Generic data for specific component type**

C: **Test data for a new system**

D: **Test data for a new component**

E: **Expert opinion on an existing component**

F: **Test and operational data for a similar component**

G: **Test and operational data for a well established component**

H: **Test and operational data for a dissimilar component**

• Figure 3-1. Qualitative listing of the types of data/information that are used for risk and reliability inference.

Generic data	Surrogate or non-specific information related to a class of parts, components, subsystems, or systems.

As new data are collected, analysts will incorporate these into existing data collection systems. These data will be processed as specific missions are underway and are completed. Over time, the set of data and information will be used to perform inference on a variety of classes of systems and components. This knowledge base will span a variety of needs, from inferring "high level" performance of complex systems to "low level" reliability of specific parts.

3.2 Mission Development Phase and Data Availability

Different types of data come about for a component/system at different times or phases. For example, information is known in the design phase that may provide some use even if the information is imprecise or only partially applicable. During a mission, data may be available on component/system performance, but, depending on the mission type, specific data on the nature of an incident may not be available.

A variety of process phases are followed for NASA activities. For example, the System Engineering Technical Processes (as defined by NPR 7123.1 Systems Engineering Processes and Requirements and the SP-2007-6105 Systems Engineering Handbook), list:
- Requirements Design
- Technical Solution Definition
- Technical Planning
- Technical Control
- Technical Assessment
- Technical Decision Analysis
- Product Transition

- Evaluation
- Design Realization

Some of these engineering phases are programmatic in nature and may provide (relatively) more information than data.

3.3 Sources of Data and Information

3.3.1 NASA and Aerospace-Related Data Sources

NASA has performed risk and reliability assessments for a variety of vehicles and missions for over 40 years. Each of these quantitative evaluations tends to increase the general collection of risk and reliability information when this information is stored or published for later use. In addition to the individual quantitative evaluations, NASA also manages incident reporting systems, for example via the Problem and Corrective Action (PRACA) system. PRACA systems have served as key information repositories and have been used in analyses such as the Shuttle PRA and the Galileo RTG risk assessment. A selection of other NASA data collection systems includes:

- Center-specific Problem Reporting systems (to record pre- and operational anomalies)
- The Spacecraft On-Orbit Anomaly Reporting system
- The Problem Report/Problem Failure Report (PR/PFR) system
- Incident, surprise, and anomaly reports
- PRA and reliability analysis archives (e.g., Shuttle, ISS)
- Apollo Mission Reports
- The Mars Exploration Rover Problem Tracking Database
- Results of expert elicitation

Within NASA, the treatment of intellectual property and data rights fall under the umbrella of the Office of General Counsel. In short, processes are in place to control issues such as data rights, publication of data, and the release of information outside of NASA. NASA, in general, does fall under the Freedom of Information Act (FOIA), which permits the right of access to agency records unless specifically exempted. Proprietary documents containing private sector trade secrets and commercial information, generated outside NASA, but in the control of NASA, are exempt from disclosure by FOIA.

Implications of proprietary data will manifest themselves for any computerized implementation of a data repository. The data repository may need to be constructed such that proprietary data are characterized as such so that only applicable users have access to this information. In this case, inference will – in many cases – draw upon both proprietary and non-proprietary sources. Data classified as International Traffic in Arms Regulations (ITAR) will restrict its availability to U.S. persons.

3.3.2 Other Data Sources

Outside the NASA and associated industries, a large set of risk and reliability data/information is collected. While many of these knowledge sources fall into the category of "generic" data, their applicability to NASA applications may be high in certain instances. Examples of these sources include:

- IEEE Std 500-1984
- NUCLARR (updated version is called NARIS)
- Nuclear industry EPIX/RADS system
- The Military Handbook for Reliability Prediction of Electronic Equipment MIL-HDBK-217F
- Government-Industry Data Exchange Program (GIDEP)
- Reliability Information Analysis Center (RIAC)
- Nonelectronic Parts Reliability Data, [Rome Air Development Center, 1995] NPRD-95
- International Common Cause Failure Data Exchange (ICDE)

The GIDEP is a cooperative activity between government and industry for the goal of sharing technical information essential during the life cycle of systems or components. As part of GIDEP, the "Reliability and Maintainability Data" contains information for methods for reliability and maintainability processes.

Other sources of data include non-U.S. experience such as launch vehicle performance (e.g., Ariane) and Russian experience (e.g., Soyuz, Proton). However, the availability of quality non-U.S. data is generally limited with a few exceptions (e.g., the OREDA Offshore REliability DAta).

3.4 Risk and Reliability Lexicon

The goal of inference is to enrich an organization's knowledge base. To successfully carry out this inference process within an organization, information must be transmitted between entities in the organization. Consequently, a common vocabulary, or lexicon, must be established to avoid increasing inference uncertainty due to miscommunication.

Failure databases use a hierarchical system (i.e., taxonomy) to create structure and logic between the listed items. One of the key items found when constructing any taxonomy is an identifiable boundary around the piece(s) of equipment to be tracked. A second key item in failure taxonomies is the text to describe the failure itself, or the so-called "failure mode." To depict these system-related constituents, we need to define the terms of interest. Definitions of the terms in our failure hierarchy are:

System	A group of subsystems and components organized as a whole. A system may have one or more functions.
Subsystem	A group of components comprised as a portion of a system.
Component	A group of parts designed to work together to perform a specific function. Components are constituents of systems and subsystems.
Part	The smallest piece of hardware contained in a component. One piece of hardware, or a group of pieces, that cannot be disassembled without destruction. Examples of a part are a bolt, resister, or electronic chip.
Failure	A loss of function for a system, subsystem, component, or part.
Failure Cause	A causal impact leading, either indirectly or directly, to failure.
Failure Mode	A specific type of failure, describing the manner of failure.
Failure Effect	An outcome resulting from a failure. A failure may have multiple effects.

As we will demonstrate in Section 4, inference may be performed at any level of this failure hierarchy.

3.5 Taxonomies and Classification

Using a taxonomy implies a knowledge structure used to describe a parent-child relationship (i.e., a hierarchy). Under the guidelines for evaluation of risk and reliability-related data, the taxonomy provides the structure by which data and information elements provide meaning to analysts. In this section, we demonstrate examples of taxonomies for possible use in risk and reliability inference processes. The intent of this section is to describe ways in which taxonomies are used to classify structures related to data collection. The examples listed in this section should not be construed as definitive taxonomies for failure classification – that exercise must be accomplished during implementation of a specific data collection system.

Within the risk and reliability community, a variety of taxonomies and associated definitions are used. We will first explore a variety of these to provide a sense that, depending on the application, some variation in the structure details is possible.

In a recent paper, Tumer, Stone, and Bell acknowledge that a variety of potential failures can be detected at the conceptual design phase. [Tumer Stone Bell, 2003] As part of this work, they attempted to derive what they called "elemental" failure modes based upon physical properties, with the goal to place these properties into a failure taxonomy. Relying on failure modes and effects analysis and earlier classification work, they developed a loss-of-function failure taxonomy that included a first level of:

Corrosion	Wear	Impact	Fretting	Creep	Fatigue
Thermal	Galling	Spalling	Radiation	Buckling	Rupture

If one were concerned about the physical causes of failures, a set of physics-based causal factors such as these would be required. However, this low level of information is not necessary if the inference being made for a specific component or system is concerned with – in general – failures or successes. If, instead, we wished to infer the probability of failure conditional upon a specific failure mechanism, we would need to have information related to the nature of failure. Birolini, in his reliability text, suggests that failures should be classified by mode, cause, and effect [Birolini, 2004].

In other words, this classification can take place via a failure modes and effects analysis, similar to the functional failure modes and effects analysis as described in MIL-STD-1629. [MIL-STD-1629, 1984] Henley and Kumamoto carried this idea one step further when they proposed a formal cause-consequences structure to be stored in an electronic database. [Henley and Kumamoto, 1985] In their approach, specific keywords, called modifiers, would be assigned to equipment failures. For example, modifiers for on-off operation included: close, open, on, off, stop, restart, push, pull, and switch.

The NASA PRA Procedures Guide [NASA, 2002] describes an event classification process as:

```
System
 └ Component
   └ Failure Mode
     └ Affected Item
       └ Failure Mechanism
         └ Failure Cause
```

The lowest level in this hierarchy, failure cause, would provide a bridge back to the Tumer, Stone, and Bell loss-of-function failure taxonomy.

In 2006, the NASA Engineering and Safety Center Working Group led an activity to produce a common taxonomy for structuring nonconformances, anomalies, and problems. [NASA, 2006b] While not a failure taxonomy, Appendix A of that report provides a hierarchical identification of the problematic item.

This structure looks like:

```
Item
 └ System
   └ Subsystem
     └ Assembly
       └ Component
Problem Type
 └ Catastrophic failure
 └ Failure to meet primary objective(s)
 └ etc.
Mission Type
 └ Crewed (human)
 └ Uncrewed (robotic)
 └ etc.
Vehicle/Spacecraft Type
 └ Shuttle
 └ International Space Station
 └ etc.
Failure Mode
Lifecycle phase
 └ Manufacture
 └ Assembly and integration
 └ etc.
```

If we evaluate complex systems within specific missions, detailed types of taxonomies down to the component level may be found, for example, from
http://nasataxonomy.jpl.nasa.gov/nasinst/index_tt.htm :

```
Instruments
. Accelerometers
. Acoustic Sensors
. Anemometers
. Antennae
. Barometers
. Beta Detectors
. Cameras
. . CCD Cameras
. . Framing Cameras
. . Imaging Cameras
. . Vidicon Camera
. Charged Particle Analyzers
. Cosmic Dust Analyzers
. Dosimeters
```

Lastly, in NPR 7120.5D "NASA Space Flight Program and Project Management Requirements," a hierarchy related to programs and their life cycles was defined. [NASA, 2007] This hierarchy down to the project level appears as:

Mission Directorate
  └ Program
    └ Project

where

Program   a strategic investment by a Mission Directorate or Mission Support Office that has a defined architecture, and/or technical approach, requirements, funding level, and a management structure that initiates and directs one or more projects. A program defines a strategic direction that the Agency has identified as needed to implement Agency goals and objectives.

Project   a specific investment identified in a Program Plan having defined requirements, a life-cycle cost, a beginning, and an end. A project also has a management structure and may have interfaces to other projects, agencies, and international partners. A project yields new or revised products that directly address NASA's strategic needs.

Also, within this NPR, the program life cycle is given by two general phases:
- Formulation – The technical approach is derived and initial project planning is performed.
- Implementation – The program acquisition in which the approval, implementation, integration, operation, and ultimate decommissioning are performed.

Outside of NASA, a new standard, ISO 14224, focused on the collection and processing of equipment failure data has been produced. Other guidance on data collection taxonomies may be found from the following sources:
- ISO 6527:1982 Nuclear power plants -- Reliability data exchange -- General guidelines
- ISO 7385:1983 Nuclear power plants -- Guidelines to ensure quality of collected data on reliability
- ISO 14224:2006 Petroleum, petrochemical and natural gas industries -- Collection and exchange of reliability and maintenance data for equipment

3.6 Case Study System Description

Future (hypothetical) spacecraft most likely will employ a crew module containing a variety of systems including avionics, thermal control, power, environmental control, and so on. To demonstrate the methods applications described in Section 4 of this document, one of these systems – active thermal control – will be used as an expanded case study.

3.6.1 Active Thermal Control System (ATCS)

The ATCS reference architecture for this study is shown in Figure 3-2. The system criteria and assumptions for the ATCS are given below.

The ATCS provides the capability to remove heat from various components within the crew module. This system is a two-string system with a primary loop (Loop A) and a secondary (standby) loop (Loop B). The major components of the primary loop include:

- Refrigerant tank (accumulator)
- Coolant circulation pump
- Radiators (two in parallel for each loop)
- Mixing valve
- Level sensor
- Various check and manual (isolation) valves

Electric power to the pump and sensor is provided by a separate system (the electric power system). The pump control signals come from the avionics system, which monitors the system via a variety of temperature and coolant level sensors.

Two radiators, which are common to each loop, provide heat rejection. Heat rejection is assumed to be successful if one of the radiators is functional.

The system is failed if coolant is not supplied to at least one of the radiators in either Loop A or Loop B. Loop A operates continuously, while Loop B provides backup capability but is normally in a quiescent condition. Switchover to Loop B is controlled by crew personnel.

The level sensor has on-board software that controls its functionality. This sensor also interacts with the avionics system and is powered by the DC power system. Crew personnel interact with the avionics system in order to set operational parameters for the ATCS.

Other parts of the system include the relevant piping and electrical cabling.
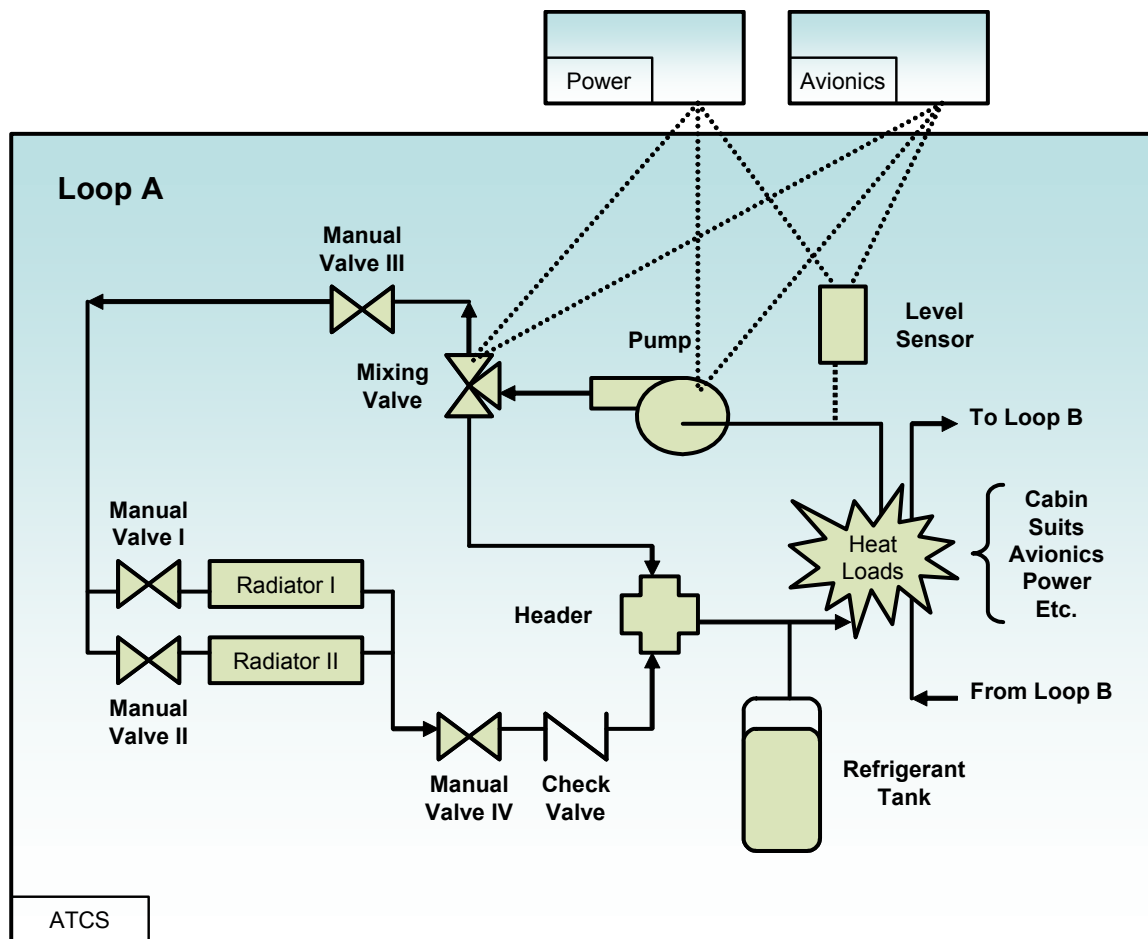
• Figure 3-2. Active thermal control system diagram.

3.6.2 ATCS Component Taxonomy

Mission Directorate: **Exploration Systems Mission Directorate**
  └ Program: **IntraSolar Exploration**
    └ Project: **New Manned Vehicle (NMV)**
      └ Subsystem: **Crew Module**
        └ Subsystem: **Active Thermal Control**

Under the ATCS subsystem, we have identified two additional subsystems:

  └ Subsystem: **Loop A**
  └ Subsystem: **Loop B**

Under each of these subsystems, a variety of components and parts exist. For example, within the Loop A subsystem, we have:

Component: Refrigerant tank
Component: Circulation pump
Component: Radiator I
Component: Radiator II
Component: Mixing valve
Component: Level sensor
Component: Manual valve I
Component: Manual valve II
Component: Manual valve III
Component: Manual valve IV
Component: Check valve
Component: Header
Component: System piping
Component: Power cabling
Component: Control cabling

4. Parameter Estimation

4.1 Preface to Chapter 4

This chapter is not written in the typical style of other data analysis guides and textbooks, in the sense that it contains few equations and references to the technical literature. Instead WinBUGS scripts are used in place of equations (for assistance in the "how to" of running these scripts, see Appendix C) in the hope that they will convey essential concepts to a more practically-oriented audience. OpenBUGS, commonly referred to as WinBUGS, is an open-source Markov chain Monte Carlo (MCMC) based Bayesian updating software that will run on Microsoft Windows™, Linux, and Mac OS X™ (using the Wine emulator) platforms. Furthermore, these scripts can be adapted to problems at hand by analysts in the various NASA Centers. To complement the MCMC approach, we provide a mathematical basis of Sections 4.2.1, 4.2.2, and 4.2.3 in Appendix D. Additional technical bases and mathematical details behind the remaining topics treated in Chapter 4 will be provided later in a supplemental report.

For most of the example problems, the document uses the MCMC approach. MCMC methods work for simple cases, but more importantly, they work efficiently on very complex cases. Bayesian inference tends to become computationally intensive when the analysis involves multiple parameters and correspondingly high-dimensional integration. As noted by NUREG/CR-6823 (the *Handbook of Parameter Estimation for Probabilistic Risk Assessment*):

> "A practical consequence of the high dimension of [the parameter of interest] is that...numerical integration and simple random sampling methods do not work well. More recently developed methods, versions of MCMC, must be used."

MCMC methods were described in the early 1950s in research into Monte Carlo sampling at Los Alamos. Recently, with the advance of computing power and improved analysis algorithms, MCMC is increasingly being used for a variety of Bayesian inference problems. MCMC is effectively (although not literally) numerical (Monte Carlo) integration by way of Markov chains. Inference is performed by sampling from a target distribution (i.e., a specially constructed Markov chain, based upon the inference problem) until convergence (to the posterior distribution) is achieved.

The approach that is taken in the document is to provide analysis "building blocks" that can be modified, combined, or used as-is to solve a variety of challenging problems. The MCMC approach used in the document is implemented via textual scripts similar to a macro-type programming language. Accompanying each script is a graphical diagram illustrating the elements of the script and the overall inference problem being solved. In a production environment, analysis could take place by running a script (with modifications keyed to the problem-specific information). Alternatively, other implementation approaches could include: (1) using an interface-driven front-end to automate an applicable script, (2) encapsulating an applicable script into a spreadsheet function call, or (3) creating an automated script-based inference engine as part of an information management system.

To assist the analyst in maneuvering through Section 4.2 of this chapter, we describe key elements of the single parameter estimation process by way of the flow diagram shown in Figure 4-1.
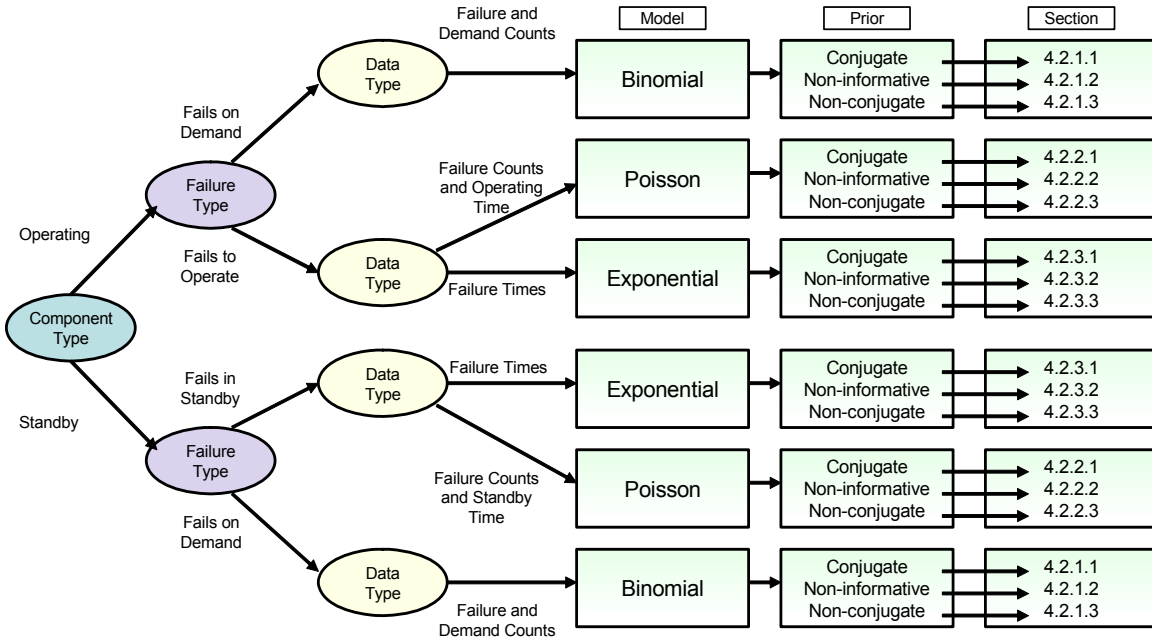
- Figure 4-1. Flow diagram to guide the selection of analysis methods for parameter estimation in Section 4.2.

In this document, WinBUGS scripts are indicated by way of formatted text inside a shaded box, such as:

```
model   { x ~ dbin(p, n)   # Binomial dist for number of failures in n demands… }
```

Note that comments, inside the script, are preceded with the "**#**" character. Additional detail on the parts of WinBUGS scripts and how to run the scripts are provided in Appendix C.

4.2 Inference for Common Aleatory Models

We begin with the most commonly encountered situations in PRA, which meet the following three assumptions:

- ☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) contains a single unknown parameter.
- ☑ The prior information is homogeneous and is known with certainty[1].
- ☑ The observed data are homogeneous and are known with certainty.

More complicated inference, where one or more of these conditions are not satisfied, is described in later sections of this chapter. We treat the three most common **aleatory models** in this section:

- Binomial distribution for failures on demand
- Poisson distribution for initiating events and failures in time
- Exponential distribution for random durations, such as time to failure or time to recover

Lastly, we close this section with guidance on selecting prior distributions for single-parameter problems.

---

[1] For discussion and examples of uncertain priors and data, see Section 4.8.

Homogeneous	A set of information made up of similar constituents. A homogeneous population is one in which each item is of the same type.

4.2.1 Binomial Distribution for Failures on Demand

This model is often used when a component must change state in response to a demand. For example, a relief valve may need to open to relieve pressure upon receipt of a signal from a controller that an over-pressure condition exists. The following assumptions underlie the binomial distribution:

- ☑ There are two possible outcomes of each demand, typically denoted as success and failure.
- ☑ There is a constant probability of failure (in PRA, this is typically success in reliability engineering) on each demand, denoted herein as p.
- ☑ The outcomes of earlier demands do not influence the outcomes of later demands (i.e., the order of failures/successes is irrelevant).

The unknown parameter in this model is p, and the observed data are the number of failures, denoted by x, in a specified number of demands, denoted as n. Both x and n are assumed to be known with **certainty** in this section. Cases in which x and n are uncertain are treated in a later section.

Note that the binomial distribution describes the aleatory uncertainty in the number of failures, x. The Bayesian inference describes how the epistemic uncertainty in p changes from the prior distribution, which describes the analyst's state of knowledge about possible values of p before data are collected, to the posterior distribution, which reflects how the data have altered the analyst's state of knowledge.

4.2.1.1 Binomial Inference with Conjugate Prior—The simplest type of prior distribution from the standpoint of the mathematics[1] of Bayesian inference is the so-called **conjugate** prior, in which the prior and posterior distribution are of the same functional type (e.g., beta, gamma), and the integration in Bayes' Theorem is circumvented. Not every aleatory model will have an associated conjugate prior, but the four most commonly used models do. **For the binomial distribution, the conjugate prior is a beta distribution.**

Two parameters are needed to describe the beta prior distribution completely, and these are denoted $alpha_{prior}$ and $beta_{prior}$. Conceptually, $alpha_{prior}$ can be thought of as the number of failures contained in the prior distribution, and the sum of $alpha_{prior}$ and $beta_{prior}$ is like the number of demands over which these failures occurred. Thus, small values of $alpha_{prior}$ and $beta_{prior}$ correspond to less information, and this translates into a broader, more diffuse prior distribution.

With the **data consisting of x failures in n demands**, the conjugate nature of the prior distribution and likelihood function allows the posterior distribution to be determined using arithmetic. The posterior distribution is also a beta distribution, with newly adjusted (labeled "post") parameters given by:

$$alpha_{post} = alpha_{prior} + x$$
$$beta_{post} = beta_{prior} + n - x.$$

From the properties of the beta distribution (see Appendix B), the prior and posterior mean of p are given by:

$$Prior\ mean = alpha_{prior}/(alpha_{prior} + beta_{prior})$$
$$Posterior\ mean = alpha_{post}/(alpha_{post} + beta_{post})$$

Credible intervals (e.g., see Figure 4-2) for either the prior or the posterior can be found using the BETAINV() function built into modern spreadsheets.

---

[1] Discussion of the mathematical details may be found in Appendix D.

Credible Interval	Bayesian inference produces a probability distribution. The "credible interval" consists of the values at a set (one low, one high) of specified percentiles from the resultant distribution. For example, a 90% credible interval ranges from the value of the 5th percentile to the value of the 95th percentile.
Percentile	A percentile, p, is a specific value x such that approximately p% of the uncertainty is lower than x and (100-p)% of the uncertainty is larger than x. Common percentiles used in PRA include the lower-bound (5th percentile), the median (50th percentile), and upper-bound (95th percentile).

In summary, for conjugate distributions (e.g., beta prior when using a binomial aleatory model), we can solve the Bayesian inference problem by:

1. Knowing that the posterior distribution is the same type, but with "updated" parameters, as the prior.
2. Numerically integrating Equation 2-2 (via tools like Mathematica or Maple) with the applicable prior distribution and aleatory model. Note that when using a conjugate prior, numerical integration is not needed since the posterior can be found directly (using the equations above), but numerical integration is a general method for Bayesian inference.
3. Numerically simulating Equation 2-2 using MCMC (via tools like OpenBUGS) with the applicable prior distribution and aleatory model. Note that when using a conjugate prior, numerical simulation is not needed since the posterior can be found directly, but numerical simulation is a general method for Bayesian inference.
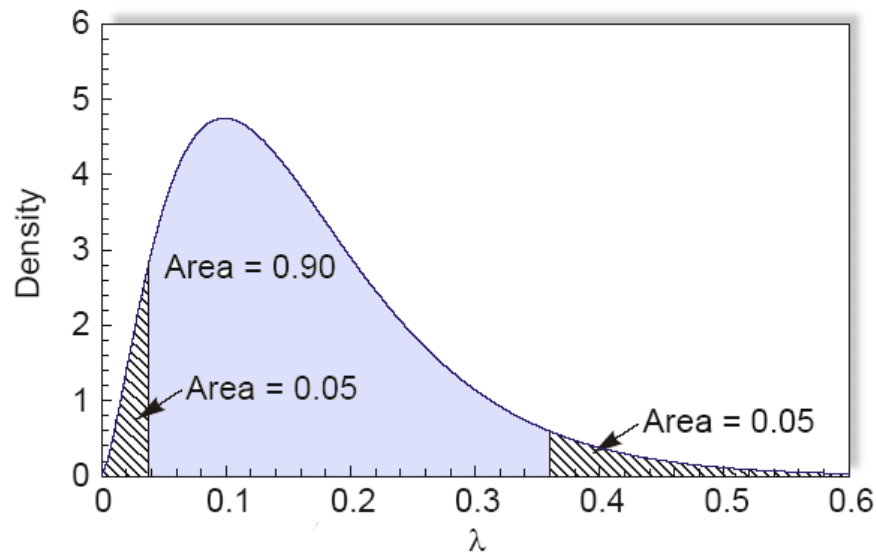
- Figure 4-2. Representation of a probability distribution (epistemic uncertainty), where the 90% credible interval (0.04 to 0.36) is shown.

We demonstrated (above) method #1 for the beta/binomial case by noting that the posterior is a beta distribution with parameters $alpha_{post} = alpha_{prior} + x$ and $beta_{post} = beta_{prior} + n - x$. Note that properties of the beta distribution may be found in Appendix B.

- Example 1. Relief valve fails to open (binomial model) and beta prior.

> The prior distribution for failure of a relief valve to open on demand is given (from an industry database) as a beta distribution with
> - $alpha_{prior}$ = 1.24
> - $beta_{prior}$ = 189,075.
>
> Assume two failures to open have been seen in 285 demands. Find the posterior mean of p, the probability that the valve fails to open on demand and a 90% credible interval for p.

Solution

We begin by noting that the mean of the beta prior distribution is 1.24/(1.24 + 189,075) = 6.56×10^{-6}. Because $alpha_{prior}$ is relatively small, the prior distribution expresses significant epistemic uncertainty about the value of p. This can be quantified by calculating a 90% credible interval for p based on the prior distribution. We use the BETAINV() function to do this. The 5th percentile of the prior distribution is given by BETAINV(0.05, 1.24, 189075) = 5.4×10^{-7} and the 95th percentile is given by BETAINV(0.95, 1.24, 189075) = 1.8×10^{-5}, a spread of almost two orders of magnitude.

With two failures to open in 285 demands of the valve, and the assumption that these failures are described adequately by a binomial distribution, the posterior distribution is also a beta distribution, with parameters $alpha_{post}$ = 1.24 + 2 = 3.24 and $beta_{post}$ = 189,075 + 285 – 2 = 189,226. The posterior mean of p is given by 3.24/(3.24 + 189,226) = 1.7×10^{-5}. The 90% posterior credible interval is found using the BETAINV() function, just as was done for the prior interval above. The posterior 5th percentile is given by BETAINV(0.05, 3.24, 189226) = 5.0×10^{-6} and the 95th percentile is given by BETAINV(0.95, 3.24, 189226) = 3.5×10^{-5}. Note how the epistemic uncertainty in the prior distribution has been reduced by the observed data. This is shown graphically in Figure 4-3, which overlays the prior and posterior distribution for this example.
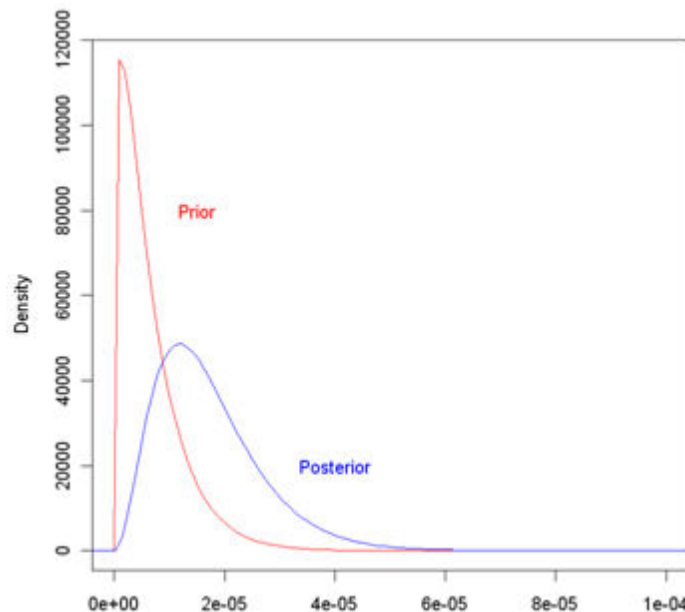
- Figure 4-3. Comparison of prior and posterior distributions for Example 1.

> **In many problems, fewer iterations would suffice. However, it is prudent to take more samples than the absolute minimum, so for problems involving one parameter, we recommend 1,000 burn-in iterations, followed by 100,000 iterations to estimate parameter values.**

Inference for conjugate cases can also be carried out using MCMC approaches (such as with OpenBUGS). Script 1 implements the binomial/beta conjugate analysis. For problems such as this, where there is only one unknown parameter to be estimated, the analyst should use 100,000 iterations, discarding the first 1,000 to allow for convergence to the posterior distribution. **Monitoring node p** will display the desired posterior results.

Within Script 1 (and the remaining scripts), the following notation is used:
- "**~**" indicates that the variable to the left of "**~**" is distributed as the *distribution* on the right of "**~**." Examples of distributions include binomial (dbin), beta (dbeta), gamma (dgamma), Poisson (dpois), normal (dnorm), and lognormal (dlnorm).
- "**#**" indicates that the text to the right of "**#**" is a comment.
- "**<-**" indicates that the variable to the left of "**<-**" is equivalent to the expression on the right of "**<-**."

```
model  {                        # A Model is defined between { } symbols
x ~ dbin(p, n)                  # Binomial dist. for number of failures in n demands
p ~ dbeta(alpha.prior, beta.prior)        # Conjugate beta prior distribution for p
}


data
list(x=2, n =285)                         # Data for Example 1
list(alpha.prior=1.24, beta.prior=189075)        # Prior parameters for Example 1
```

• Script 1. WinBUGS script for Bayesian inference with binomial likelihood and beta conjugate prior.

A directed acyclic graph (DAG) is a common way of displaying a Bayesian inference problem and is the underlying model used by WinBUGS (in script form). In a DAG, observed aleatory variables (such as x for the binomial inference problem above) are displayed as ovals that contain no children nodes (i.e., the "lowest" level in the diagram). Uncertain variables that influence x are shown at a higher level in the DAG, and are connected by arrows to the variables they influence (i.e., they are parents of the node they influence). Constant parameters (such as n above) are also shown in the DAG as diamonds. We will display the DAG associated with each WinBUGS script in this document; however, it is not necessary to develop the DAG, as WinBUGS uses the script representation for its analysis.

For relatively simple problems, a DAG can be an aid in understanding the problem, particularly for an analyst who is new to WinBUGS. However, as the complexity of the problem increases, most analysts will find that the script representation of the problem is clearer. We will use the following conventions for DAGs in this document. Note that all variables, which are referred to as *nodes* by WinBUGS, can be scalars, vectors, matrices, or arrays.
- Ovals represent stochastic variables whose uncertainty (either aleatory or epistemic) is represented by a probability distribution.
- Diamonds represent constant parameters (no uncertainty).
- Rectangles represent calculated parameters. As such, their probability distribution is not specified by the analyst but is calculated by WinBUGS from an equation within the script.[1]

---

[1] Note that within WinBUGS, rectangles always represent constant (nonstochastic) nodes. We have deviated from this convention to more clearly indicate which variables, stochastic or constant, are being calculated inside a WinBUGS script.

- Dashed lines are sometimes used for clarification when certain parameters are entered or calculated in the script as part of other nodes.
    - o In cases where the node is used as inference, the arrow will be connected to the dashed symbol.
    - o In cases where the parameter within the node is used as inference, the arrow will be connected to the symbol within the dashed node.

Figure 4-4 shows the DAG corresponding to the WinBUGS Script 1. This DAG illustrates that **x** is the observed variable, because it is a node with no children node. This node (x) is an uncertain variable, indicated by its **oval** shape. Its value is influenced by **p** (p is a parent node to x), which is the parameter of interest in this problem; we observe x (with n specified), and use this information to infer possible values for p. The dashed region at the top of the DAG, labeled "Beta Prior," clarifies the type of **prior** distribution used for p, and indicates that the parameters of this distribution (alpha and beta) are entered by the analyst.
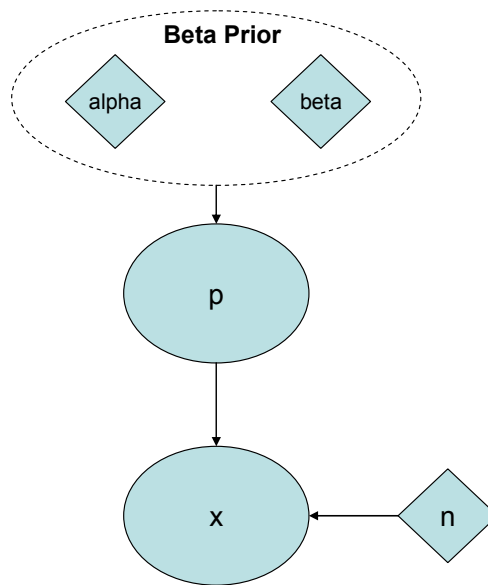
- Figure 4-4. DAG representing Script 1.

4.2.1.2 Binomial Inference with Noninformative Prior—As the name suggests, a **noninformative** prior distribution contains little information about the parameter of interest, which in this case is p. Such priors originated in a (continuing) quest to find a mathematical representation of complete uncertainty. This has led some to conclude that they should be used when one knows nothing about the parameter being estimated. As discussed in earlier sections, this is almost never the case in practice, and use of a noninformative prior in such a case can lead to excessively conservative results. Therefore, there are two situations in which a noninformative prior may be useful:

1. The first is where the observed data are abundant enough to dominate the information contained in any reasonable prior, so it does not make sense to expend resources developing an informative prior distribution.
2. The second is where the analyst wishes to use a prior that has little influence on the posterior, perhaps as a point of reference.[1]

---

[1] Note that the numerical results of a Bayesian analysis with noninformative priors will typically be similar (in numerical values) to a frequentist analysis.

> **With lots of observed data, the prior will have little influence on the posterior. So why not just use the data alone? Remember that a probability distribution is required to propagate uncertainty, so Bayes' Theorem is still used to obtain a posterior distribution.**

Noninformative priors are also known as formal priors, reference priors, diffuse priors, and vague priors. For clarity, we will refer to them as noninformative priors in this document, as that is the name most commonly employed in PRA. Unfortunately, there are many routes that lead to slightly different noninformative priors, and the intuitive choice of a uniform prior is not what is usually used in PRA. The most common noninformative prior for single-parameter inference in PRA is the **Jeffreys** prior (see Appendix B.7.8 for the Jeffreys prior for the binomial model).

The Jeffreys functional form is dependent upon the likelihood function, so there is not a single "Jeffreys prior" for all cases. Instead, there is a different Jeffreys prior for each likelihood function. For the case here, where the likelihood function is the binomial distribution, the Jeffreys prior is a beta distribution with both parameters equal to 0.5. Thus, inference with the Jeffreys prior is a special case of inference with a beta conjugate prior. Consequently, all of the results described in Section 4.2.1.1 apply in the case of the Jeffreys prior, but with alpha_{prior} and beta_{prior} both equal to a value of 0.5. Using the Jeffreys prior with the binomial model leads to a posterior mean of $(x + 0.5) / (n + 1)$.

> **Caution:**
> **When using the "zero-zero" non-informative prior (a beta distribution with both parameters set to zero) as an alternative to the Jeffreys prior, the number of failures *x* must be one or more (or the posterior will not be a proper distribution).**

Note that if x and n are small (sparse data), then adding "half a failure" to x may give a result that is felt to be too conservative. In such cases, a possible alternative to the Jeffreys prior is like a beta distribution with both parameters equal to zero (the "zero-zero" beta distribution). This is not a proper probability distribution, but as long as x and n are greater than zero, the posterior distribution will be proper and the posterior mean will be x / n. Conceptually, adjusting the beta prior so that alpha_{prior} and beta_{prior} both have small values (in the limit, zero) tends to reduce the impact of the prior and allows the data to dominate the results. Note, though, that when alpha_{prior} and beta_{prior} are equal, the mean of this beta prior is 0.5. The prior should reflect what information, if any, is known independent of the data.

4.2.1.3 Binomial Inference with Nonconjugate Prior—A **nonconjugate** prior is one in which the prior and posterior distribution are not of the same functional form. In such cases, numerical integration is required for the denominator of Bayes' Theorem. In the past, this has been a limitation of Bayesian inference, and is one reason for the popularity of conjugate priors. However, cases often arise in which a nonconjugate prior is desirable, despite the increased mathematical difficulty. As an example, generic databases often express epistemic uncertainty in terms of a lognormal distribution, which is not conjugate with the binomial likelihood function. In this section, we describe how to carry out inference with a lognormal prior, which is a commonly-encountered nonconjugate prior, and with a logistic-normal prior, which is similar to a lognormal prior but is more appropriate when the values of p are expected to be nearer one.[1]

Although spreadsheets can be used to carry out the required numerical integration for the case of a single unknown parameter, another way to deal with nonconjugate priors is with WinBUGS. We illustrate the case of a lognormal prior with the following example.

---

[1] Note that in a *reliability* analysis, as opposed to a PRA, the unknown parameter of interest might be $q = 1 - p$, in which case q would be close to one for many components.

- Example 2. Relief valve fails to open (binomial model) and lognormal prior.

  Continuing with the relief valve from Example 1, assume that instead of the conjugate prior in that example, we are using a generic database that provides a lognormal prior for p.

  Assume the generic database lists the mean failure probability as 10^{-6} with an error factor of 10. As in Example 1, assume that our observed data are two failures in 285 demands. The WinBUGS script shown below is used to analyze this example.

```
model  {
x ~ dbin(p, n)                         # Binomial model for number of failures
p ~ dlnorm(mu, tau)                    # Lognormal prior distribution for p
tau <- 1/pow(log(prior.EF)/1.645, 2)   # Calculate tau from lognormal error factor
# Calculate mu from lognormal prior mean and error factor
mu <- log(prior.mean) - pow(log(prior.EF) / 1.645, 2) / 2
}
data
list(x=2, n=285, prior.mean=1.E-6, prior.EF=10)
```

- Script 2. WinBUGS script for Bayesian inference with binomial likelihood function and lognormal prior.
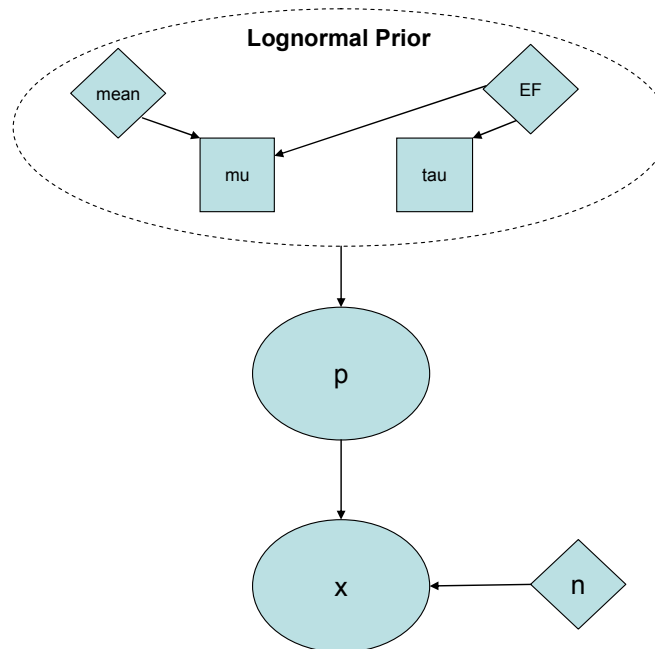
- Figure 4-5. DAG representing Script 2.

Solution

Running this script for 100,000 iterations, discarding the first 1,000 iterations to allow for convergence to the posterior distribution, gives a posterior mean for p of 4.7×10^{-5} and a 90% credible interval of (1.9×10^{-6}, 1.8×10^{-4}). Note that when the prior distribution is not conjugate, the posterior distribution cannot be written down in closed form. In such cases, an analyst may replace the numerically defined posterior with a distribution of a particular functional form (e.g., lognormal), or may use the empirical results of the WinBUGS analysis to construct a histogram.

> **When working with a lognormal prior distribution from a generic database, be sure to know whether the distribution is expressed in terms of the prior mean or median value.**

Generic databases may not always describe the lognormal distribution in terms of a mean value and an error factor; quite often the median (50th percentile) is specified rather than the mean value. This may also be the case when eliciting information from experts as an expert may be more comfortable providing a median value. In this case, the analysis changes only slightly. In Script 2, the line that calculates mu from the lognormal prior mean and error factor is replaced by the following line:

```
mu <- log(prior.median)
```

and prior.median is loaded in the data statement instead of prior.mean.

> **Beware of lognormal priors for p when p is expected to be near one since the tail of a lognormal distribution may exceed a value of 1.0.**

Cases may arise where the value of p could be approaching unity. In such cases, using a lognormal prior is problematic because it allows for values of p greater than unity, which is not meaningful since p is a probability (either failure probability or reliability, depending on the context). In such cases, a logistic-normal prior is a "lognormal-like" distribution, but one that constrains the values of p to lie between zero and one. The WinBUGS Script 3 (and associated DAG shown in Figure 4-6) uses the lognormal mean and error factor (e.g., from a generic database), but "constrains" the distribution to lie between zero and one by replacing the lognormal distribution with a logistic-normal distribution.

```
model  {
x ~ dbin(p, n)                         # Binomial distribution for number of failures
p <- exp(p.constr)/(1 + exp(p.constr)) # Logistic-normal prior distribution for p
p.constr ~ dnorm(mu, tau)
tau <- 1/pow(log(prior.EF)/1.645, 2)   # Calculate tau from lognormal error factor

# Calculate mu from lognormal prior mean and error factor
mu <- log(prior.mean) - pow(log(prior.EF)/1.645, 2)/2
}

data
list(x=2,n=256, prior.mean=1.E-6, prior.EF=10)
```

• Script 3. WinBUGS script for Bayesian inference with binomial likelihood function and logistic-normal prior.
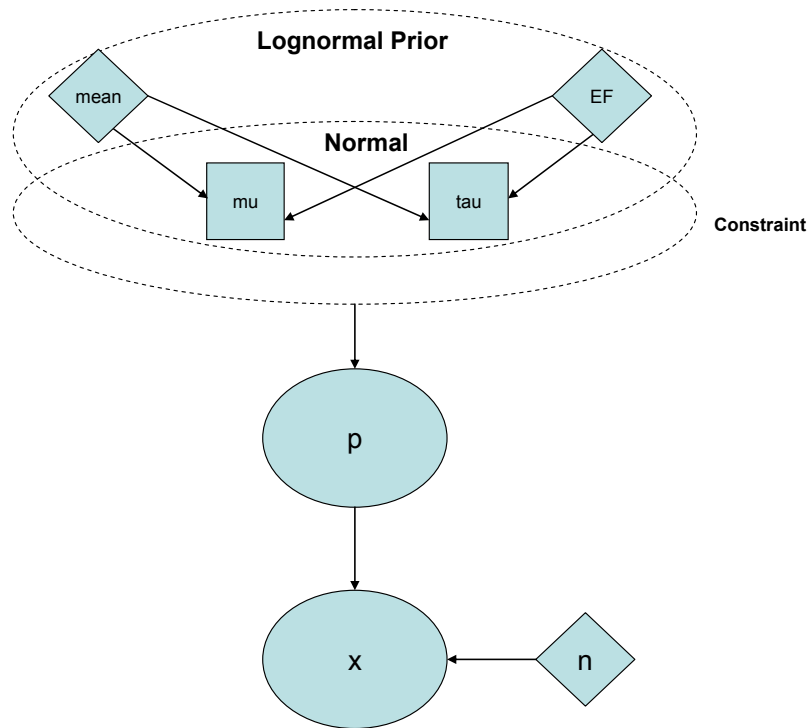
• Figure 4-6. DAG representing Script 3.

4.2.2 Poisson Distribution for Initiating Events or Failures in Time

The Poisson model is often used for failures of normally operating components, failures of standby components that occur at some point in time prior to a demand for the component to change state, and for initiating events. In the ATCS example, we might use a Poisson distribution as our aleatory model for failures of the operating circulation pump. The following assumptions underlie the Poisson distribution:

    ☑   The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval. The constant of proportionality is denoted by lambda.

    ☑   The probability of simultaneous events in a short interval of time is approximately zero.

    ☑   The occurrence of an event in one time interval does not affect the probability of occurrence in another, non-overlapping time interval.

> **Note that lambda is a rate and has units of inverse time.**
>
> **Also note that lambda is *not* a function of time, so the simple Poisson distribution cannot be used for reliability growth or aging.**

The unknown parameter in this model is lambda, and the observed data are the number of events, denoted x, in a specified time period, denoted t. Both x and t are assumed to be known with certainty in this section. Cases in which x and t also have epistemic uncertainty are treated in a later section. Note that the Poisson distribution describes the aleatory uncertainty in the number of failures, x. The Bayesian inference describes how the epistemic uncertainty in lambda changes from the prior distribution, which describes the analyst's state of knowledge about possible values of lambda before empirical data are collected, to the posterior distribution, which reflects how the observed data have altered the analyst's prior state of knowledge.

4.2.2.1 Poisson Inference with Conjugate Prior—As was the case with the binomial distribution, a conjugate prior is sometimes chosen for purposes of mathematical convenience. For the Poisson distribution, the conjugate prior is a gamma distribution. Two parameters are needed to describe the gamma prior distribution completely, and these are denoted $alpha_{prior}$ and $beta_{prior}$. Conceptually, $alpha_{prior}$ can be thought of as the number of events contained in the prior distribution, and $beta_{prior}$ is like the period of time over which these events occurred. Thus, small values of $alpha_{prior}$ and $beta_{prior}$ correspond to little information, and this translates into a broader, more diffuse prior distribution for lambda.

> **Do not confuse alpha$_{prior}$ and beta$_{prior}$ here with the parameters of the beta distribution above.**

With the observed data consisting of x failures in time t, the conjugate nature of the prior distribution and likelihood function allows the posterior distribution to be written down immediately using simple arithmetic: the posterior distribution is also a gamma distribution, with new (adjusted) parameters given by:

> **Note that beta$_{prior}$ has units of time, and the units have to be the same as for t.**

$$alpha_{post} = alpha_{prior} + x$$
$$beta_{post} = beta_{prior} + t.$$

From the properties of the gamma distribution (see Appendix B.7.6), the prior and posterior mean of lambda are given by $alpha_{prior}/beta_{prior}$ and $alpha_{post}/beta_{post}$, respectively. Credible intervals for either distribution can be found using the GAMMAINV() function built into modern spreadsheets.

> **Caution: Be sure to know how your spreadsheet software parameterizes the gamma distribution. Most packages use the reciprocal of beta as the parameter.**

- Example 3. Circulating pump fails to operate (Poisson model) and gamma prior.
  The prior distribution for the circulating pump is given as a gamma distribution with parameters $alpha_{prior}$ = 1.6 and $beta_{prior}$ = 365,000 hours. No failures are observed in 200 days of operation.

  Find the posterior mean and 90% interval for the circulating pump failure rate. Use the posterior mean to find the probability that the pump will operate successfully for a mission time of 1,000 hours.

Solution

Because the gamma prior distribution is conjugate with the Poisson likelihood function, the posterior distribution will also be a gamma distribution, with parameters $alpha_{post}$ = 1.6 + 0 = 1.6 and $beta_{post}$ = 365,000 hours + (200 days)(24 hours/day) = 369,800 hours. The posterior mean is the ratio of $alpha_{post}$ to $beta_{post}$, which is 4.3×10^{-6}/hour.

The 90% credible interval is found using the gamma inverse (GAMMAINV) function. Note that most spreadsheet[1] software uses the reciprocal of beta as the second parameter. This can be dealt with either by entering 1/beta as the argument, or entering one as the argument, and dividing the overall result of the function call by beta. Thus, the 5th percentile is given by either GAMMAINV(0.05, 1.6, 1/369800) or [GAMMAINV(0.05, 1.6, 1)]/369800. Either way, the answer is 5.6×10^{-7}/hour. Similarly, the 95th percentile is given by either GAMMAINV(0.95, 1.6, 1/369800) or [GAMMAINV(0.95, 1.6, 1)]/369800. The answer either way is 1.1×10^{-5}/hour.

> **WinBUGS, unlike most spreadsheets, uses beta rather than 1/beta to parameterize the gamma distribution.**

---

[1] For this Handbook, we used Microsoft Excel™ to illustrate the spreadsheet-based examples.

Using the posterior mean failure rate of 4.3×10^{-6}/hour, the probability that the pump operates successfully for 1,000 hours is just exp[-$(4.33 \times 10^{-6}$/hour)(1000 hours)] = 0.996.
The plot below shows how little the prior distribution has been affected by the relatively sparse data in this example.
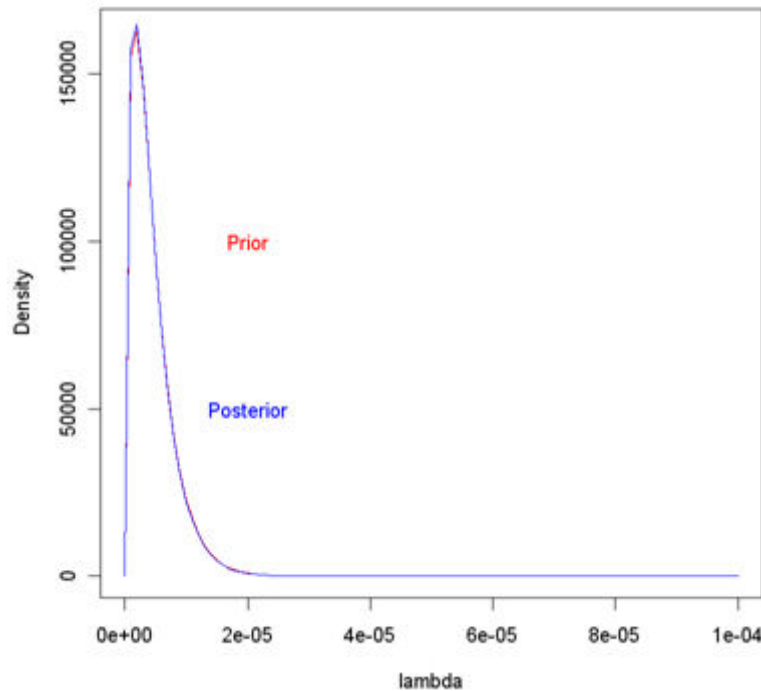
- Figure 4-7. Comparison of prior and posterior distributions for Example 3.

Bayesian inference can also be carried out using WinBUGS. The script below implements the analysis. Monitoring node lambda will display the desired posterior results.

```
model   {
x ~ dpois(mean.poisson)                 # Poisson likelihood function
mean.poisson <- lambda*time.hr          # Parameterize in terms of failure rate, lambda
time.hr <- time*24                      # Convert days to hours
lambda ~ dgamma(1.6, 365000)            # Gamma prior for lambda
}
data
list(x=0, time=200)
```

- Script 4. WinBUGS script for Bayesian inference with Poisson likelihood and gamma conjugate prior.
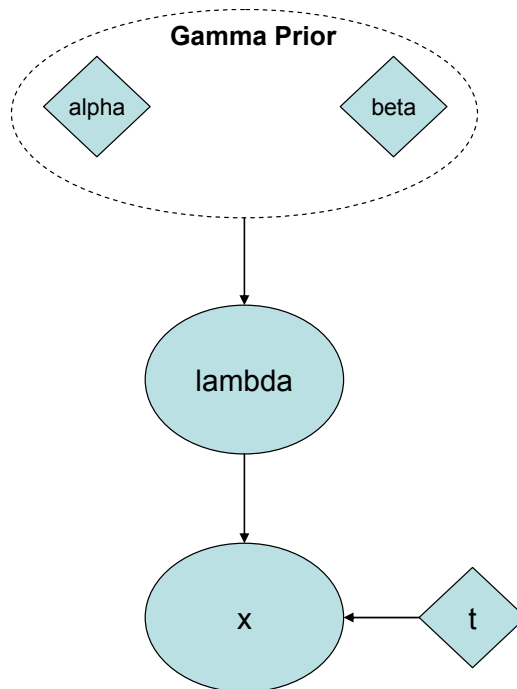
- Figure 4-8. DAG representing Script 4.

4.2.2.2 Poisson Inference with Noninformative Prior—As was the case for the binomial distribution, there are many routes to a noninformative prior for lambda, with the most commonly used one in PRA being the Jeffreys prior (see Appendix B.7.6). In the case of the Poisson likelihood the Jeffreys noninformative prior is like a gamma distribution with $alpha_{prior} = 0.5$ and $beta_{prior} = 0$. This is not a proper distribution, as the integral over all possible values of lambda is not finite. However, it always yields a proper posterior distribution, with parameters $alpha_{post} = x + 0.5$ and $beta_{post} = t$. Thus, the posterior mean of lambda is given by $(x + 0.5)/t$. Inference with the Jeffreys prior can be thought of as a special case of inference with a gamma conjugate prior, with Section 4.2.2.1 applicable.

> **Caution: When using the "zero-zero" noninformative prior as an alternative to the Jeffreys prior, the number of failures (x) must be one or more.**

Note that if x and t are small (sparse data), then adding "half an event" to x may give a result that is felt to be too conservative. In such cases, a possible alternative to the Jeffreys prior is like a gamma distribution with both parameters equal to zero. This is not a proper probability distribution, but as long as x and t are greater than zero, the posterior distribution will be proper and the posterior mean will take on the value (x / t).

4.2.2.3 Poisson Inference with Nonconjugate Prior—As was the case for the parameter p in the binomial distribution, a lognormal distribution is a commonly encountered nonconjugate prior for lambda in the Poisson distribution. The analysis can be carried out with WinBUGS, exactly as was done for p in the binomial distribution. Here, however, there is no concern about values of lambda greater than one, because lambda is a rate instead of a probability, and can take on any positive value, in principle.

- Example 4. Circulating pump fails to operate (Poisson model) and lognormal prior.
  Assume that the prior distribution for the failure rate of the circulating pump is lognormal with a median of 5×10^{-7}/hour and an error factor of 14. Again, assume the observed data are no failures in 200 days. The WinBUGS script below can be used to find the posterior mean and 90% interval for lambda.

```
model  {
x ~ dpois(mean.poisson)              # Poisson distribution for number of events
mean.poisson <- lambda*time.hr       # Poisson parameter
time.hr <- time*24                   # Convert days to hours
lambda ~ dlnorm(mu, tau)             # Lognormal prior distribution for lambda
tau <- 1/pow( log(prior.EF)/1.645, 2)  # Calculate tau from lognormal error factor
mu <- log(prior.median)              # Calculate mu from lognormal median
}

data
list(x=0, time=200, prior.median=5.E-7, prior.EF=14)
```

- Script 5. WinBUGS script for Bayesian inference with Poisson likelihood function and lognormal prior.
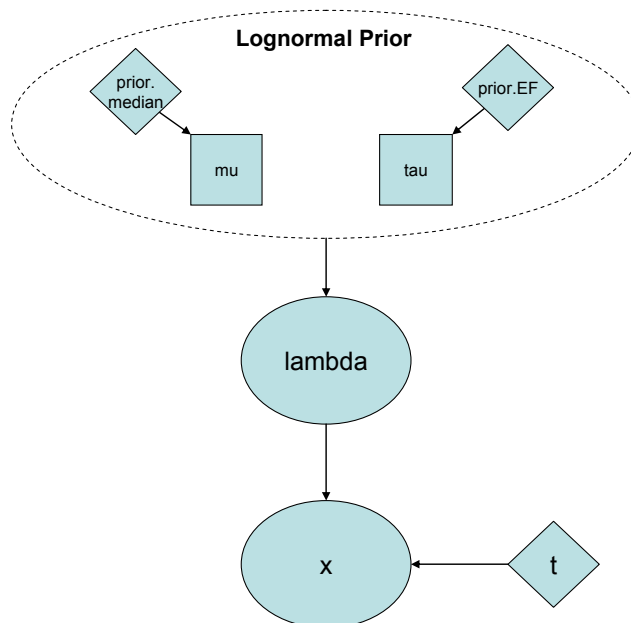
- Figure 4-9. DAG representing Script 5.

Running this script for 100,000 iterations, discarding the first 1,000 iterations to allow for convergence, gives a posterior mean for lambda of 1.6×10^{-6}/hour and a 90% credible interval of $(3.5 \times 10^{-8}$/hour, 6.5×10^{-6}/hour).

4.2.3 Exponential Distribution for Random Durations

There are cases where the times at which random events occur are observed, instead of the number of such events in a specified period of time. Examples are times to failures of components, times to suppress a fire, etc. If the assumptions for the Poisson distribution as described in Section 4.2.2 are

> **If the times at which Poisson-distributed events occur are observed, then the likelihood function is now based on the exponential distribution.**

met, then the times between events are exponentially distributed with unknown parameter lambda; this is the same lambda that appears as the unknown parameter in the Poisson distribution.

The following assumptions underlie the Exponential distribution:
- ☑ The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval. The constant of proportionality is denoted by lambda.
- ☑ The probability of simultaneous events in a short interval of time is approximately zero.
- ☑ The occurrence of an event in one time interval does not affect the probability of occurrence in another, non-overlapping time interval.
- ☑ The random event that is observed is the time to an event.

Because the observed data consist of n number of failure times (with n specified), the form of the likelihood function changes from a Poisson distribution to a product of n exponential distributions. However, much of the analysis is very similar to the analysis done for the Poisson distribution in Section 4.2.2. In this section we treat only the case in which all failure times are observed and known with certainty. In **later** sections we will cover cases in which not all components fail (i.e., censored data), and in which observed failure times have epistemic uncertainty (i.e., the times are uncertain).

> *4.2.3.1 Exponential Inference with Conjugate Prior*—As was the case for the Poisson distribution in Section 4.2.2.1, the conjugate prior for the exponential likelihood is again a gamma distribution, with parameters denoted $alpha_{prior}$ and $beta_{prior}$. Once again, $beta_{prior}$ has units of time, and these units must match the units of the observed times that constitute the data. The posterior distribution will again be a gamma distribution with parameters $alpha_{post} = alpha_{prior} + n$ (the number of observed times), and $beta_{post} = beta_{prior} + t_{total}$, where t_{total} is the sum of the observed times. From the properties of the gamma distribution (see Appendix B), the prior and posterior mean of lambda are given by $alpha_{prior} / beta_{prior}$ and $alpha_{post} / beta_{post}$, respectively. Credible intervals for either distribution can be found using the GAMMAINV() function built into modern spreadsheets.

- Example 5. Circulating pump fails to operate (exponential model) and gamma prior.

> The following seven times to failure (in hours) have been recorded for ATCS circulating water pumps: 55707, 255092, 56776, 111646, 11358772, 875209, and 68978. Using the gamma prior for lambda from Example 3 find the posterior mean and 90% credible interval for the circulating water pump failure rate lambda.

> **Remember to be careful about how your spreadsheet software parameterizes the gamma distribution.**

> **Note the use of the FOR loop to model the n data points in Script 6.**

Solution
The prior distribution was given in Example 3 as gamma with $alpha_{prior} = 1.6$ and $beta_{prior} = 365,000$ hours. In this example, we have $n = 7$ and $t_{total} = 12782181$ hours. Thus, the posterior distribution is gamma with parameters $alpha_{post} = 1.6 + 7 = 8.6$ and $beta_{post} = 365000$ hours $+ 12782181$ hours $= 13147181$ hours. The posterior mean is given by $alpha_{post}/beta_{post} = 6.5 \times 10^{-7}$/hour. The 5th percentile is given by [GAMMAINV(0.05, 8.6, 1)]/13147181 hours $= 3.4 \times 10^{-7}$/hour. The 95th percentile is given by [GAMMAINV(0.95, 8.6, 1)]/13147181 hours $= 1.1 \times 10^{-6}$/hour.

WinBUGS can also be used for this example. The WinBUGS Script 6 shows how to do this.

```
model  {
    for(i in 1:n)  {
        time[i] ~ dexp(lambda)  # Exponential likelihood function for n failure times
    }
lambda ~ dgamma(alpha, beta)           # Gamma prior for lambda
}

data                                   # Note the nested () for the time array
list(time=c(55707, 255092, 56776, 111646, 11358772, 875209, 68978), n=7, alpha=1.6,
beta=365000)
```

• Script 6. WinBUGS script for Bayesian inference with exponential likelihood and gamma conjugate prior.
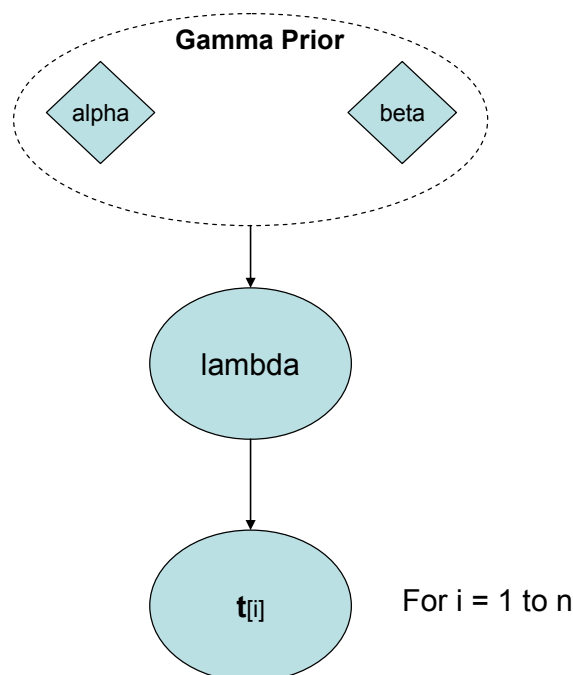
• Figure 4-10. DAG representing Script 6.

***4.2.3.2 Exponential Inference with Noninformative Prior*—**The Jeffreys noninformative prior for the exponential likelihood is like a gamma distribution with both parameters equal to zero. This might seem odd, given the relationship between the exponential and Poisson distributions mentioned above. In fact, it *is* odd that the Jeffreys prior changes, depending on whether one counts failures or observes actual failure times. However, we will not delve into the reasons for this difference and its philosophical implications here. Again, the Jeffreys prior is an improper distribution, but it always results in a proper posterior distribution. The parameters of the posterior distribution will be n and t_{tot}, resulting in a posterior mean of n/t_{tot}. This mean is numerically equal to the frequentist MLE, and credible intervals will be numerically equal to confidence intervals from a frequentist analysis of the data.

> **WinBUGS can only accept proper distributions, so enter the Jeffreys prior for exponential data as dgamma(0.0001, 0.0001). An initial value for lambda will have to be provided, as WinBUGS cannot generate an initial value from this distribution.**

4.2.3.3 Exponential Inference with Nonconjugate Prior—Again, the lognormal distribution is a commonly encountered nonconjugate prior for a failure rate. The only thing that changes from the earlier discussion in Section 4.2.2.3 is the likelihood function, which is now a product of exponential distributions. We again use WinBUGS to carry out the analysis.

- Example 6. Circulating pump fails to operate (exponential model) and lognormal prior.

  Using the prior distribution from Example 4 and the failure times from Example 5, find the posterior mean and 90% interval for the failure rate lambda.

Solution
The WinBUGS script for this example is shown below.

```
model  {
    for(i in 1:n)  {
        time[i] ~ dexp(lambda)  # Exponential likelihood function for n failure times
    }
lambda ~ dlnorm(mu, tau)              # Lognormal prior for lambda
tau <- 1/pow( log(prior.EF)/1.645, 2)   # Calculate tau from lognormal error factor
mu <- log(prior.median)               # Calculate mu from lognormal mean
}

Data                                  # Note the nested () for the time array
list(time=c(55707, 255092, 56776, 111646, 11358772, 875209, 68978), n=7,
prior.median=5.E-7, prior.EF=14)
```

- Script 7. WinBUGS script for Bayesian inference with exponential likelihood function and lognormal prior.
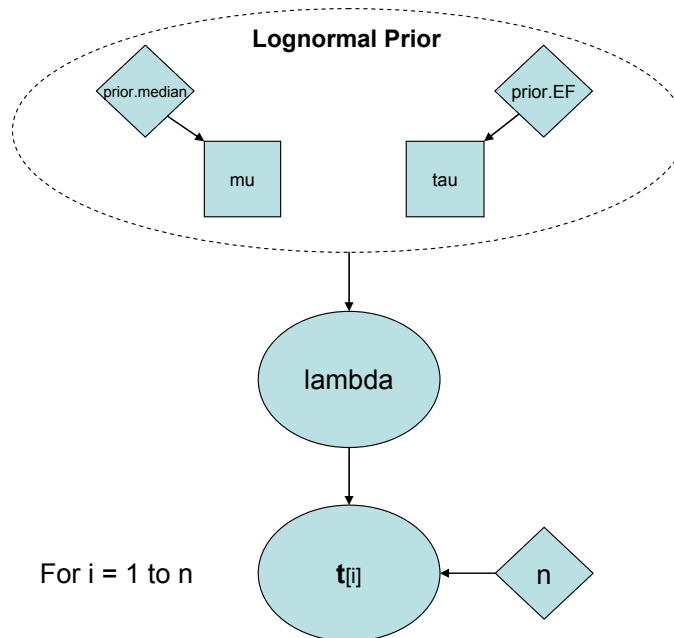
- Figure 4-11. DAG representing Script 7.

Using 100,000 iterations, with 1,000 burn-in iterations discarded to allow for convergence to the posterior distribution, the posterior mean is found to be 5.5×10^{-7}/hour, with a 90% credible interval of $(2.6 \times 10^{-7}$/hour, 9.2×10^{-7}/hour).

4.2.4 Multinomial Distribution

The multinomial distribution generalizes the binomial distribution to more than two possible outcomes. The remaining binomial assumptions continue to apply.

- ☑ There are multiple possible outcomes.
- ☑ There is a constant probability of the ith outcome.
- ☑ The outcomes of earlier trials do not influence the outcomes of later trials.

A common application of the multinomial distribution in PRA is for common-cause failure (CCF), where it serves as an aleatory model for failure of a group of redundant components. Given that a failure occurs in the group, the possible outcomes are one component fails, two fail due to common cause, three fail due to common cause, etc., up to all of the components in the group fail due to common cause. This model can be parameterized in different ways, with the most common being one in which the parameters represent the fraction of failures of the group that involve a specific number of components in the group. These parameters are denoted alpha$_k$, and thus this parameterization is referred to as the *alpha-factor model* for CCF. See [Mosleh, 1991] for additional details on CCF modeling in PRA.

Alpha$_k$ represents the fraction of failures that involve k components of the group. The prior distribution for alpha, which is now a vector of dimension equal to the component group size, is almost always assumed to be a Dirichlet distribution, which is **conjugate** to the multinomial likelihood. Like alpha, the parameter of the Dirichlet distribution, which we will denote as theta, is a vector of dimension equal to that of alpha. Because the Dirichlet distribution is conjugate to the multinomial likelihood, the posterior distribution for alpha will also be Dirichlet, with parameter theta + n, again a vector, with k^{th} component equal to theta$_k$ + n$_k$.

If a vector theta has a Dirichlet distribution, then each constituent of theta has a beta distribution, each with parameters theta$_k$ and theta$_{tot}$ – theta$_k$, with theta$_{total}$ = Σ theta$_k$. A noninformative prior for the multinomial likelihood is a Dirichlet prior with each theta$_k$ = 1.

The observed data consist of failure counts, n$_k$, with n$_k$ being the number of failures in the group that involve k components. Often in CCF analysis, it can be difficult to judge the exact values of the n$_k$s, leading to epistemic uncertainty about the observed data. Here, we will assume that the n$_k$s are known with certainty; in a **later** section, we will examine the more usual case in which the n$_k$s are subject to epistemic uncertainty.

- Example 7. Multinomial model for common-cause failure – estimating CCF parameters.

> A group of three redundant components has suffered 33 failures. Of these 30 failures 20 involved 1 component, 5 involved 2 components, and 1 involved all 3 components. Estimate alpha factors and corresponding parameters of another common CCF model, the Multiple Greek Letter (MGL) model, given this data.

The MGL model does not have a well-defined likelihood function, making it hard to do direct Bayesian inference for the parameters of this model. The easiest way is to estimate the alpha-factor parameters and use WinBUGS to transform to the MGL parameters, as done here.

Solution

We will assume a noninformative prior for alpha, which will be a Dirichlet distribution with all three parameters equal to 1. The WinBUGS script shown in Script 8 updates this distribution with the given data, and also transforms the posterior distribution for each component of alpha into the corresponding parameter (beta, gamma) of the MGL model. Running the script in the usual way gives the results shown in Table 2.

- Table 2. CCF example parameter results.

Parameter	Mean	90% Interval
Alpha-1	0.72	(0.58, 0.85)
Alpha-2	0.21	(0.10, 0.34)
Alpha-3	0.07	(0.01, 0.16)
Beta	0.28	(0.15, 0.42)
Gamma	0.25	(0.05, 0.52)

```
model   {
n[1:group.size] ~ dmulti(alpha[1:group.size], N)
# The variable group.size is loaded via data block
N <- sum(n[1:group.size])                # N is the total number of 'group failure events'
alpha[1:group.size] ~ ddirch(theta[])    # The noninformative prior for alpha
theta[1] <- 1                            # A noninformative prior for the multinomial
theta[2] <- 1                            #  likelihood is a Dirichlet prior with each theta_k = 1
theta[3] <- 1
# Calculate MGL parameters (from Table A-2 in NUREG/CR-5485)
beta <- alpha[2] + alpha[3]
gamma <- alpha[3]/(alpha[2] + alpha[3])
}
data
list(n=c(20, 5, 1), group.size=3)
```

- Script 8. WinBUGS script for estimating alpha factors and MGL parameters for a component group of size 3, no data uncertainty.
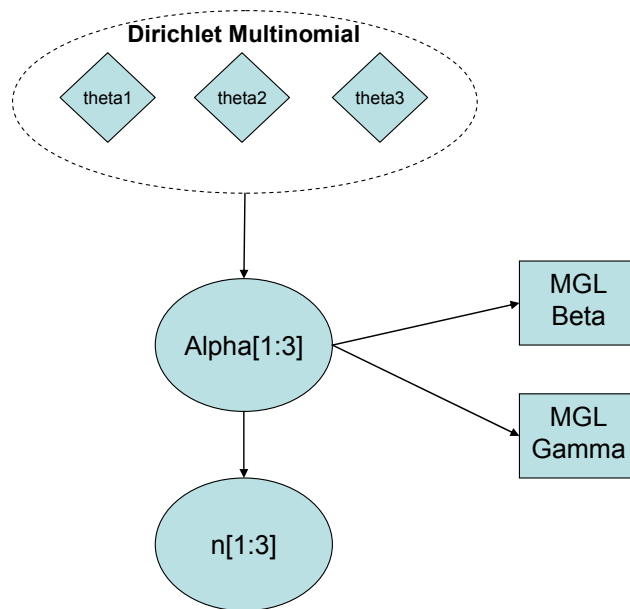
- Figure 4-12. DAG representing Script 8.

4.2.5 Developing Prior Distributions

In practice, the analyst must develop a prior distribution from available engineering and scientific information, where the prior should reflect (a) what information is know about the inference problem at hand and (b) be independent of the data that is collected. This section provides some high-level guidance and examples for doing this in single-parameter problems. In this section we consider only priors developed from a single source of information; later sections will deal with developing priors from multiple sources of information.

    4.2.5.1 Developing a Conjugate Prior—The beta and gamma distributions used as conjugate priors in the earlier sections each have two parameters that specify the distribution. Therefore, two pieces of information are generally needed to select such a conjugate prior. Common information from which the analyst must develop a prior is:
1. A measure of central tendency (e.g., median or mean) and an upper bound (e.g., 95th percentile)
2. Upper and lower bound (e.g., 95th and 5th percentile)
3. A mean and variance (or standard deviation).

We discuss each of these three cases below.

        4.2.5.1.1 Using Mean or Median and Upper Bound—When the information provided[1] takes the form of a mean or median value and an upper bound, numerical analysis is required in order to find a gamma or beta distribution satisfying this information. Fortunately, modern spreadsheet tools make such analysis feasible. Note that "bound" is not usually interpreted in an absolute sense as a value that cannot be exceeded. Instead, it is interpreted as an upper percentile of the distribution. The 95th percentile is the most common choice in PRA, but other percentiles, such as the 85th, can be chosen. We illustrate the numerical solution with some examples.

---

[1] The "information provided" represents the analyst's state of knowledge for the system or component being evaluated and is independent from any data to be used as part of the Bayesian inference.

- Example 8. Developing a gamma prior for transistor failure rate in avionics portion of ATCS using point estimate and upper bound.

> The failure rate of a bipolar transistor in the avionics control circuit of the ATCS is to be estimated. The information in MIL-HDBK-217F is used to develop a gamma conjugate prior distribution for the failure rate, lambda.
>
> MIL-HDBK-217F provides a point estimate of 2×10^{-10}/hour. The analyst does not have much confidence in the accuracy of this estimate and therefore decides to select a 95th percentile that is ten times the point estimate (i.e., 2×10^{-9}/hour). Develop a gamma prior distribution assuming:
> - The point estimate is a median value.
> - The point estimate is a mean value.

Solution

Part 1. There are two equations that must be solved numerically to find alpha and beta(the parameters of the gamma prior distribution). First, using the median value, we can write GAMMAINV(0.5, alpha, 1)/beta = 2×10^{-10}/hour. Second, using the 95th percentile, we can write GAMMAINV(0.95, alpha, 1)/beta = 2×10^{-9}/hour. We set up the spreadsheet as shown below to allow use of the SOLVER function.

Now bring up the SOLVER menu from the Tools drop-down menu.

We are telling SOLVER to adjust alpha and beta until the median value in cell C2 is equal to 2×10^{-10}/hour. As constraints, we tell SOLVER that the 95th percentile in cell E2 should be 2×10^{-9}/hour. Further, alpha and beta must both be greater than zero. To avoid numerical problems, it is a good idea

to limit alpha to values greater than about 0.1, as shown above. The user can change the precision of the solution by choosing the Options button on the screen above. The answers above were attained by setting the initial alpha and beta values to 0.1 and 0.01 respectively and setting the precision to 1E-10. Higher precision (e.g., 1E-11) leads to numerical problems, so the closest we can come to reproducing the desired median and upper bound is a gamma distribution with alpha equal to about 0.6 and beta equal to about 1.2×10^9 hours.

Part 2. Again, there are two equations to solve numerically. From Appendix B, we know the mean of a gamma distribution is given by alpha/beta. The second equation is the same as we used above in terms of the 95th percentile. However, in this case we can use the properties of the gamma distribution to simplify the problem. Taking the ratio of the 95th percentile to the mean, we find this ratio is equal to GAMMAINV(0.95, alpha, 1)/alpha. We set up the spreadsheet as shown below to allow use of either the SOLVER or Goal Seek function.

> **Note that beta, although shown in the spreadsheet, is not relevant to this calculation. Only alpha affects the spread of the gamma distribution.**

Unfortunately, we find that we cannot make the ratio of the 95th percentile to the mean larger than about 5.8; this is an inherent property of the gamma distribution. Therefore, in this case we cannot treat the point estimate as a mean value. One way around this problem is to treat the upper bound as a 99th percentile. If we do this, and use SOLVER to set the ratio to a value of 10, we find that alpha is about 0.24. We would then use the equation for the mean to find beta equal to about 1.2×10^9 hours.

- Example 9. Developing a beta prior mixing valve failure probability in ATCS system using point estimate and upper bound.

> We are estimating the probability that the mixing valve in the ATCS system fails to change position in response to a signal from the avionics unit. Our past experience with similar valves in similar applications suggests a point estimate failure probability of 10^{-4}, with an upper bound of 5×10^{-4}.
>
> Assuming the upper bound is the 95th percentile, find the beta conjugate prior that encodes this information assuming 1) the point estimate is the median, and 2) assuming the point estimate is the mean.

Solution

1. Again, there are two equations in two unknowns, which we must solve numerically to find alpha and beta, the parameters of the beta prior distribution that encodes the information provided. Using the median value, we can write BETAINV(0.5, alpha, beta) = 10^{-4}. Then, using the upper bound, we can write BETAINV(0.95, alpha, beta) = 5×10^{-4}. Unfortunately, SOLVER cannot find a solution to these two equations; the beta distribution is numerically difficult to work with. However, there is an approximation that can be used. When p is small, the beta distribution can be approximated by a gamma distribution, allowing us to analyze the problem as in Example 8 above. Therefore, we set up the spreadsheet as shown below.

> **Note the lack of units in this case, since we are dealing with a failure probability, not a failure rate.**

> **Always check the validity of approximations wherever possible.**

	C2		f_x =GAMMAINV(0.5,A2,1)/B2	
	A	B	C	D
1	alpha	beta	median-beta	95th-beta
2	0.785315	5.21E+03	9.36E-05	4.93E-04

Using SOLVER gives the results shown. Note the use of the GAMMAINV function rather than the BETAINV() function. So how good was the approximation? Substituting the estimated values of alpha and beta into the BETAINV() function, we find a median of 9.4×10^{-5} and a 95th percentile of 4.9×10^{-4}, so the approximation of the beta distribution by a gamma distribution was quite good.

2. If the point estimate is treated as a mean value, we have as our first equation (from Appendix B), that alpha/(alpha + beta) = 10^{-4}. The second equation is the same as in part (1) above. We set the spreadsheet up as shown below to allow use of the SOLVER function.

	C2		f_x =A2/(A2+B2)	
	A	B	C	D
1	alpha	beta	mean-beta	95th-beta
2	0.332719	2.99E+03	1.11E-04	0.000492405

> **Although Bayesian inference with a conjugate prior is mathematically easy, it may not be so easy to find the conjugate prior in the first place.**

Running SOLVER, we obtain the values shown in the spreadsheet. Because one of the equations to be solved was algebraic, we did not need to approximate the beta distribution with a gamma distribution. However, we could have used this approximation because p is small.

> ***4.2.5.1.2 Using Upper and Lower Bound***—Sometimes information will be provided in the form of a range, from a lower bound to an upper bound. As above, the bounds are not absolute bounds on the parameter value, but are lower and upper percentiles (e.g., 5th and 95th) of the prior distribution. Again, we will have to resort to numerical methods to find the parameters of a conjugate prior distribution.
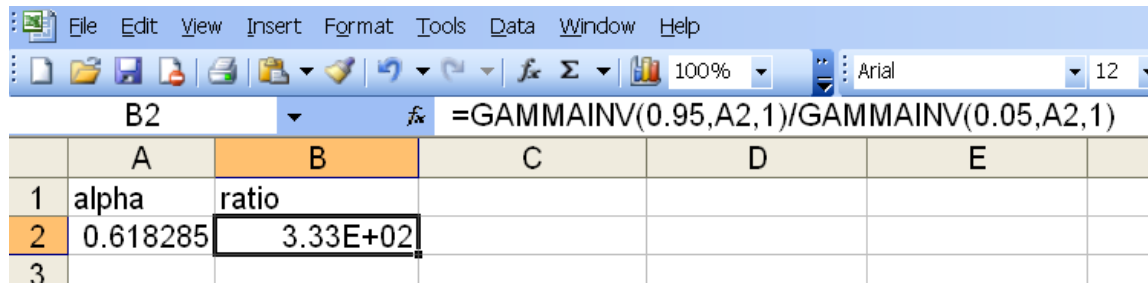
• Example 10. Developing gamma prior for circulation pump failure rate in ATCS system using upper and lower bounds.

> Assume the ATCS shown earlier is a new design, so that only very limited information is available. Knowing only that the circulation pump is mechanical in nature, a generic information source provides a lower bound on the failure rate of 3×10^{-6}/hour, and an upper bound of 10^{-3}/hour.
>
> Treating these as 5th and 95th percentiles, respectively, find the gamma conjugate prior distribution that encodes this information.

Solution
For the gamma distribution, the ratio of the 95th to the 5th percentile can be written as GAMMAINV(0.95, alpha, 1)/GAMMAINV(0.05, alpha, 1). We set up the spreadsheet as shown below and use either Goal Seek or SOLVER to find alpha such that the ratio is equal to $10^{-3}/3 \times 10^{-6}$ = 333. The value of alpha is that shown, which is about 0.62.
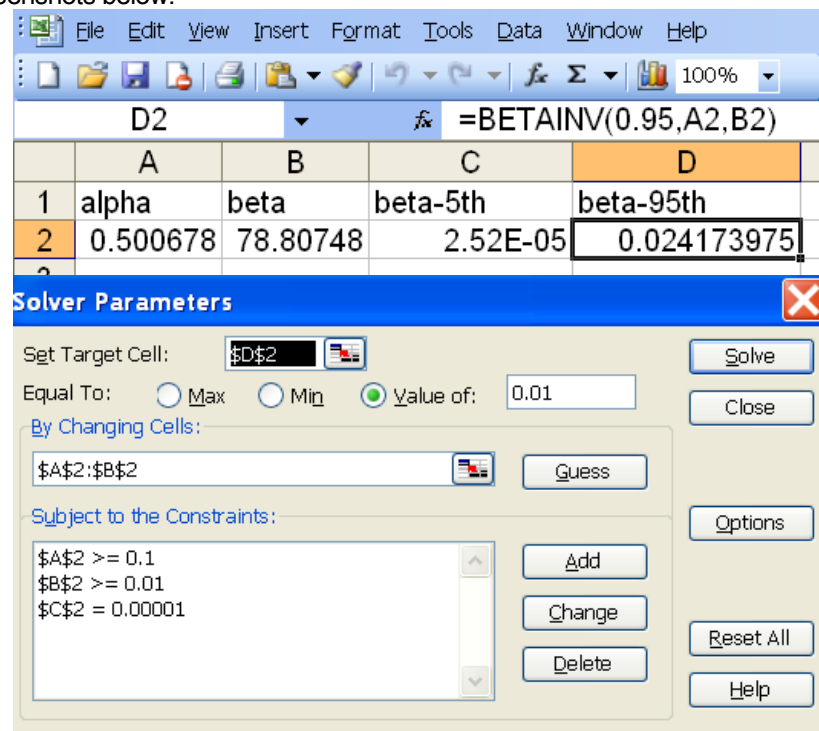
After we have found alpha, we can use the equation for either the 5th or 95th percentile to find beta. Because PRA is usually more concerned with large failure rates, we will use the 95th percentile. Thus, we have beta = GAMMAINV(0.95, 0.62, 1)/10^{-3} = 2,200 hours.

- Example 11. Developing beta prior for check valve in ATCS system using upper and lower bounds.

> Suppose one of the failure modes in our newly designed ATCS system is failure of the check valve to close to prevent reverse flow. Again, since this is a new design, assume we have only bounds on the check valve failure probability, and that these bounds are 10^{-5} to 10^{-2}. Find a conjugate beta prior distribution that encodes this information, assuming these bounds are 5th and 95th percentiles, respectively.

Solution

We have two equations that must be solved numerically to find alpha and beta. Treating the bounds as 5th and 95th percentiles, the equations are BETAINV(0.05, alpha, beta) = 10^{-5} and BETAINV(0.95, alpha, beta) = 10^{-2}. Using SOLVER, we must decide which of these values is the target and which is the constraint. Because PRA is more concerned with high failure probability, we will use the upper bound as the target and treat the lower bound as a constraint. This is shown in the spreadsheet and SOLVER screenshots below.

Using the Options button, the precision and convergence were both set to 0.1 to avoid numerical problems. This resulted in the values of alpha and beta shown above. Because p is relatively small, we could have also approximated the beta distribution with a gamma distribution, as shown earlier. The spreadsheet for doing this is shown below.

	A	B	C	D	E	F
	B2	=GAMMAINV(0.95,A2,1)/GAMMAINV(0.05,A2,1)				
1	alpha	ratio				
2	0.497938	1000				
3						

> **Sometimes approximations can give better results when they help to avoid tricky numerical calculations.**
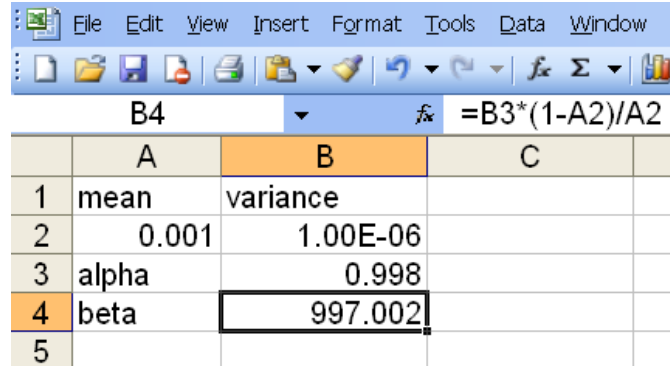
Using Goal Seek or SOLVER, we find that a value for alpha of about 0.5 gives the desired ratio of upper bound to lower bound. We can then use the upper bound to find beta, using the equation beta = GAMMAINV(0.95, 0.5, 1)/0.01 = 192. Checking the approximation, we have BETAINV(0.05, 0.5, 191.6) = 1.0E-05 and BETAINV(0.95, 0.5, 191.6) = 1.0E-02. In this case, because of the numerical difficulties caused by the beta distribution, the approximation gives a better result.

4.2.5.1.3 Using Mean and Variance or Standard Deviation—This is the easiest situation to deal with, but perhaps the least frequently encountered in practice. It is relatively easy because the equations are algebraic and do not require numerical solution. Unfortunately, our information sources are not often encoded in terms of the mean and variance or standard deviation (the square root of the variance). However, some analysts in the past, in an attempt to avoid working with a nonconjugate prior, have converted the encoded information into a mean and variance in order to replace the nonconjugate prior with a conjugate prior using simple algebraic calculations. This can lead to nonconservatively low estimates in some cases, however, and is not recommended in today's environment where tools such as WinBUGS make Bayesian inference with a nonconjugate prior straightforward.

For the gamma distribution (see Appendix B), the mean as alpha/beta and the variance as alpha/beta^2. Thus, we find beta = mean/variance, and then we can substitute in the value of beta to find alpha = beta/mean. If the standard deviation is given instead of the variance, we can square the standard deviation and proceed as above.

The beta distribution is just a bit trickier algebraically. From Appendix B, the mean is equal to alpha/(alpha + beta) and the variance is a complicated expression in terms of alpha and beta. This expression can be rewritten in terms of the mean as mean(1 – mean)/(alpha + beta + 1), and a spreadsheet can be used to solve for alpha and beta, as shown below.

	A	B	C	D
	B3	=A2^2*(1-A2)/B2-A2		
1	mean	variance		
2	0.001	1.00E-06		
3	alpha	0.998		
4	beta	997.002		
5				
6				

4.2.5.2 Developing a Nonconjugate (Lognormal) Prior—One of the things that makes the lognormal distribution attractive as a prior in PRA is the ease with which it can encode information about a parameter that varies over several orders of magnitude. As illustrated above (e.g., Script 2), the information encoded about the lognormal distribution is not usually provided in terms of the parameters (mu and tau) needed by WinBUGS. More commonly, information is given in terms of a median or mean value and an error factor, or sometimes in terms of an upper and lower bound. Using the properties of the lognormal distribution given in Appendix B, any of these sets of information can be translated into the mu and tau parameters needed by WinBUGS, as shown in the script excerpts below.

```
# Use the following lines if median and error factor given
mu <- log(median)
tau <- pow(log(EF)/1.645, -2)

# Use the following lines if mean and error factor given
mu <- log(mean) - pow(log(EF)/1.645, 2) / 2
tau <- pow(log(EF)/1.645, -2)

# Use the following lines if median and upper bound given
mu <- log(median)
tau <- pow(log(upper/median)/1.645, -2)

# Use the following lines if mean and upper bound given
# Caution:  mean/upper must be > 0.258
tau <- pow(sigma, -2)
sigma <- (2*1.645 + sqrt(4*pow(1.645, 2) + 8*log(mean/upper))) / 2
mu <- log(mean) – pow(sigma, 2)/2

# Use the following lines if upper and lower bound given
mu <- log(sqrt(upper*lower))
tau <- pow(log(sqrt(upper/lower)/1.645, -2)
```

- Script 9. Excerpts of WinBUGS script used to find mu and tau for lognormal prior.

4.2.5.3 Developing a Prior from Limited Information—In some cases, not enough information may be available to completely specify an informative prior distribution, as two pieces of information are typically needed. For example, in estimating a failure rate, perhaps only a single estimate is available. This section describes how to use such limited information to develop a prior distribution, which encodes the available information with as much epistemic uncertainty as possible, reflecting the limited information available. As expected, because the information on which the prior is based is very limited, the resulting

> **All of these distributions are conjugate except the last one.**

prior distribution will be diffuse, encoding significant epistemic uncertainty about the parameter value. However, it will not be as diffuse as the noninformative priors discussed in Sections 4.2.1.2, 4.2.2.2, and 4.2.3.2. The table below summarizes the results for commonly encountered cases.

- Table 3. Prior distributions encoding limited information about parameter values.

Information Available	Suggested Prior Distribution
Mean value for lambda in Poisson distribution	Gamma distribution with alpha = 0.5 and beta = 1/(2 × mean)
Mean value for p in binomial distribution	Beta distribution with alpha ≈ 0.5 and beta = (1 – mean)/(2 × mean)
Mean value for lambda in exponential distribution	Gamma distribution with alpha = 1 and beta = 1/mean
p in binomial distribution lies between a and b	Uniform distribution between a and b

4.2.5.4 Ensure Prior is Consistent with Expected Data—Preposterior Analysis—For the final step in selecting an informative prior distribution, use the candidate prior distribution to generate potential data; if it turns out to be extremely unlikely that the prior distribution can produce data that is expected, then the prior has **not** encoded the analyst's state of knowledge adequately. In the past, such a check was difficult because it requires numerical calculations. However, modern tools such as WinBUGS make the analysis straightforward. We illustrate the method with a couple of examples.

- Example 11. Preposterior analysis for refrigerant tank in ATCS system.

> In Example 1, the prior for the relief valve on the refrigerant tank in the ATCS failing to open on demand was given as a beta distribution with alpha = 1.24 and beta = 189,075.
> Suppose the analyst thinks that seeing 2, 3, or possibly even 4 failures over 350 demands would **not** be surprising. Is this belief consistent with the stated beta prior distribution?

Solution

> **This example, which involves a conjugate prior, could be done in a spreadsheet. However, WinBUGS can do the same calculation with a nonconjugate prior, a calculation which cannot be done quite so easily with a spreadsheet.**

To solve this problem, we need to calculate the probability of seeing 2 or more failures to open in 350 demands on the relief valve (since we expect to see at least 2 failures). The probability of seeing x failures in n demands is given by the binomial distribution, but this probability is conditional upon a value of p, and p is uncertain. The specified beta prior distribution is a candidate for describing the analyst's epistemic uncertainty about p. If it is compatible with his belief that 2 or more failures in 350 demands are likely, then a weighted-average probability of seeing at least 2 failures in 350 demands, with the weights supplied by the candidate prior distribution, should not be too small. We use the WinBUGS script shown below to estimate the unconditional probability of seeing 2 or more failures in 350 demands, based on the candidate beta prior distribution. Note that this script tests the prior [beta(1.24, 189,075)] since we are **not** assigning any data to node **x**.

```
model   {
x ~ dbin(p, n)              # Binomial likelihood function
p ~ dbeta(alpha, beta)    # Beta prior distribution for p
p.value <- step(x - x.exp)        # If mean value of this node is < 0.05, prior is incompatible
}                               # with expected data

data
list(alpha=1.24, beta=189075, x.exp=2, n=350)
```

- Script 10. WinBUGS script for preposterior analysis.
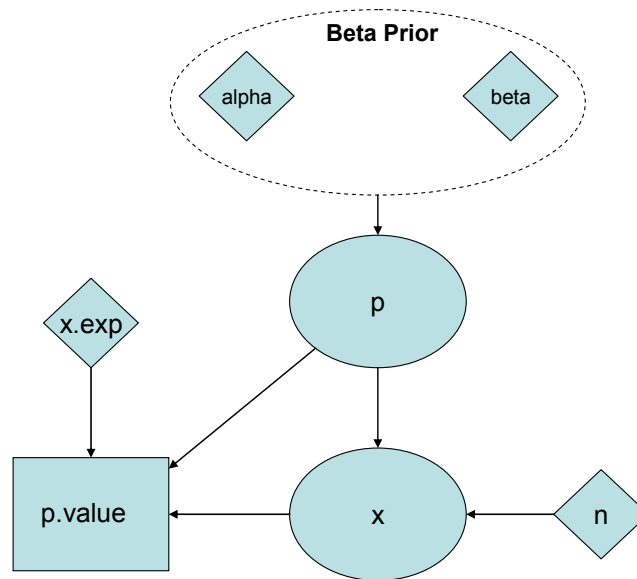
- Figure 4-13. DAG representing Script 10.

Running this script for 100,000 iterations with 1,000 burn-in iterations for convergence gives a mean for the p.value node of 0.0, indicating that the expected data are very incompatible with the candidate beta prior distribution. The concept of a Bayesian "p-value" is shown in Figure 4-14, where results with a low (less than 0.05) p-value represents cases where the posterior is much lower than the expected – or predicted – distribution. The converse, when the p-value is high (greater than 0.95), represents cases where the posterior is larger than expected. The ideal case is obtained when the p-value is close to 0.5 – this case represents the posterior being close to the expected results. Note that the "expected results" could be a constant instead of a distribution and we could check against the prior instead of the posterior, as was the case in Script 10, where we tested the prior against the possibility of seeing two failures (given by x.exp).
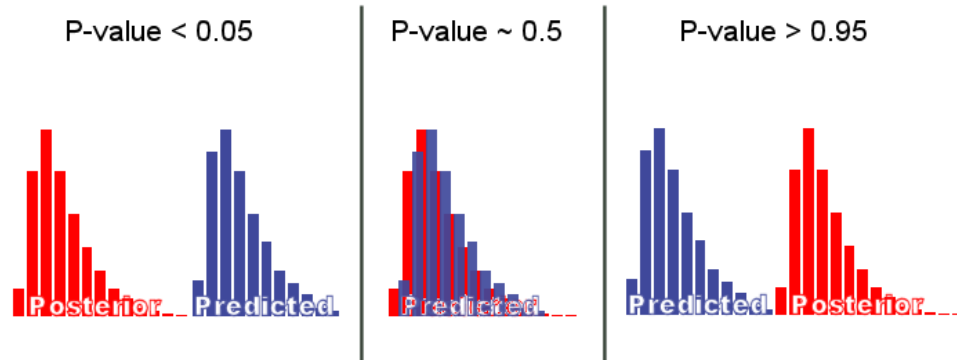
- Figure 4-14. Illustration of the Bayesian p.value calculation used to test on prior, posterior, or model validity.

4.3 Model Validation

In this section, we examine the predictions of our Bayesian inference model ("model" for short) as a test of how good the model is. Recall that the Bayesian inference model comprises the prior distribution, (representing epistemic uncertainty in parameter values) and the likelihood function (representing aleatory uncertainty in our probabilistic model of the world). We will primarily examine how well our model can replicate the observed data; models for which the observed data are highly unlikely to be replicated are problematic and will lead us to alternative prior distributions or to more complicated likelihood functions, such that the resulting model is better able to replicate the observed data. Our tool for this analysis is WinBUGS.

4.3.1 Checking the Validity of the Binomial Model

Recall that the binomial distribution, used as an aleatory model for failures on demand, is derived from underlying assumptions:
- ☑ There are two possible outcomes of each demand typically denoted success and failure.
- ☑ There is a constant probability of failure (typically in PRA; typically success in reliability engineering) on each demand, denoted herein as p.
- ☑ The outcomes of earlier demands do not influence the outcomes of later demands.
- ☑ The model contains a single unknown parameter.
- ☑ The prior information is homogeneous and is known with certainty[1].
- ☑ The observed data are homogeneous and are known with certainty.

We have also selected a prior distribution for p, and Bayes' Theorem has been used to combine this prior with observed data, producing a posterior distribution for p. The probability of observing *future* data can be calculated by averaging the binomial distribution for the number of failures in n demands over the posterior distribution for p. If this probability is small, then the model is suspect. Perhaps the prior distribution is at fault, or perhaps one of the binomial assumptions listed above is grossly violated. Further investigation will be required to identify where the breakdown has occurred.

- Example 12. Posterior predictive check for relief valve in the ATCS.

> In Example 1, the prior for the relief valve on the refrigerant tank in the ATCS failing to open on demand was given as a beta distribution with alpha = 1.24 and beta = 189075.
>
> The observed data were 2 failures in 285 demands. Find the probability of seeing at least this many failures in the next 285 demands.

Solution

In the WinBUGS script shown below, x.rep represents the predicted number of failures in the next n demands. The p.value node calculates the probability that x.rep is at least as big as the observed value, x. If the mean of this node is less than about 0.05, a problem may exist with the model, either in the prior distribution or in one or more of the assumptions underlying the binomial distribution.

Running the script for 100,000 iterations, with 1,000 burn-in iterations for convergence, gives a mean for the p.value node of 10^{-5}, indicating the model is not able to replicate the observed data; it severely under-predicts the number of failures. At this point, the analyst may decide that the choice of prior distribution was overly optimistic because it puts high probability on small values of p, or perhaps an engineering investigation may reveal that the two observed failures were not independent, violating one of the assumptions behind the binomial distribution. This could be the case, for example, if the cause of the first failure was not completely repaired. However, in this example the probability of seeing 1 or more failure in 285 demands is only about 0.005, suggesting that the prior distribution for p is inconsistent with the actual performance of the relief valve.

---

[1] For discussion and examples of uncertain priors and data, see Section 4.8.

```
model  {
x ~ dbin(p, n)            # Binomial likelihood function
x.rep ~ dbin(p, n)        # Likelihood function for replicated data
p ~ dbeta(alpha, beta)    # Beta prior distribution for p
p.value <- step(x.rep - x)         # If the mean value of this node is < 0.05, there is a
problem with the model
}

data
list(alpha=1.24, beta=189075, x=2, n=285)
```

- Script 11. WinBUGS script to generate replicate data from binomial distribution.
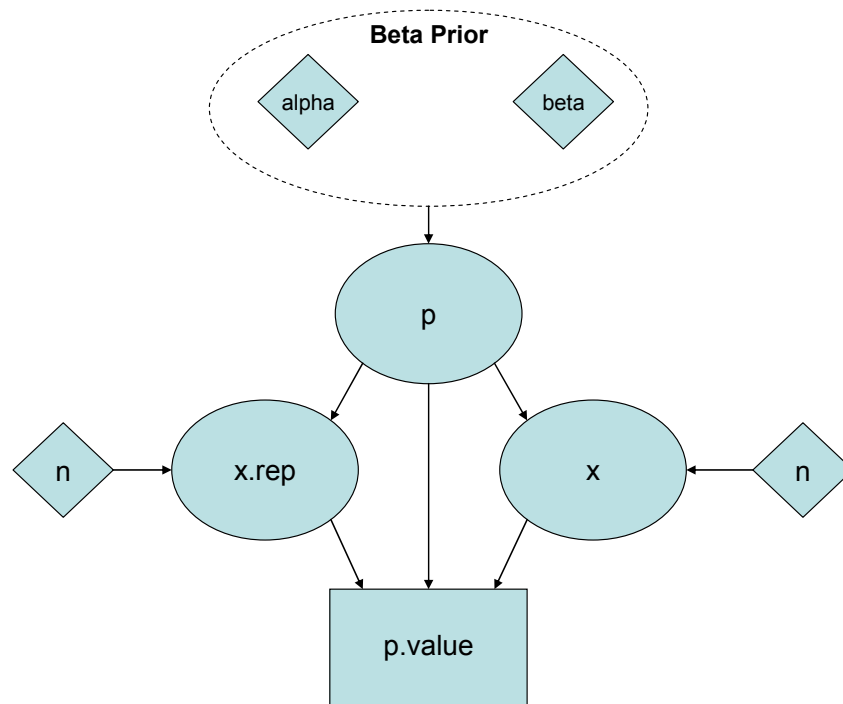
- Figure 4-15. DAG representing Script 11.

- Example 13. Check for time trend in relief valve leakage probability in the ATCS.

Consider a case where we are concerned about the relief valve on the refrigerant tank in the ATCS system leaking, a new failure mode not considered earlier.   Assume that a check is performed once a week on this valve and a notation is made of whether or not the valve is leaking. The analyst decides to use a binomial distribution for the number of times the valve is leaking in a year, so that n is 52. The unknown parameter of interest is p, the probability that the valve is leaking in any given week. Assume that over time, the data below has been collected for this valve.

Year	Failures	Demands
1	4	52
2	2	52
3	3	52
4	1	52
5	4	52
6	3	52
7	4	52
8	9	52
9	6	52

The analyst decides to use a Jeffreys prior with the data pooled across the nine years of leak-testing (36 failures in 486 demands). This results in a posterior mean for p of 0.08 and a 90% credible interval of (0.06, 0.099). Investigate the validity of the analyst's approach.

Solution
One of the assumptions underlying the binomial distribution is that p does not change from one demand to the next. In our example, where data has been collected over a period of time, this also implies that p should not change over time. A qualitative way to check this assumption is to calculate a 95% credible interval for each of the nine years, plot them side by side, and look for overlap. If the intervals all overlap with one another, this is evidence that p is not varying significantly over time. Note that p could exhibit a time trend, perhaps increasing over time due to wearout of the valve, or it could vary significantly from one year to the next. Either situation could make the simple binomial likelihood a poor model.

The WinBUGS script used to construct a side-by-side plot of the credible intervals for each year (using a Jeffreys noninformative prior) is shown in Script 12. The block data is loaded by highlighting the first letter of the block data, in this case "x", and clicking *load data* in the specification tool. The second data list is loaded separately.

```
model  {
    for (i in 1:N) {
    x[i] ~ dbin(p[i], n[i])          # Binomial distribution for failures in each year
    p[i] ~ dbeta(0.5, 0.5) # Jeffreys prior for p, used in developing interval plot of p
    }
}
Data    # Illustrate the use of "block data" entry format for x failures in n demands
x[] n[]
4    52
2    52
3    52
1    52
4    52
3    52
4    52
9    52
6    52
END
list(N=9)
```

• Script 12. WinBUGS script to generate caterpillar plot for p with block data entry.
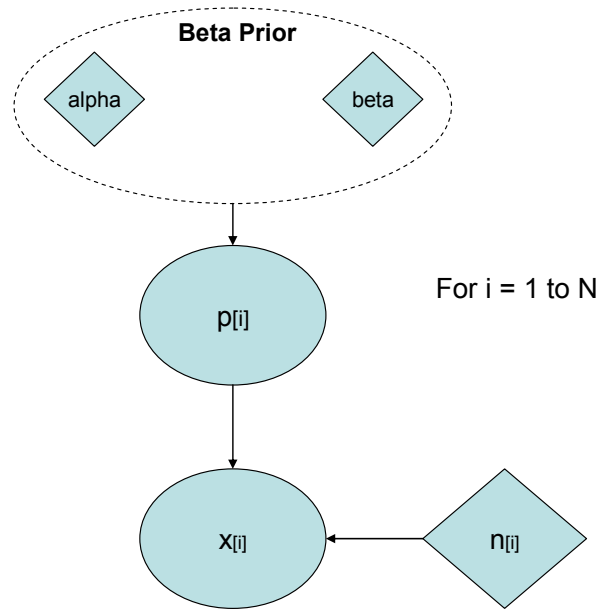
- Figure 4-16. DAG representing Script 12.

> **Note the use of the block format for entering data. Also, there are now two steps to loading data, first the data block then the list.**

This script was run for 100,000 iterations, discarding the first 1,000 iterations to allow for convergence. The caterpillar plot created through the inference comparison tool produced Figure 4-17 which shows the 95% credible intervals for p in each year. The plot suggests that p might be increasing with time, but there is a lot of uncertainty, so it is difficult to judge whether a trend may actually be present. A more quantitative measure is needed.
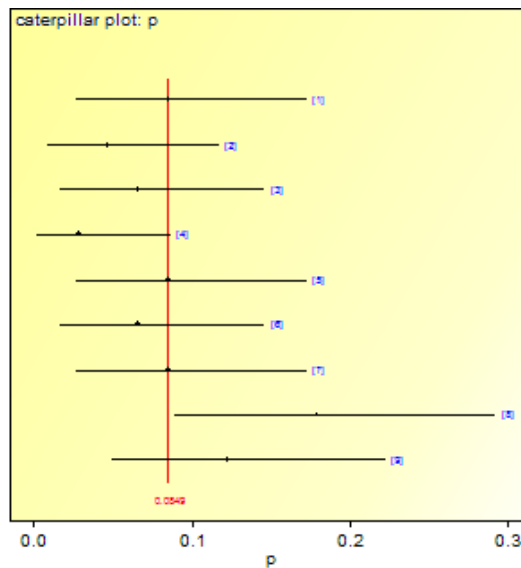
- Figure 4-17. Side-by-side plot of 95% credible intervals for p in each of the 9 years. Dots indicate posterior mean, and red line is the average of the posterior means.

The WinBUGS Script 13 is used to generate a quantitative measure of how well a model that pools the data can replicate the observed data in each year. A Jeffreys prior is used for the single parameter (p.constant) that is estimated. Running this script in the usual way gives a mean for the p.value node of 0.18. If the model were good at replicating the observed data, we would expect the mean of this node to be near 0.5. Monitoring the x.rep node shows that the constant-p model tends to predict too many failures in early years and too few in later years, suggesting that a trend model might be better at replicating the observed data. We will examine this model in Section 4.4.1 below.

```
model  {
    for (i in 1:N) {
        x[i] ~ dbin(p[i], n[i])        # Binomial distribution for failures in each year
        x.rep[i] ~ dbin(p[i], n[i])  # Replicate value from posterior predictive distribution
        diff.rep[i] <- pow(x.rep[i] - n[i]*p[i], 2)/(n[i]*p[i]*(1-p[i]))     # chi-square-type metric
        diff.obs[i] <- pow(x[i] - n[i]*p[i], 2)/(n[i]*p[i]*(1-p[i]))        # chi-square-type metric
        p[i] <- p.constant        # Constant parameter in each year, resulting in pooling
    }                                        # of data
chisq.rep <- sum(diff.rep[])                    # Sum the "replicated" chi-square-type metric
chisq.obs <- sum(diff.obs[])                    # Sum the observed chi-square-type metric
p.value <- step(chisq.rep - chisq.obs)      # If the "replicated" metric is larger than
                                                # observed add 1 to p.value (for each iteration)
                                                # otherwise add 0
p.constant ~ dbeta(0.5, 0.5)                # Jeffreys prior for p.constant
    }
Data
x[]  n[]
4    52
2    52
3    52
1    52
4    52
3    52
4    52
9    52
6    52
END
list(N=9)
```

• Script 13. WinBUGS script for testing assumption that p in binomial distribution does not vary over time.
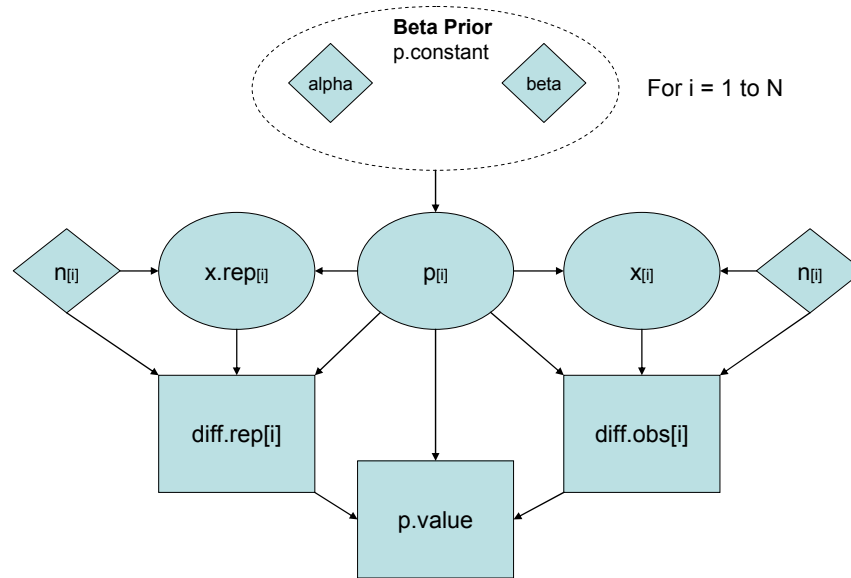
- Figure 4-18. DAG representing Script 13 and Script 14.

- Example 14. Check for source-to-source variability for circuit breaker failure data in the ATCS.

In this example we have data from multiple sources that we would like to use to estimate p for the circuit breaker connecting electric power to the pump in our ATCS system. The data are from similar circuit breakers in similar systems and we need to decide if the information from the various sources can be pooled. If it can be pooled, then we could start with a noninformative prior for p, update with the pooled data, and then use the resulting posterior distribution as the prior distribution for the circuit breaker in our specific system. However, if the data cannot be pooled a more sophisticated analysis is required, as described in a later section.

The data in this example consist of failure counts and demands, and are shown below.

Source	Failures	Demands
1	0	164
2	1	322
3	0	13
4	2	186
5	2	6151
6	0	3264
7	1	4747
8	1	3211
9	4	457
10	4	456
11	3	277

We first examine the data qualitatively, by plotting the credible intervals for each source, based on updating a Jeffreys prior. This is done using Script 12 above, with the data in that script replaced by the data shown above. The lack of overlap of intervals in the caterpillar plot displayed in Figure 4-19 shows that there is apparently significant variability in p from source to source.
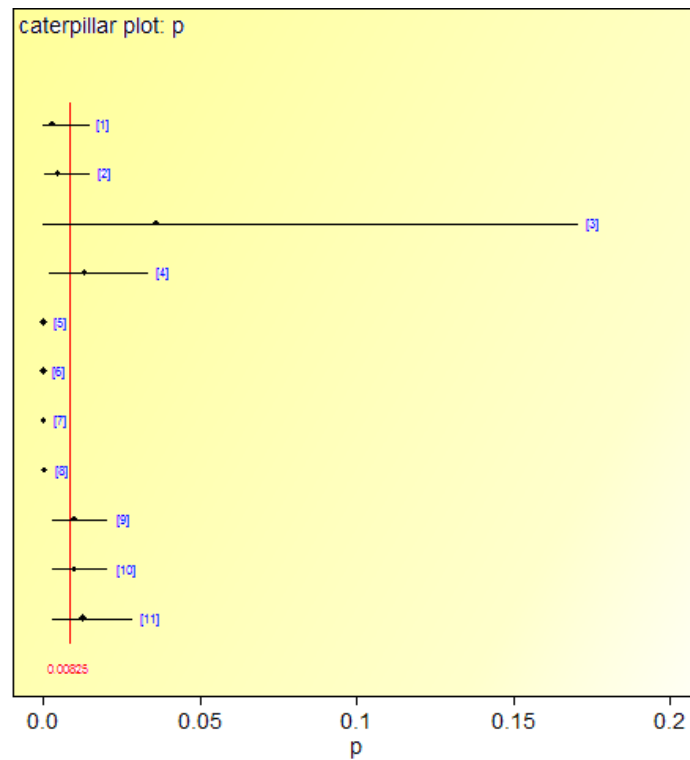
- Figure 4-19. Side-by-side plot of 95% credible intervals for circuit breaker data, illustrating source-to-source variability in p.

We can quantify our judgment about source-to-source variability with the WinBUGS Script 14. Run this script in the usual way and monitor the mean of the p.value node. A mean for the p.value node near 0 or 1 is evidence that the sources should **not** be pooled.

In our example, the mean value is 7.0×10^{-5}, a very small value, providing strong quantitative evidence that the data should **not** be pooled. A Bayesian approach that **preserves** this source-to-source variability is described in Section 4.5.1.

```
model  {
    for (i in 1 : N) {
        x[i] ~ dbin(p[i], n[i])                 # Binomial for number of failures in each source
        p[i] <- p.constant                      # Use this line to test for poolability
        # p[i] ~ dbeta(0.5, 0.5) # Use this line to generate caterpillar plot of credible intervals
        x.rep[i] ~ dbin(p[i], n[i])             # Replicate from posterior predictive distribution
        diff.obs[i] <- x[i] - p[i]*n[i]         # Difference between observed and expected x
        chisq.obs[i] <- pow(diff.obs[i], 2)/(n[i]*p[i]*(1-p[i]))        #Observed chi-square
        diff.rep[i] <- x.rep[i] - p[i]*n[i]     # Difference between replicated and expected x
        chisq.rep[i] <- pow(diff.rep[i], 2)/(n[i]*p[i]*(1-p[i]))        #Replicated chi-square
    }
    p.constant ~ dbeta(0.5, 0.5)                # Jeffreys prior for p.constant, use for poolability test
    chisquare.obs <- sum(chisq.obs[])
    chisquare.rep <- sum(chisq.rep[])
    p.value <- step(chisquare.rep - chisquare.obs)   # Mean of this node should be near 0.5
}
data
x[]  n[]
0    164
1    322
0    13
2    186
2    6151
0    3264
1    4747
1    3211
4    457
4    456
3    277
END
list(N=11)
```

• Script 14. WinBUGS script to generate caterpillar plots and perform quantitative test for poolability of binomial data.

Figure 4-18 showed the DAG that represents Script 14.

4.3.2 Checking the Validity of the Poisson Model

Recall the three underlying assumptions for the Poisson distribution:

- ☑ The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval. The constant of proportionality is denoted by lambda.
- ☑ The probability of simultaneous events in a short interval of time is approximately zero. Thus, common-cause failures are not considered.
- ☑ The occurrence of an event in one time interval does not affect the probability of occurrence in another, non-overlapping time interval.

We have also selected a prior distribution for lambda, and Bayes' Theorem has been used to combine this prior with observed data, producing a posterior distribution for lambda. The probability of observing *future* data can be calculated by averaging the Poisson distribution for the number of events in time t over the posterior distribution for lambda. If this probability is small, then the model is suspect. Perhaps the prior distribution is at fault, or perhaps one of the Poisson assumptions listed above is grossly violated. Further investigation will be required to identify where the breakdown has occurred.

- Example 15. Posterior predictive check for circulating pump failure in the ATCS.

> In Example 3 the prior distribution for the circulating pump was given as a gamma distribution with parameters alpha_{prior} = 1.6 and beta_{prior} = 365000 hours.
> Suppose that the observed data had been 2 failures in 200 days. As a check on the validity of the model find the probability of seeing at least 2 failures in the next 200 days.

Solution

In the WinBUGS script shown below, x.rep represents the predicted number of failures in the next 200 days. The p.value node calculates the probability that x.rep is at least as big as the observed value, x. If the mean of this node is less than about 0.05, a problem may exist with the model, either in the prior distribution or in one or more of the assumptions underlying the Poisson distribution.

```
model  {
x ~ dpois(mean.poisson)                 # Poisson distribution for number of events
x.rep ~ dpois(mean.poisson)             # Replicate value of x from posterior predictive
                                        # distribution
mean.poisson <- lambda*time.hr          # Poisson parameter
time.hr <- time*24                      # Convert days to hours
lambda ~ dgamma(alpha, beta)            # Gamma prior distribution for lambda
p.value <- step(x.rep - x)              # If mean of this node is < 0.05, there is a
}                                       # problem with the model

data
list(x=2, time=200, alpha=1.6, beta=365000)
```

- Script 15. WinBUGS script to generate replicate data from Poisson distribution.
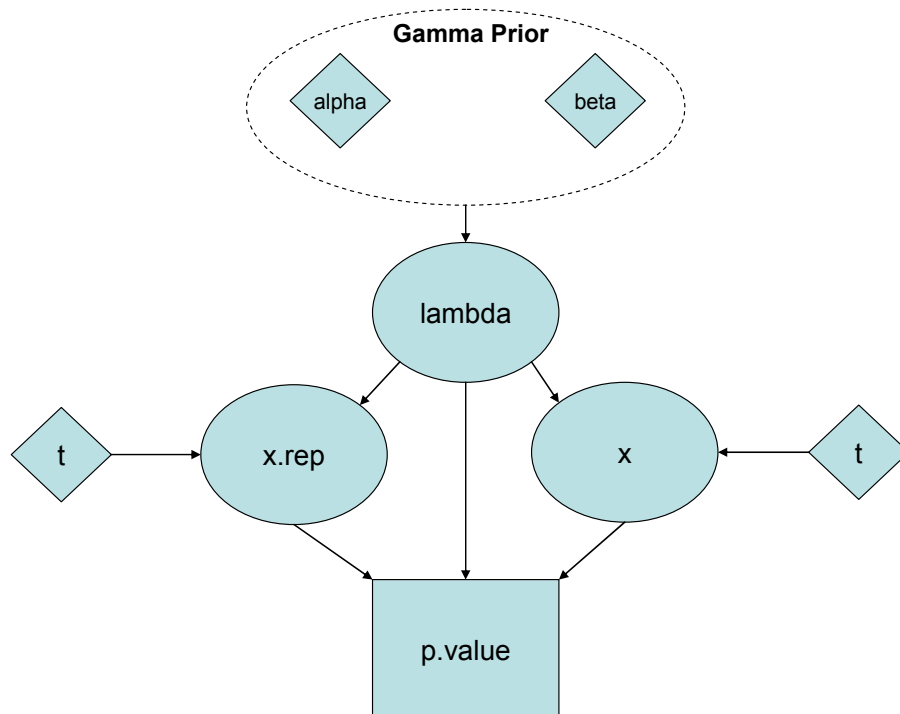
- Figure 4-20. DAG representing Script 15.

Running the script for 100,000 iterations, with 1,000 burn-in iterations for convergence, gives a mean for the p.value node of 0.001, indicating the model is not able to replicate the observed data; it severely

under-predicts the number of failures. At this point, the analyst may decide that the choice of prior distribution was overly optimistic because it puts high probability on small values of lambda, or perhaps an engineering investigation may reveal that the two observed failures were not independent, violating one of the assumptions behind the Poisson distribution. This could be the case, for example, if the cause of the first failure was not completely repaired. However, in this example the probability of seeing 2 or more failure in 200 days is only about 3.5×10^{-4}, suggesting that the prior distribution for lambda is inconsistent with the actual performance of the circulating pump.

• Example 16. Check for time trend in initiating event frequency for heat load on ATCS system.

Consider the following data for an initiating event that places a heat load on the ATCS. The analyst decides to use a Jeffreys prior with the data pooled across the seven years. This results in a posterior mean for lambda of 0.08 and a 90% credible interval of (0.06, 0.099). Investigate the validity of the analyst's approach.

Year	Number of Events	Exposure Time
1	16	14.63
2	10	14.15
3	7	15.75
4	13	17.77
5	9	17.11
6	6	17.19
7	2	17.34

Solution

One of the assumptions underlying the Poisson distribution is that lambda does not change over time. A qualitative way to check this assumption is to calculate a 95% credible interval for each of the seven years, plot them side by side, and look for overlap. If the intervals all overlap with one another, this is evidence that lambda is not varying significantly over time. Note that lambda could exhibit a time trend, perhaps increasing over time due, or it could vary significantly from one year to the next. Either situation could make the simple Poisson likelihood a poor model.

Script 17 (see page 69) can be used to construct a side-by-side plot of the credible intervals for each year (using a Jeffreys noninformative prior), replacing the data with the data for this example. The side-by-side interval plot is shown in Figure 4-21.
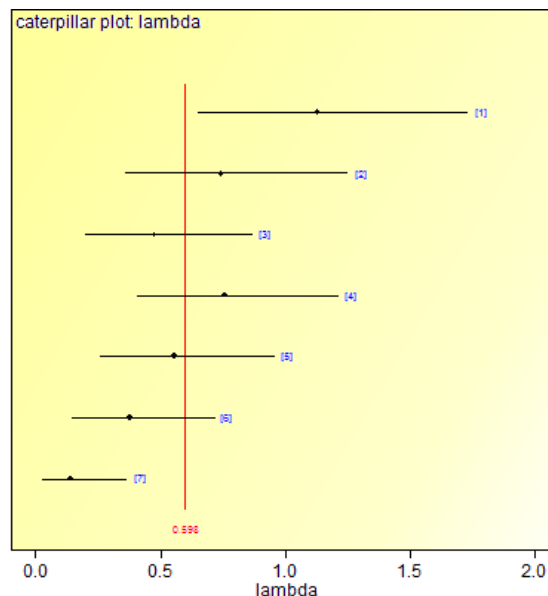
• Figure 4-21. Side-by-side interval plot illustrating decreasing trend in lambda over time.

> **Because WinBUGS can have difficulty generating initial values from the Jeffreys prior, we give it a starting value.**

The WinBUGS Script 16 provides a quantitative measure of how well a model that pools the data can replicate the observed data in each year. A Jeffreys prior is used for the single parameter (lambda.constant) that is estimated. Running the script gives a mean for the p.value node of 0.018. If the model **were good** at replicating the observed data, we would expect the mean of this node to be **near 0.5**. Monitoring the x.rep node shows that the constant-lambda model tends to predict too few failures in early years and too many in later years, suggesting that a trend model might be better at replicating the observed data. We will examine this model in Section 4.4.2.

```
model  {
    for(i in 1:N) {
        x[i] ~ dpois(mu[i])         # Poisson distribution for failures in each source
        x.rep[i] ~ dpois(mu[i])    # Replicate value from posterior predictive distribution
        mu[i] <- lambda[i]*t[i]     # Parameter of Poisson distribution
        lambda[i] <- lambda.constant    # Use this line to test for poolability
        diff.obs[i] <- pow(x[i] - mu[i], 2)/mu[i]
        diff.rep[i] <- pow(x.rep[i] - mu[i], 2)/mu[i]
    }
chisq.obs <- sum(diff.obs[])
chisq.rep <- sum(diff.rep[])
p.value <- step(chisq.rep - chisq.obs)                 # Mean of this node should be near 0.5
lambda.constant ~ dgamma(0.5, 0.0001)                  # Jeffreys prior for lambda
}
data
x[]   t[]
16   14.63
10   14.15
7    15.75
13   17.77
9    17.11
6    17.19
2    17.34
END
list(N=7)
inits
list(lambda.constant=0.001)
```

• Script 16. WinBUGS script to test for poolability of Poisson data.
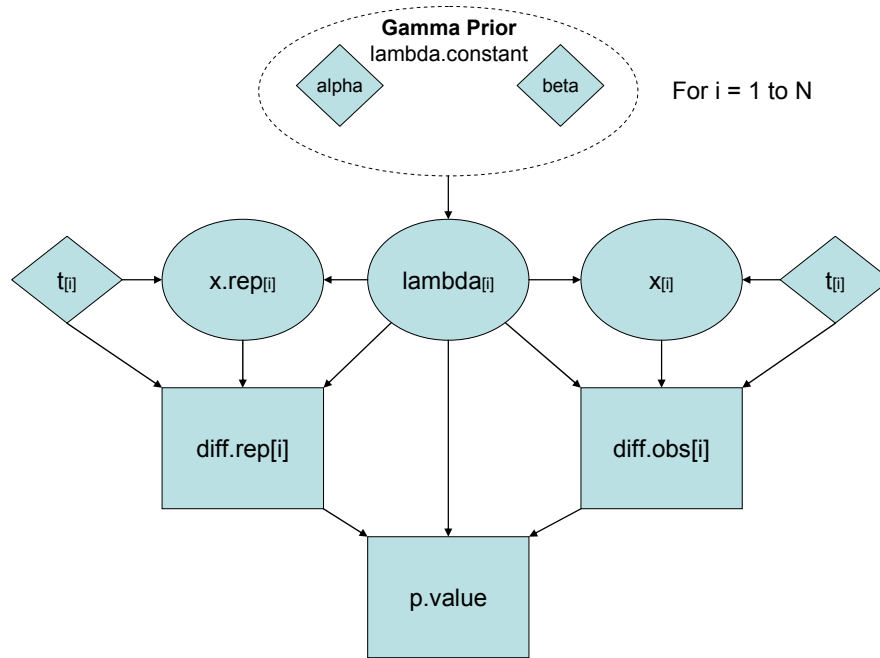
- Figure 4-22. DAG representing Script 16 and Script 18.

- Example 17. Check for source-to-source variability in circuit board failure rate data in the ATCS.

> The following data are available for failure of a particular circuit board in the avionics control package of the ATCS system. These data are from circuit boards in similar applications and we need to decide if the information from the various sources can be pooled. If we can pool the data, then we could start with a noninformative prior for lambda update this with the pooled data and then use the resulting posterior distribution from that update as the prior distribution for the circuit board in our specific system. However if the data cannot be pooled a more sophisticated analysis is required as described in a later section.

The data in this example consist of failure counts and exposure times, and are shown below.

Source	Failures	Exposure Time (hrs)
1	0	87600
2	7	525600
3	1	394200
4	0	87600
5	8	4555200
6	0	306600
7	0	394200
8	0	569400
9	5	1664400
10	1	3766800
11	4	3241200
12	2	1051200

We first examine the data qualitatively, by plotting the credible intervals for each source, based on updating a Jeffreys prior. This is done using the WinBUGS Script 17. The lack of overlap of intervals in the caterpillar plot displayed in Figure 4-24 shows that there is apparently significant variability in lambda from source to source.

```
model  {
    for(i in 1:N) {
        x[i] ~ dpois(mu[i])        # Poisson dist. for number of failures in each source
        mu[i] <- lambda[i]*t[i]              # Parameter of Poisson distribution
        lambda[i] ~ dgamma(0.5, 0.0001)       # Jeffreys prior for caterpillar plot
    }
}
data
x[]  t[]
0    87600
7    525600
1    394200
0    87600
8    4555200
0    306600
0    394200
0    569400
5    1664400
1    3766800
4    3241200
2    1051200
END
list(N=12)
```

• Script 17. WinBUGS script to generate caterpillar plots for lambda with multiple data sources (block data entry).
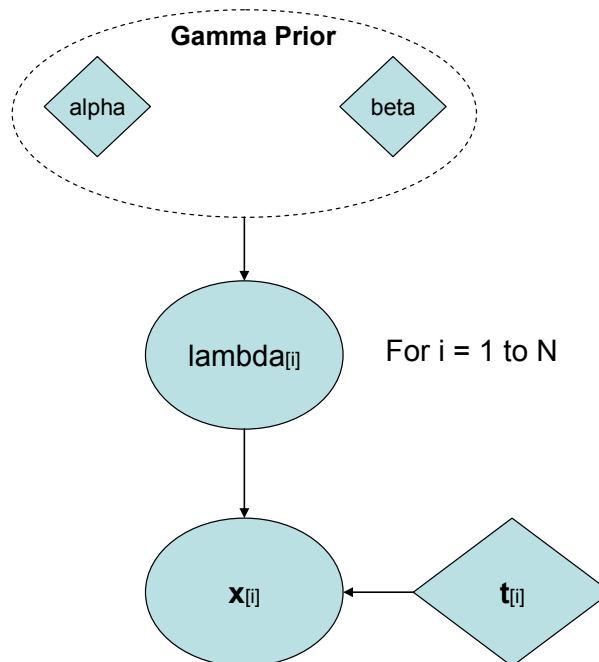
• Figure 4-23. DAG representing Script 17.
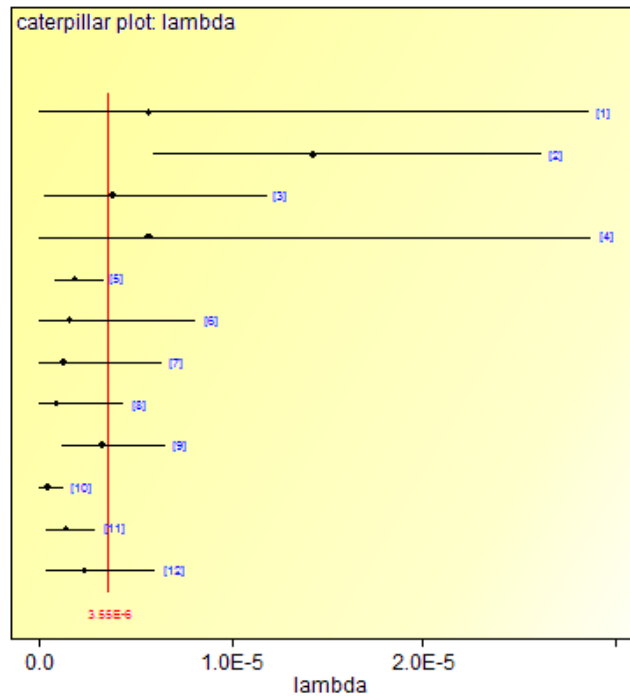
- Figure 4-24. Side-by-side plot of 95% credible intervals for circuit board data, illustrating source-to-source variability in lambda.

We can quantify our judgment about source-to-source variability with the following WinBUGS script. Run this script in the usual way and monitor the mean of the p.value node. A mean near 0 or 1 is evidence that the sources should not be pooled. In our example, the mean value is 0.002, a very small value, providing strong quantitative evidence that the data should **not be pooled**. A Bayesian approach that preserves the source-to-source variability is described in Section 4.5.2.

```
model  {
    for(i in 1:N) {
        x[i] ~ dpois(mu[i])          # Poisson dist. for number of failures in each source
        x.rep[i] ~ dpois(mu[i])      # Replicate value from posterior predictive distribution
        mu[i] <- lambda[i]*t[i]               # Parameter of Poisson distribution
        lambda[i] <- lambda.constant      # Use this line to test for poolability
        # lambda[i] ~ dgamma(0.5, 0.0001)        # Jeffreys prior for waterfall plot
        diff.obs[i] <- pow(x[i] - mu[i], 2)/mu[i]
        diff.rep[i] <- pow(x.rep[i] - mu[i], 2)/mu[i]
    }
chisq.obs <- sum(diff.obs[])
chisq.rep <- sum(diff.rep[])
p.value <- step(chisq.rep - chisq.obs)                  # Mean of this node should be near 0.5
lambda.constant ~ dgamma(0.5, 0.0001)          # Jeffreys prior for lambda
}
data
x[]  t[]
0    87600
7    525600
1    394200
0    87600
8    4555200
0    306600
0    394200
0    569400
5    1664400
1    3766800
4    3241200
2    1051200
END
list(N=12)
inits
list(lambda.constant=0.001)
```

• Script 18. WinBUGS script to generate caterpillar plots and perform quantitative test for poolability of Poisson data.

Refer to Figure 4-22 for the DAG representing Script 18.

4.3.3 Checking the Validity of the Exponential Model

With the exponential distribution, the observed data are a set of times (e.g., times to failure, times to recovery) and have the following assumptions:

☑ The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval. The constant of proportionality is denoted by lambda.

☑ The probability of simultaneous events in a short interval of time is approximately zero.

☑ The occurrence of an event in one time interval does not affect the probability of occurrence in another, non-overlapping time interval.

☑ The random event that is observed is the time to an event.

> **With an assumed exponential likelihood and a Jeffreys prior on lambda, this test of the assumed exponential likelihood cannot be used, as the mean value of the percentile node will be about 0.5 regardless of which aleatory model generates the data. We illustrate another test below that can be used for this situation.**

WinBUGS can be used to generate replicate times from the posterior predictive distribution, and these replicate times can be compared with the observed times to check for model validity. For the exponential distribution, it is possible to compare the sums of the observed and replicated times. The sum of the observed times is a single number, while the sum of the replicated times is a random variable having a distribution. If the Bayesian inference model is valid, we expect the sum of the observed times to be near the center of the distribution for the sum of the replicated times.

- Example 18. Posterior predictive check for observed failure times of ATCS circulating water pumps.

> In Example 5 we had the following times to failure (in hours) for ATCS circulating water pumps: 55707, 255092, 56776, 111646, 11358772, 875209, and 68978. The prior distribution was gamma with alpha = 1.6 and beta = 365000 hours. The sum of the observed failure times is t_{tot} = 12782181 hours.
>
> Use the script below to compare the observed total time with the distribution of the sum of the replicated times.

```
model  {
    for(i in 1:n)  {
        time[i] ~ dexp(lambda)  # Exponential likelihood function for n failure times
        time.rep[i] ~ dexp(lambda)      # Replicate times from posterior predictive
    }                                    # distribution
sum.rep <- sum(time.rep[])
percentile <- step(sum.rep - sum(time[]))          # Mean value should be near 0.5
lambda ~ dgamma(alpha, beta)          # Gamma prior for lambda
}

data
list(time=c(55707, 255092, 56776, 111646, 11358772, 875209, 68978), n=7, alpha=1.6,
beta=365000)
```

- Script 19. WinBUGS script for posterior predictive check of times to failure against exponential likelihood assumption.
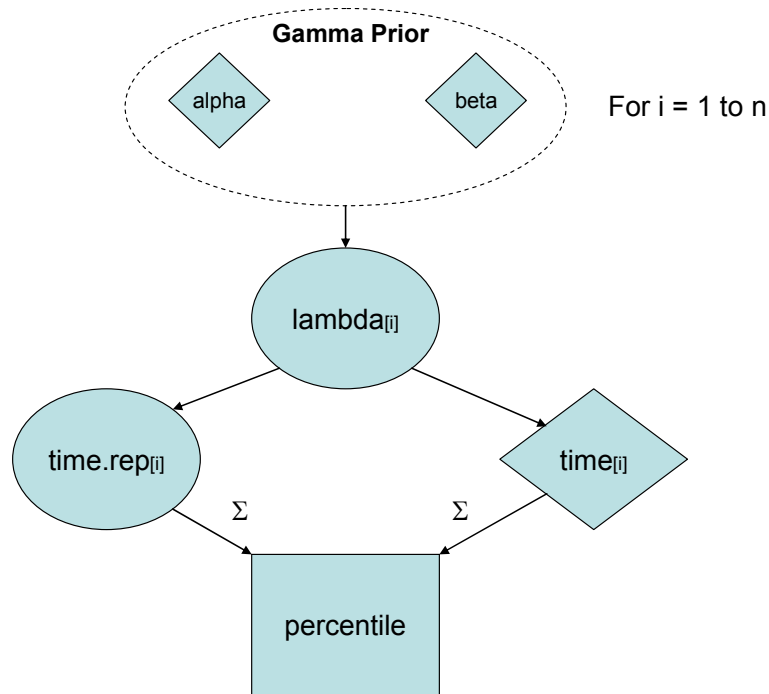
- Figure 4-25. DAG representing Script 19.

Running this script in the usual way and monitoring the percentile node, we see a mean value of 0.36, meaning the observed total time is at the 64th percentile of the posterior predictive distribution for the replicated total time. This value means that our Bayesian inference model is under-predicting times to failure somewhat. The prior distribution may be placing too much weight on small values of lambda, or the exponential likelihood may not be adequate, because lambda is not the same for each of the components for which we have observed a failure time.

To check the influence of the prior distribution, we can use a Jeffreys prior, which gives a posterior distribution that is influenced only by the observed data. However, in this case, we cannot use the test in Script 19 because the mean value of the percentile node will be about 0.5 regardless of which aleatory model generates the data. The script below implements another test, which does not have this failing.

```
model  {
    for (i in 1 : N) {
        time.supp[i] ~ dexp(lambda)                # Exponential dist. for suppression times
        time.supp.ranked[i] <- ranked(time.supp[],i)
        time.rep[i] ~ dexp(lambda)
        time.rep.ranked[i] <- ranked(time.rep[], i)
        F.obs[i] <- 1 - exp(-lambda*time.supp.ranked[i])
        F.rep[i] <- 1 - exp(-lambda*time.rep.ranked[i])
        diff.obs[i] <- pow(F.obs[i] - (2*i-1)/(2*N), 2)
        diff.rep[i] <- pow(F.rep[i] - (2*i-1)/(2*N), 2)
    }
lambda ~ dgamma(0.0001,0.0001)         # Diffuse prior for exponential parameter
CVM.obs <- sum(diff.obs[])
CVM.rep <- sum(diff.rep[])
p.value <- step(CVM.rep - CVM.obs)       # Small value indicates problem with exponential
likelihood
}
data
list(time.supp=c(55707, 255092, 56776, 111646, 11358772, 875209, 68978), N=7)
init
list(lambda=.001)
```

• Script 20. WinBUGS script to test for the assumption of exponential times.
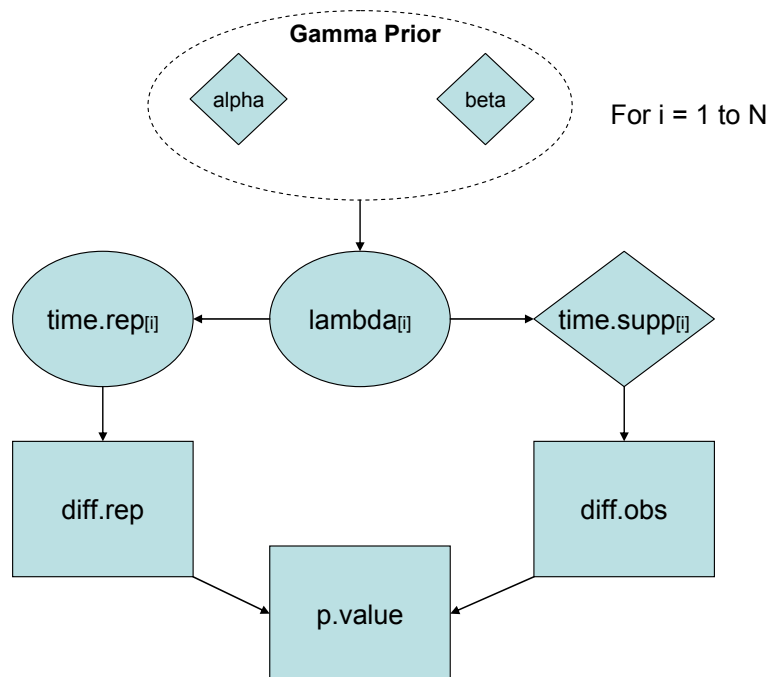
• Figure 4-26. DAG representing Script 20.

Running this script in the usual way and monitoring the mean of the p.value node gives a result of 0.009. Such a **small** value strongly indicates that the exponential model is **not** an **adequate** aleatory model for these observed times. Alternatives to the exponential distribution are covered in Section 4.6.1.

4.4 Time-Trend Models for p and Lambda

Example 13 analyzed valve leakage data over a period of nine years and concluded that p might be increasing with time over that period. In this section, we will see how to develop models in which p and lambda are explicit functions of time, relaxing the assumptions of constant p and constant lambda in the binomial and Poisson distribution, respectively. This introduces new unknown parameters of interest and makes the Bayesian inference significantly more complicated mathematically. However, modern tools such as WinBUGS make this analysis only slightly more complicated than the single-parameter cases analyzed earlier.

4.4.1 Time-Trend Model for p in the Binomial Distribution

The standard way to allow for p to vary in time (monotonically) is by letting f(p) = a + bt, where f(p) is a user-defined function of p, a and b are constants, and t is time. If b = 0, then this reduces to the constant-p model discussed earlier. If b > 0, then p is increasing with time. If b < 0, p is decreasing with time. The new parameters to be estimated are a and b. It is common to use diffuse prior distributions for these parameters. Of course, if prior information is available about the parameters a and b, an informative prior can be used, but this is not common in practice. In this subsection, the following assumptions apply:

- ☑ The aleatory model of the world contains a single unknown parameter.
- ☑ The prior information is homogeneous and is known with certainty.
- ☑ The observed data are homogeneous and are known with certainty.
- ☑ There are two possible outcomes of each demand typically denoted success and failure.
- ☑ There is a probability of failure on each demand, denoted herein as p, that **may vary** from one demand (or time, if demands are tracked over time) to another.

The WinBUGS Script 21 below carries out inference with a commonly used function for p – we used the logit function, logit(p(i)), in the script. The logit is given as $\ln\left(\dfrac{p(i)}{1-p(i)}\right)$ where p(i) should be between a value of 0 and 1. Other functions that may be used in place of the logit are shown in the script (e.g., probit and complementary log-log), but are commented out. In many cases, changing the f(p) function will not make any practical difference in the inference results.

```
model  {
    for (i in 1:N) {
        x[i] ~ dbin(p[i], n[i])        # Binomial distribution for failures in each year
        logit(p[i]) <- a + b*i          # Use of logit() link function for p[i]
        #probit(p[i]) <- a + b*i         # Use of probit() link function for p[i]
        #cloglog(p[i]) <- a + b*i  # Use of complementary loglog function for p[i]
        x.rep[i] ~ dbin(p[i], n[i])  # Model validation section
        diff.obs[i] <- pow(x[i] - n[i]*p[i], 2)/(n[i]*p[i])
        diff.rep[i] <- pow(x.rep[i] - n[i]*p[i], 2)/(n[i]*p[i])
    }
logit(p[10]) <- a + b*10                 # Used to predict p in 10th year
a~dflat()                                # Diffuse prior for a
b~dflat()                                # Diffuse prior for b
chisq.obs <- sum(diff.obs[])
chisq.rep <- sum(diff.rep[])
p.value <- step(chisq.rep - chisq.obs) # A small value for this node indicates a problem
}

Data
x[]  n[]
4    52
2    52
3    52
1    52
4    52
3    52
4    52
9    52
6    52
END
list(N=9)

Inits
list(a=1, b=0)
list(a=-1, b=0.1)
```

• Script 21. WinBUGS script for modeling a time trend in p.

For i = 1 to N
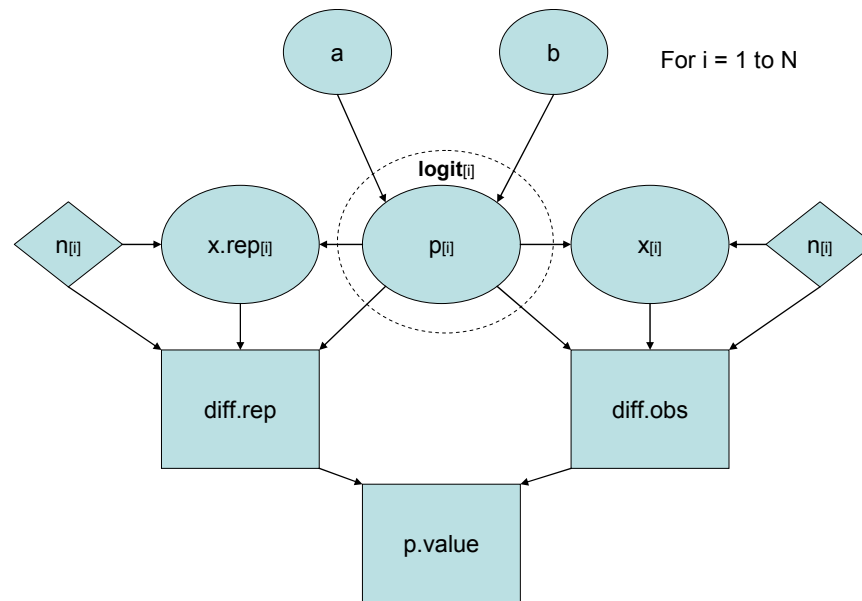
- Figure 4-27. DAG representing Script 21.

Because this is a more **complicated** model, we will run **two** chains, starting at different points, as an aid in deciding when convergence to the posterior distribution has occurred. We do this by selecting two chains before pressing the compile button during the specification step. Also, we must give starting values for each chain, as WinBUGS cannot generate initial values of a and b from the flat prior distributions we have used. These are listed in the Inits portion of the script and are loaded one chain at a time. Following the initial values load for the two chains, the *generate inits* button can be clicked to generate any other initial values required. Running 1,000 iterations gives the following history plots for the parameters a and b. It appears from these plots that we have convergence within the first 1,000 iterations.
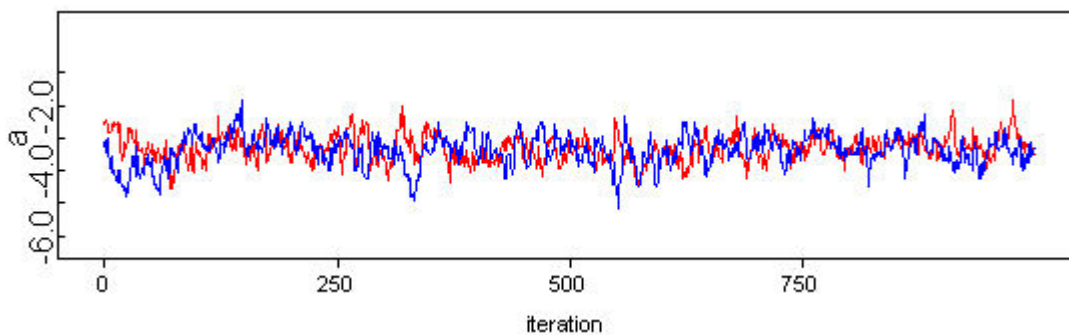
- Figure 4-28. Plot of values of first 1,000 values for "a" parameter, illustrating convergence.
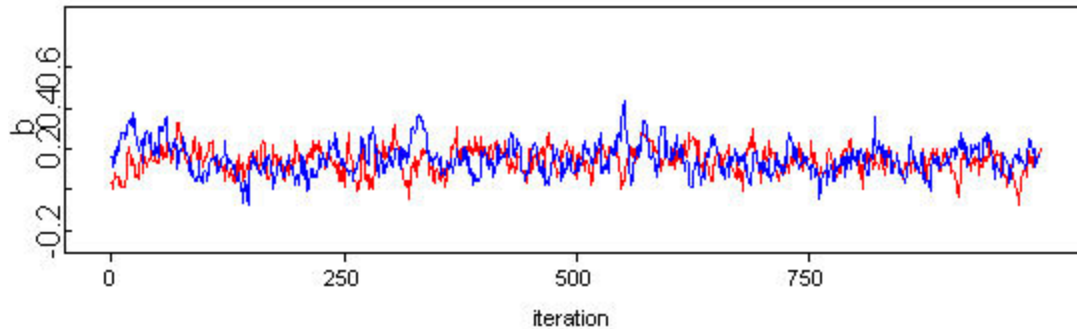
- Figure 4-29. Plot of values of first 1,000 values for "b" parameter, illustrating convergence.

> **For convergence, the red line in the BGR plot should be close to 1.0, and the blue and green lines should be stable.**

We can also plot the BGR diagnostic, since we have run more than one chain. The plots for both parameters are indicative of convergence.
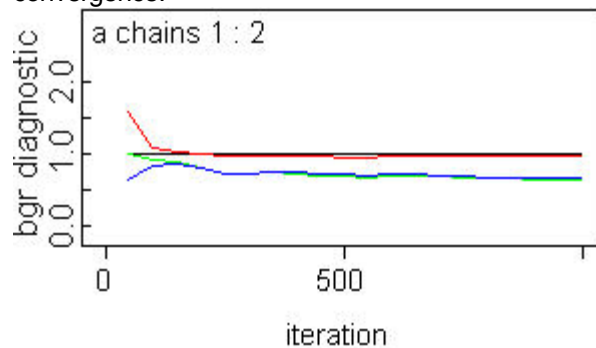
- Figure 4-30. BGR diagnostic for "a" parameter, indicating convergence.
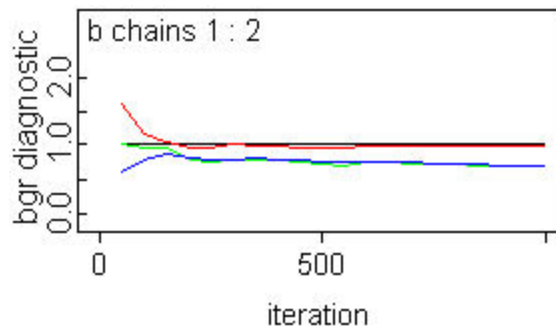
- Figure 4-31. BGR diagnostic for "b" parameter, indicating convergence.

Now that we are confident of convergence, we run another 100,000 iterations to estimate parameter values. Examining the posterior density for the "b" parameter will help us judge the significance of any trend that might be present: if the posterior distribution is mostly to the right of zero, this indicates an increasing trend, and vice versa if the posterior distribution is mostly to the left of zero. By monitoring the b node, we obtain a posterior probability of at least 0.975 that b > 0, suggesting a statistically significant increasing trend in p. The plot of the posterior distribution for b below shows this graphically.
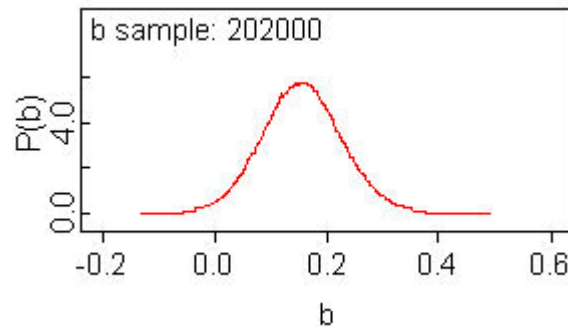
- Figure 4-32. Posterior distribution for "b" parameter.   Values below zero are very unlikely, suggesting an increasing trend in p over time.

We can also quantify the ability of this model to replicate the observed data by monitoring the mean of the p.value node. As before, a mean near 0.5 indicates good replicative ability. In this example, the mean is 0.47. Comparing this with the value of about 0.18 obtained for a constant-p model, we see that a model in which p increases with time is much better at replicating the observed data.
Finally, we can use this model to estimate p in the next year (year 10). This is given by node p[10] in the script above, and would be what we would use in a PRA. Monitoring this node, we find a posterior mean of 0.15 and a 90% interval of (0.085, 0.22). Compare this with the estimates from the constant-p model of 0.08 for the mean and (0.06, 0.099) for the 90% interval. These results are shown in Figure 4-33.
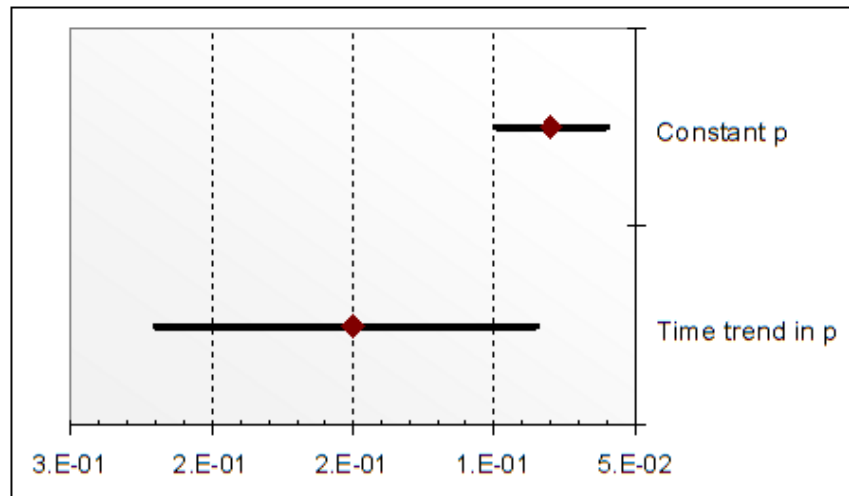
- Figure 4-33. Comparison of time trend versus constant probability results for the binomial model.

Figure 4-34 shows the 95% credible intervals for each year, based on the trend model for p.
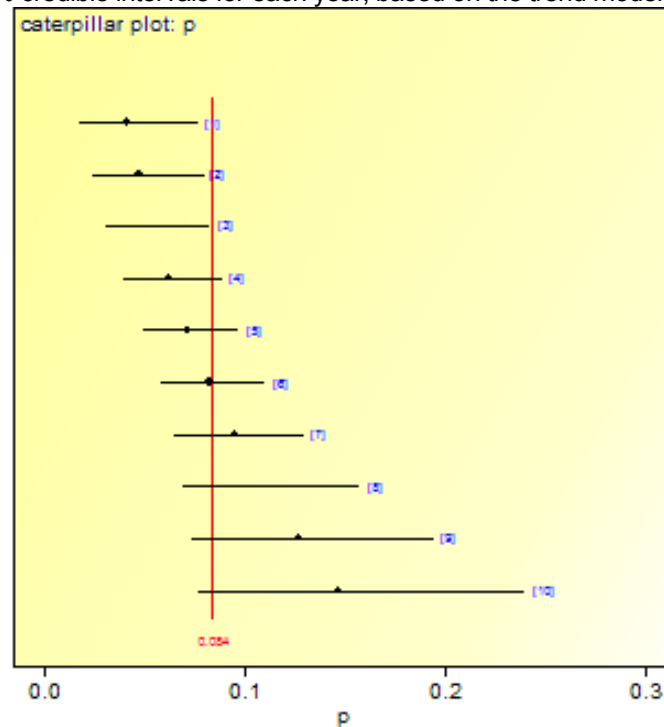
- Figure 4-34. Plot of 95% interval for p in each year based on a trend model in which p is increasing with time.

4.4.2 Time-Trend Model for Lambda in the Poisson Distribution

A standard way to allow for lambda to vary in time (monotonically) is by letting f(lambda) = a + (b t), where f(lambda) is a user-defined function of lambda, a and b are constants, and t is time. If b = 0, then this reduces to the constant-lambda model discussed earlier. If b > 0, then lambda is increasing with time. If b < 0, lambda is decreasing with time. The new parameters to be estimated are a and b. It is common to use diffuse prior distributions for these parameters. Of course, if prior information is available, an informative prior can be used, but this is not common in practice. WinBUGS Script 22 carries out inference with a commonly-used function of lambda[loglinear, where log(lambda) = a + b t]. In this subsection, the following assumptions apply:

☑   The aleatory model of the world contains a single unknown parameter.
☑   The prior information is homogeneous and is known with certainty.
☑   The observed data are homogeneous and are known with certainty.
☑   The probability of an event in a small time interval depends on the length of the interval and the time at which the event occurs. In other words, lambda is a function of time.
☑   The probability of simultaneous events in a short interval of time is approximately zero.

```
model   {
for(i in 1:N)        {
  x[i] ~ dpois(mu[i])               # Poisson dist. for number of failures in each source
  x.rep[i] ~ dpois(mu[i])           # Replicate value from posterior predictive distribution
  mu[i] <- lambda[i]*t[i]           # Parameter of Poisson distribution
  log(lambda[i]) <- a + b*i         # Loglinear model for lambda
  diff.obs[i] <- pow(x[i] - mu[i], 2)/mu[i]
  diff.rep[i] <- pow(x.rep[i] - mu[i], 2)/mu[i]
  }
log(lambda[8]) <- a + b*8         # Used to predict lambda in 8th year
chisq.obs <- sum(diff.obs[])
chisq.rep <- sum(diff.rep[])
p.value <- step(chisq.rep - chisq.obs)    # Mean of this node should be near 0.5
a~dflat()                                  # Diffuse priors for a and b
b~dflat()
 }
data
x[]  t[]
16  14.63
10  14.15
7    15.75
13  17.77
9    17.11
6    17.19
2    17.34
END
list(N=7)
Inits
list(a=0.1, b=0)
list(a=0.1, b=-0.01)
```

• Script 22. WinBUGS script for modeling time trend in lambda.

For i = 1 to N
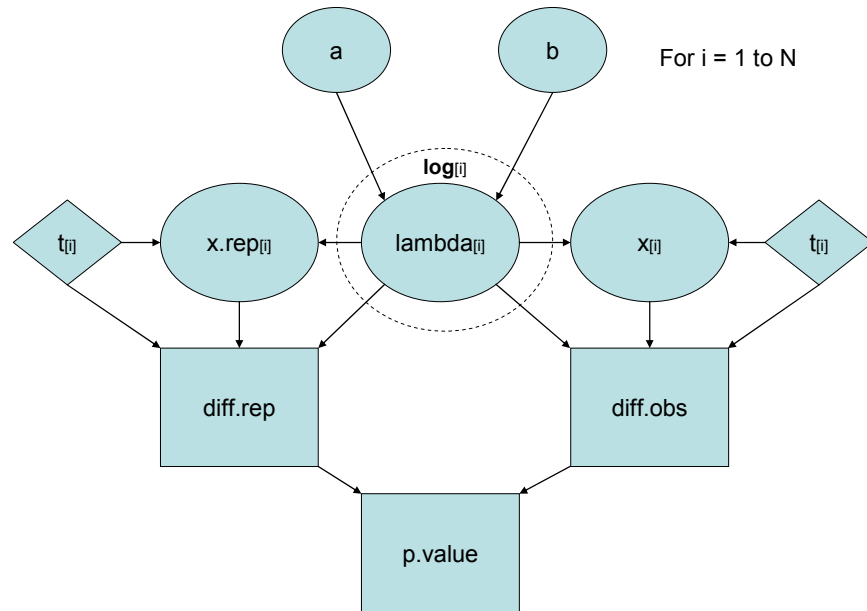
- Figure 4-35. DAG representing Script 22.

Because this is a more complicated model, we will run two chains, starting at different points, as an aid in deciding when convergence has taken place. Also, we must give starting values for each chain. These are listed in the "Inits" portion of the script. Running 1,000 iterations gives the following history plots for the parameters **a** and **b**. It appears from these plots that we have convergence within the first 1,000 iterations.
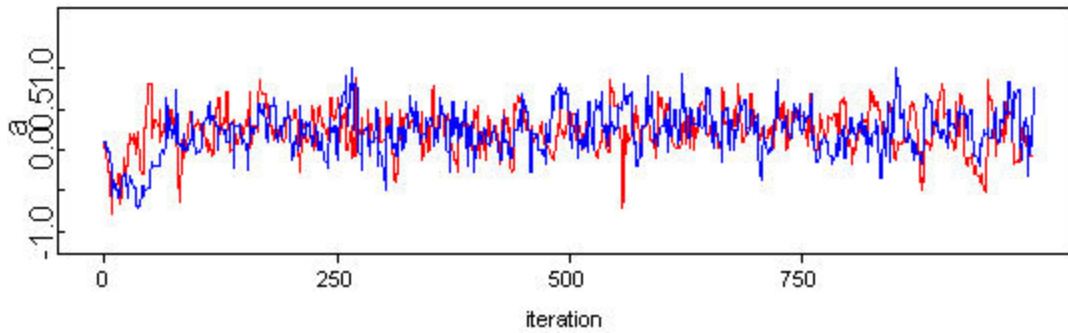
- Figure 4-36. Plot of values of first 1,000 values for "a" parameter, illustrating convergence.
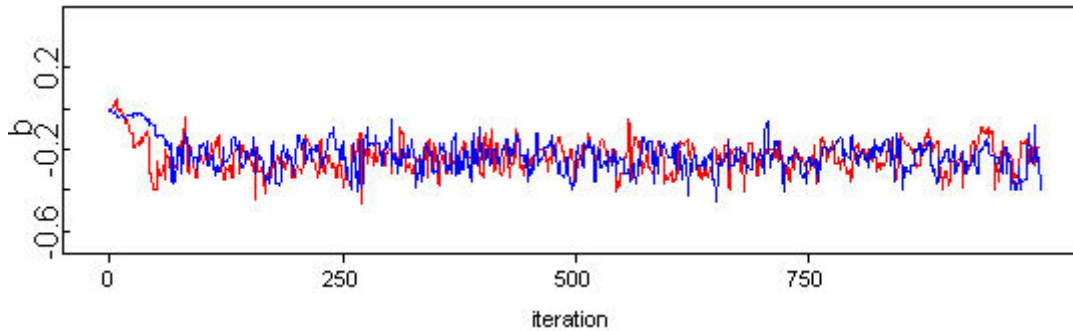
- Figure 4-37. Plot of values of first 1,000 values for "b" parameter, illustrating convergence.

We can also plot the BGR diagnostic, since we have run more than one chain. The plots for both parameters are indicative of convergence.

> **For convergence, the red line in the BGR plot should be close to 1.0, and the blue and green lines should be stable.**
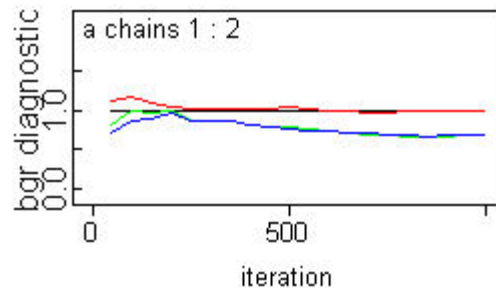
- Figure 4-38. BGR diagnostic for "a" parameter, indicating convergence.
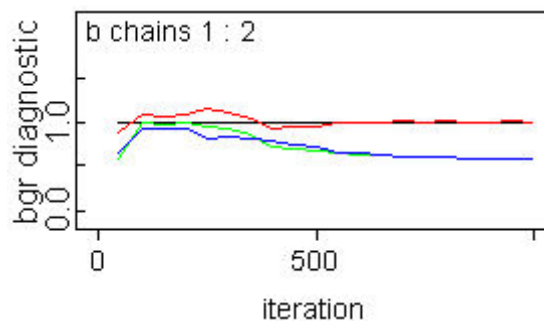
- Figure 4-39. BGR diagnostic for "b" parameter, indicating convergence.

Now that we are confident of convergence, we run another 100,000 iterations to estimate parameter values. Examining the posterior density for the "b" parameter will help us judge the significance of any trend that might be present: if the posterior distribution is mostly to the right of zero, this indicates an increasing trend, and vice versa if the posterior distribution is mostly to the left of zero. By monitoring the b node, we obtain a posterior probability of at least 0.975 that b < 0, suggesting a statistically significant decreasing trend in lambda. The plot of the posterior distribution for b below shows this graphically.
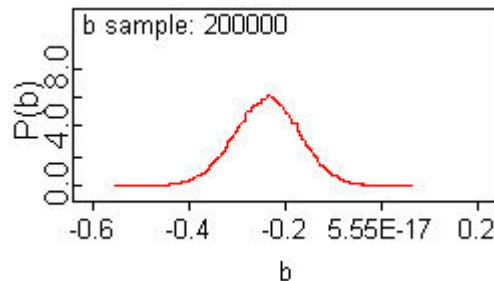
- Figure 4-40. Posterior distribution for "b" parameter. Values above zero are very unlikely, suggesting a decreasing trend in lambda over time.

We can also quantify the ability of this model to replicate the observed data by monitoring the mean of the p.value node. As before, a mean near 0.5 indicates good replicative ability. In this example, the mean is 0.46. Comparing this with the value of about 0.018 obtained for a constant-lambda model, we see that a model in which lambda decreases with time is much better at replicating the observed data. Finally, we can use this model to estimate lambda in the next year (year 8). This is given by node lambda[8] in the script above, and would be what we would use in a PRA. Monitoring this node, we find a posterior mean of 0.21 and a 90% interval of (0.11, 0.34). Compare this with the estimates from

the constant-lambda model of 0.55 for the mean and (0.45, 0.68) for the 90% interval. The figure below shows the 95% credible intervals for each year, based on the trend model for lambda.
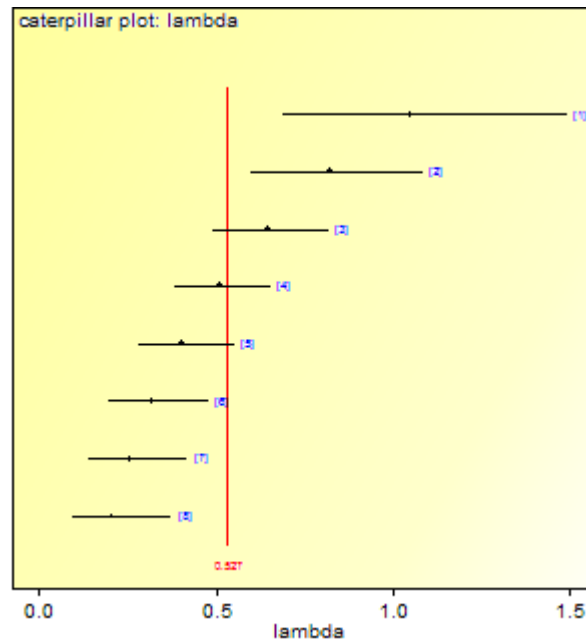
- Figure 4-41. Plot of 95% interval for lambda in each year based on a trend model in which lambda is decreasing with time.

4.5 Population Variability Models

4.5.1 Population Variability Model for p in the Binomial Distribution

In Example 14, we had data from multiple sources that we wanted to use to estimate p for the circulating pump circuit breaker in the ATCS. The data were from similar breakers in similar systems, and we wanted to develop a prior distribution that captured both the central estimate from these sources and the variability in p indicated by the data. Recall that the **assumptions from** Example 14 were:

- ☑ There are two possible outcomes of each demand typically denoted success and failure.
- ☑ There is a constant probability of failure (typically in PRA; typically success in reliability engineering) on each demand, denoted herein as p.
- ☑ The outcomes of earlier demands do not influence the outcomes of later demands.
- ☑ The model contains a single unknown parameter.
- ☑ The prior information is homogeneous and is known with certainty. In this section, we will **relax** the assumption of homogeneity.
- ☑ The observed data are homogeneous and are known with certainty. In this section, we will **relax** the assumption of homogeneity.

The conclusions from Example 14 indicated that the mean of the "p.value" node was a very small value, providing strong quantitative evidence that the data should **not** be pooled. Consequently, we need a Bayesian approach that **preserves** this source-to-source variability. Such analysis is referred to as hierarchical Bayesian analysis, because the prior distribution is developed in stages or hierarchies. We first specify a prior to represent the variability in the data sources (the "population variability curve", or PVC) and then specify a second prior describing the epistemic uncertainty in the parameters of the first-stage prior. The analysis is quite complicated mathematically, but WinBUGS makes the analysis straightforward.

> **Sensitivity to the functional form of the PVC should always be checked.**

We must begin by choosing a functional form for the PVC. Often a conjugate form is chosen (e.g., beta distribution if the parameter is p in a binomial distribution), but this is not a requirement. In fact, in cases where there is extremely high variability in p, choosing a conjugate form can lead to suspect results. In such cases, a distribution with a heavier tail, such as the lognormal, may be a better choice. Or it may be better to develop a weighted-average prior, where the analyst specifies subjective weights for each data source (see Section 4.8.4). For our example, we will take the PVC to be a beta distribution with parameters alpha and beta.

As the second stage in the hierarchy, we must specify a second-stage prior distribution for alpha and beta (alpha and beta are called **hyperparameters** and the second-stage prior is referred to as a

> **Because of potential convergence and other numerical difficulties, population variability analysis should always be reviewed by an expert.**

hyperprior). This is usually done by choosing independent, diffuse distributions for both parameters. Choosing hyperpriors that are diffuse, yet that avoid numerical difficulties can be a bit of an art, and this is one of the reasons why analysis of population variability should always be reviewed by an expert analyst.

The WinBUGS Script 23 is used to perform the PVC analysis. Node p.avg will represent variability in p across the sources. It will be the prior distribution for inference about the circuit breaker failure probability in the ATCS, and will be updated with data collected from operation or testing of our system. Note that this script also calculates a node called p.value, which is used to check the ability of the model to replicate the observed data.

```
model   {
for(i in 1:N)        {
     x[i] ~ dbin(p[i], n[i])              # Binomial model for number of events in each source
     p[i] ~ dbeta(alpha, beta)            # First-stage beta prior
     x.rep[i] ~ dbin(p[i], n[i])          # Replicate value from posterior predictive
distribution
     #Generate inputs for Bayesian p-value calculation
     diff.obs[i] <- pow(x[i] - n[i]*p[i], 2)/(n[i]*p[i]*(1-p[i]))
     diff.rep[i] <- pow(x.rep[i] - n[i]*p[i], 2)/(n[i]*p[i]*(1-p[i]))
     }
p.avg ~ dbeta(alpha, beta)               # Average beta population variability curve
# Calculate Bayesian p-value
chisq.obs <- sum(diff.obs[])
chisq.rep <- sum(diff.rep[])
p.value <- step(chisq.rep - chisq.obs)   # Mean of this node should be near 0.5
# Hyperpriors for beta first-stage prior
alpha ~ dgamma(0.0001, 0.0001)
beta ~ dgamma(0.0001, 0.0001)
}
data
x[]  n[]
0    164
1    322
0    13
2    186
2    6151
0    3264
1    4747
1    3211
4    457
4    456
3    277
END
list(N=11)
inits
list(alpha=0.5, beta=200)          # Chain 1
list(alpha=2, beta=100)  # Chain 2
```

• Script 23. WinBUGS script for analyzing population variability in p.
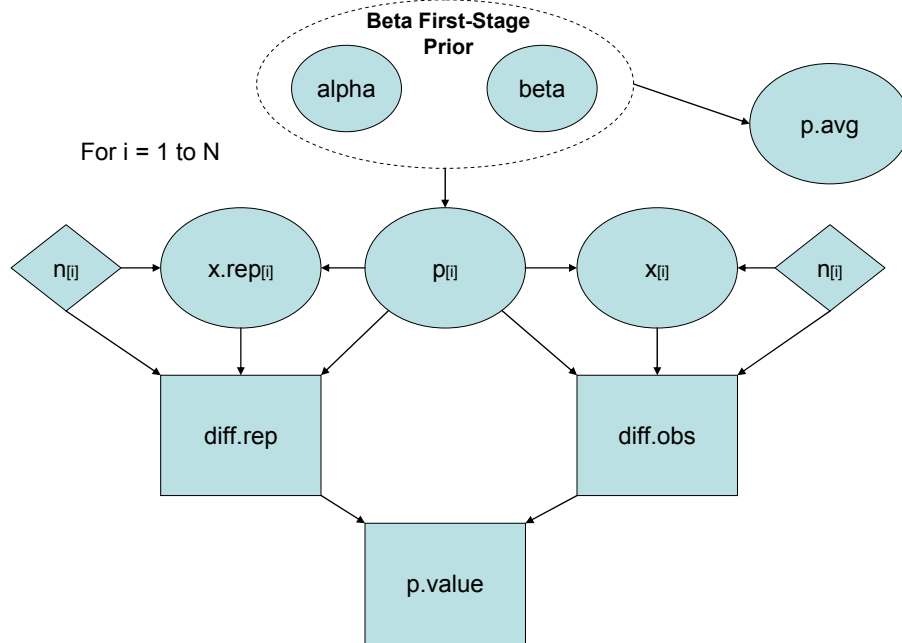
- Figure 4-42. DAG representing Script 23.

The initial values for each chain are estimates distributed around what are felt to be likely values of alpha and beta. Often, accurate estimates are not needed. However, in some cases, more care will have to be used in picking initial values, and more than two chains may be needed. Such problems will require expert assistance, but should not be too common in practice.

We run this script for 2,000 iterations, and check for convergence by examining history plots and BGR diagnostics for each parameter. These are shown below, and indicate convergence within the first 2,000 iterations, so we will discard these as burn-in samples.
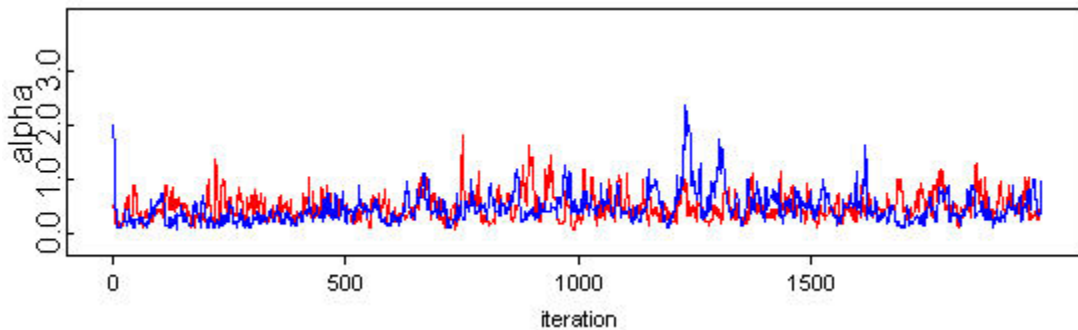
- Figure 4-43. Plot of first 2,000 values of alpha, indicating convergence.
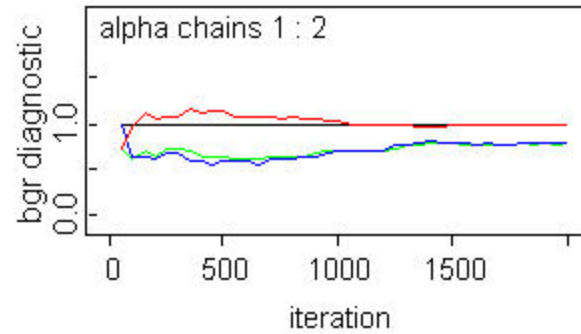
- Figure 4-44. BGR diagnostic for alpha, indicating convergence.
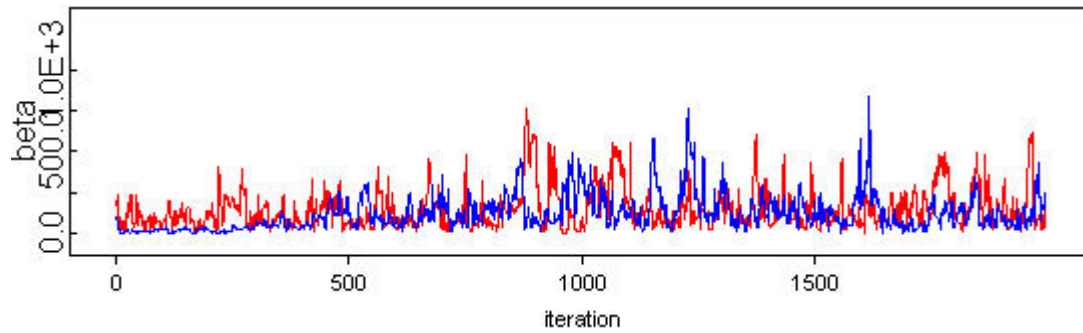
- Figure 4-45. Plot of first 2,000 values of beta, indicating convergence.
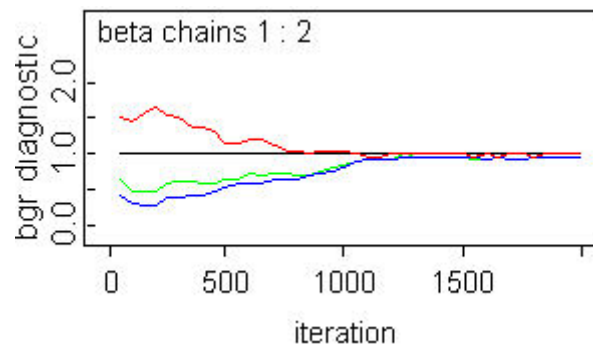
- Figure 4-46. BGR diagnostic for beta, indicating convergence.
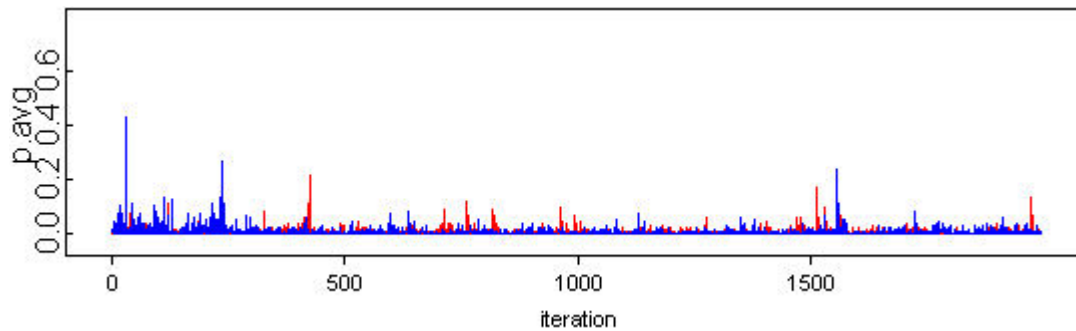
- Figure 4-47. Plot of first 2,000 values of p.avg, indicating convergence.
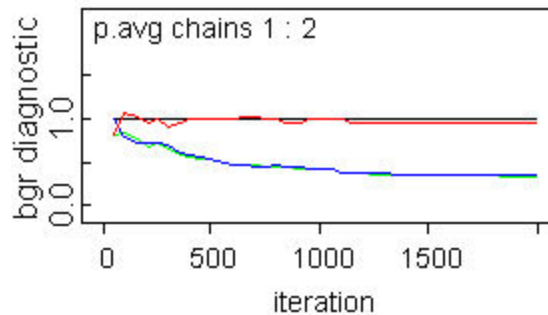
- Figure 4-48. BGR diagnostic for p.avg, indicating convergence.

This script runs slower than earlier scripts we have used, so we may want to make parameter estimates with less than the usual 100,000 iterations. Let us try 10,000 additional iterations and examine the Monte Carlo error for each parameter to judge if enough samples have been taken to accurately estimate each parameter. The results are listed below.

	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
alpha	0.5	0.2635	0.007467	0.1872	0.4438	1.01	2001	20000
beta	149.9	125.8	3.676	22.23	116.2	387.4	2001	20000
p.avg	0.005016	0.01399	1.219E-4	1.767E-6	0.001569	0.0197	2001	20000

A heuristic to decide if more samples are needed is that the Monte Carlo error should be no more than about 5% of the mean value. This is satisfied for all three parameters, so we can stop after 10,000 iterations (12,000 including burn-in). The posterior mean of p.avg is 0.005, with a 90% credible interval of $(1.8 \times 10^{-6}, 0.020)$. This is a very wide range of variability.

Note that the distribution of node p.avg is not a beta distribution, despite what one might be led to believe by looking at Script 23. It is a weighted average of beta distributions, with the weights provided by the posterior distribution of alpha and beta. Thus, it cannot be written in closed form. However, if desired, it can be approximated by a distribution of known form to aid in further analysis, although this is not necessary in WinBUGS. Let us explore these "approximation" options, assuming that the observed data for the circuit breaker in our system is **1 failure in 403 demands** on the breaker.

Option 1: Replace average PVC with beta distribution having same mean and variance
This option might be taken by an analyst who desires a conjugate prior for further updates with binomial data, and who wants to use simple algebra to estimate the parameters of the beta conjugate prior. The algebra was illustrated in Section 4.2.5.1.3. The resulting estimates for alpha and beta are 0.12 and 24.3, respectively. The corresponding 90% credible interval is $(5.5 \times 10^{-13}, 0.029)$. Because of the small value of alpha, this distribution places considerable weight on very small values of p. Updating this conjugate prior with our observed data of 1 failure in 403 demands gives a beta posterior distribution with $alpha_{post} = 1.12$ and $beta_{post} = 426.3$. The posterior mean is 2.6×10^{-3} and the posterior 90% credible interval is $(1.8 \times 10^{-4}, 7.5 \times 10^{-3})$.

Option 2: Replace average PVC with lognormal distribution having same mean and 95th percentile
This option might be taken by an analyst who is concerned about the high weight that the beta conjugate prior obtained in Option 1 places on very small values of p. If zero failures were observed, this would lead to a posterior mean that is too low, perhaps by a large enough amount to distort the risk importance of the circuit breaker in the overall PRA. We will use WinBUGS to perform the Bayesian inference, since the lognormal distribution is not a conjugate prior, and the appropriate lines from Script 9 will be used to convert the mean and upper bound to mu and tau, the parameters needed by WinBUGS to describe the lognormal distribution. The new script is shown as Script 24. Note that the solution for sigma involves a quadratic equation, and the roots for this equation will not be real unless the ratio of the mean to the 95th percentile is greater than about 0.258. In our case this ratio is 0.250,

so we need to adjust the value of "upper" until this constraint is satisfied. This is achieved when "upper" < 0.0194, so we replace the estimated 95th percentile of 0.02 above with a value of 0.019.

```
model  {
x ~ dbin(p, n)              # Binomial distribution for number of failures
p ~ dlnorm(mu, tau)        # Lognormal prior distribution for p

#Calculate mu and tau from lognormal mean and upper bound
tau <- pow(sigma, -2)
sigma <- (2*1.645 + sqrt(4*pow(1.645, 2) + 8*log(mean/upper)))/2
mu <- log(mean) - pow(sigma, 2)/2
}
data
list(x=1,n=403, mean=0.005, upper=0.019)
```

- Script 24. WinBUGS script to replace average PVC with lognormal prior using mean and upper bound, and to update prior with observed data of 1 failure in 403 demands.
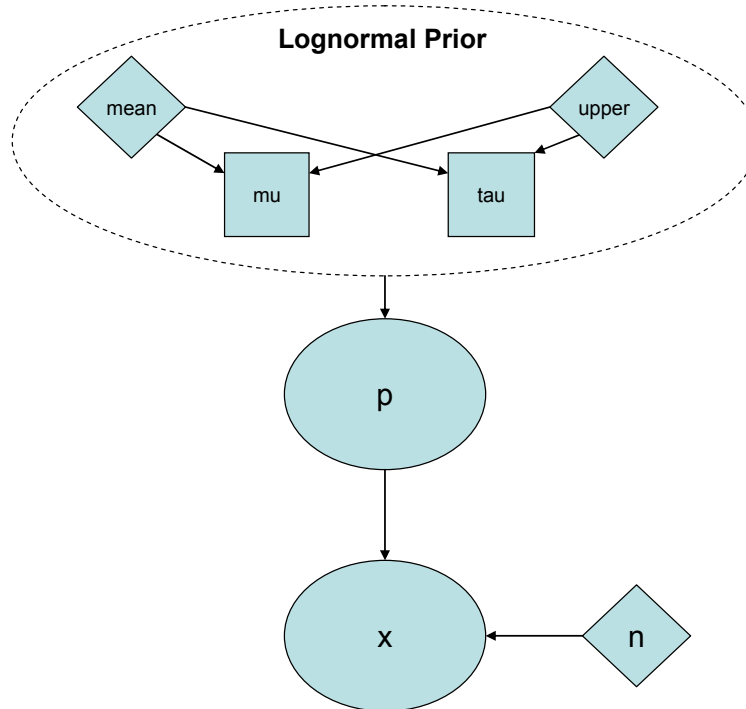
- Figure 4-49. DAG representing Script 24.

Sensitivity to the functional form of the PVC should always be checked.

Running Script 24 in the usual way gives a posterior mean for p of 2.2×10^{-3} and a 90% credible interval of (2.3×10^{-4}, 6.2×10^{-3}), similar to the posterior results under Option 1 above.

Option 3: Use WinBUGS to update average PVC with observed data (exact approach)
This option involves **no approximations** and is straightforward: we just add a new data line (with the 1 failure in 403 demands, and changing N to 12) to Script 23 and rerun the analysis, monitoring node p[12]. The results are shown below.

	mean	sd	MC_error	val5.0pc	median	val95.0pc
p[12]	0.002747	0.002271	1.962E-5	3.212E-4	0.002155	0.007173

In Figure 4-50, the comparison of the means and 90% credible intervals for the three options described in this section are plotted. Also, the figure shows the results from the original Example 14 calculation where no population variability was assumed. As can be seen in the figure, the two fitted results are comparable to the exact calculation. However, the "no variability assumption" calculation has a lower mean value (as compared to the exact calculation) and the uncertainty on the probability is greatly underestimated.
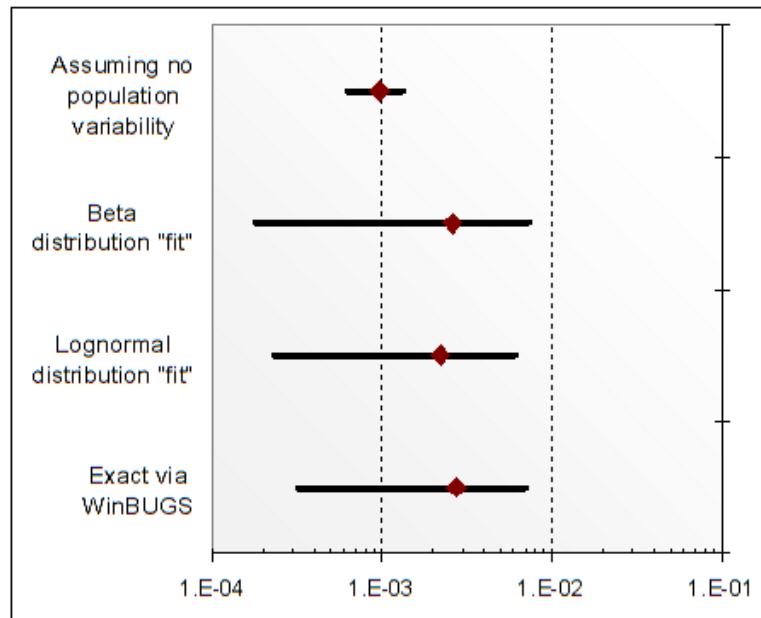
• Figure 4-50. Comparison of binomial Example 14 results for the exact calculation, two fitted (to the exact) results, and the no population variability results.

4.5.2 Population Variability Model for Lambda in the Poisson Distribution

In Example 17, we had data from circuit boards similar to one of concern in the avionics control portion of our ATCS system. In that example, we decided that we could not pool the data from these sources to generate a prior distribution for lambda. Recall the following assumptions for the example:
- ☑ The aleatory model of the world contains a single unknown parameter.
- ☑ The prior information is homogeneous and is known with certainty. In this section, we will relax the assumption of homogeneity.
- ☑ The observed data are homogeneous and are known with certainty. In this section, we will relax the assumption of homogeneity.
- ☑ The probability of an event in a small time interval depends only on the length of the interval.
- ☑ The probability of simultaneous events in a short interval of time is approximately zero.

Because of potential convergence problems and other numerical difficulties, population variability analysis should always be reviewed by an expert.

In this section, we need to develop a prior distribution that captures both the central estimate from the different data sources and the variability in lambda indicated by the data. The approach is the same as for p in the binomial distribution described in Sec 4.5.1. We begin by specifying a first-stage prior describing the source-to-source variability in lambda, and then a second-stage prior (hyperprior) for the parameters of the first-stage prior.

We will use a conjugate gamma distribution as the first-stage prior, although other functional forms can be used. We will use independent diffuse priors for the parameters of the first-stage gamma prior. The WinBUGS Script 25 is used to carry out the analysis.

```
model  {
    for(i in 1:N) {
        x[i] ~ dpois(mu[i])         # Poisson dist. for number of failures in each source
        mu[i] <- lambda[i]*t[i]     # Parameter of Poisson distribution
        lambda[i] ~ dgamma(alpha, beta)        # First-stage gamma prior
        x.rep[i] ~ dpois(mu[i])    # Replicate value from posterior predictive distribution
        diff.obs[i] <- pow(x[i] - mu[i], 2)/mu[i]      # Inputs to calculate Bayesian p-value
        diff.rep[i] <- pow(x.rep[i] - mu[i], 2)/mu[i]
    }
lambda.avg ~ dgamma(alpha, beta)        #Overall avg. population variability for lambda
chisq.obs <- sum(diff.obs[])                # Calculate Bayesian p-value
chisq.rep <- sum(diff.rep[])
p.value <- step(chisq.rep - chisq.obs)     # Mean of node should be near 0.5
alpha <- pow(CV, -2)       #Reparameterize in terms of mean and coefficient of variation
beta <- alpha/mean
CV ~ dexp(1)                             # Maximum entropy hyperprior for coefficient of variation
mean ~ dgamma(0.0001, 0.0001)           # Diffuse hyperprior on mean
}

data
x[]  t[]
0    87600
7    525600
1    394200
0    87600
8    4555200
0    306600
0    394200
0    569400
5    1664400
1    3766800
4    3241200
2    1051200
END
list(N=12)

inits
list(CV=1, mean=0.00001)
list(CV=2, mean=0.000001)
```

- Script 25. WinBUGS script for analyzing population variability in lambda with Poisson likelihood.
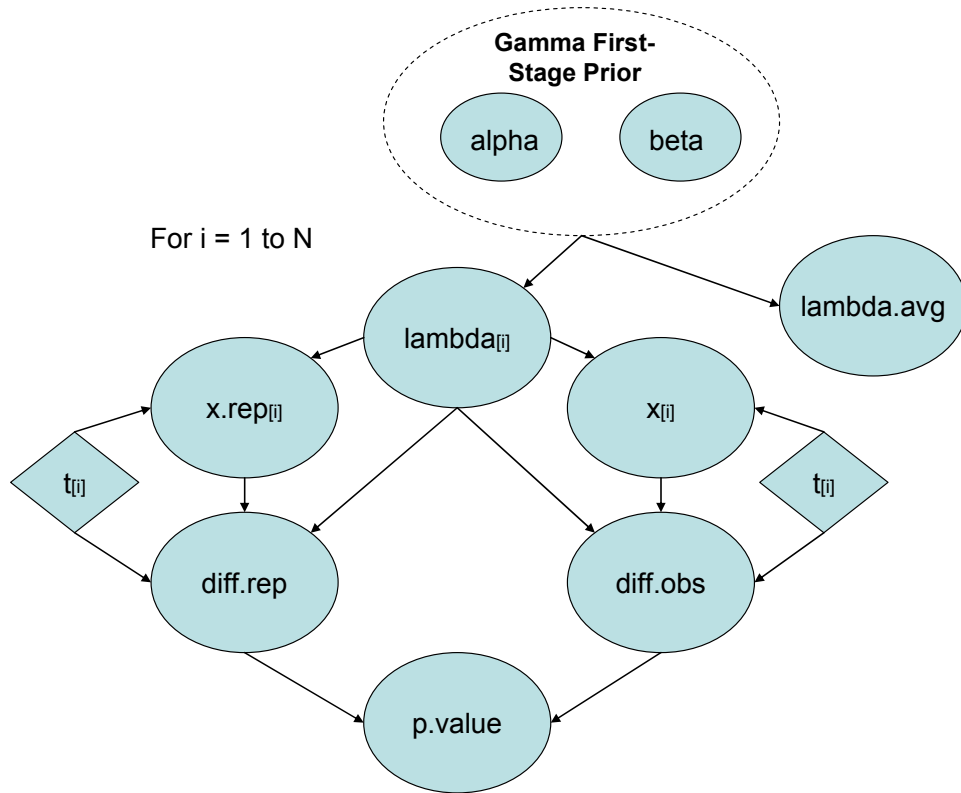
- Figure 4-51. DAG representing Script 25.

The initial values for each chain are estimates distributed around what are felt to be likely values of CV and the mean. Often, accurate estimates are not needed. However, in some cases, more care will have to be used in picking initial values, and more than two chains may be needed. Such problems will require expert assistance, but should not be too common in practice.

> **Note that Script 25 changes the parameters from alpha and beta to different parameters to improve numerical performance. This re-parameterization is an example of where expert consultation may be necessary in complex problems.**

We run this script for 1,000 iterations, and check for convergence by examining history plots and BGR diagnostics for each parameter. These are shown below, and indicate convergence within the first 1,000 iterations, so we will discard these as burn-in samples.
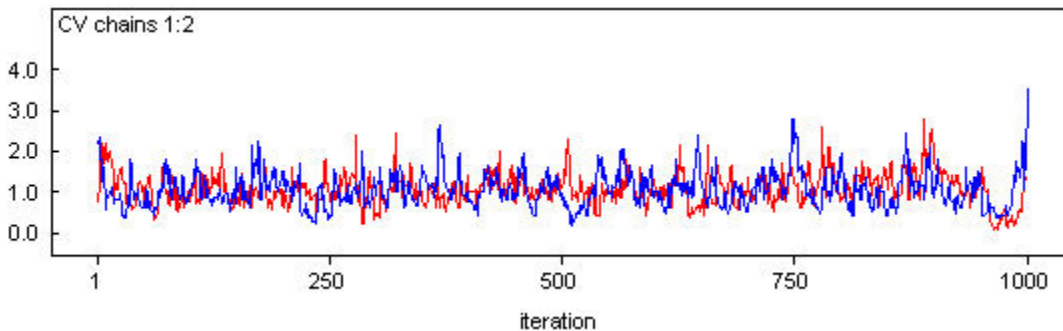
- Figure 4-52 Plot of first 1,000 values of CV, indicating convergence.
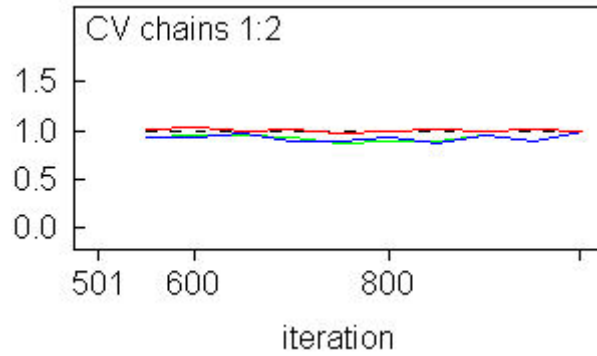
- Figure 4-53 BGR plot for CV, indicating convergence.
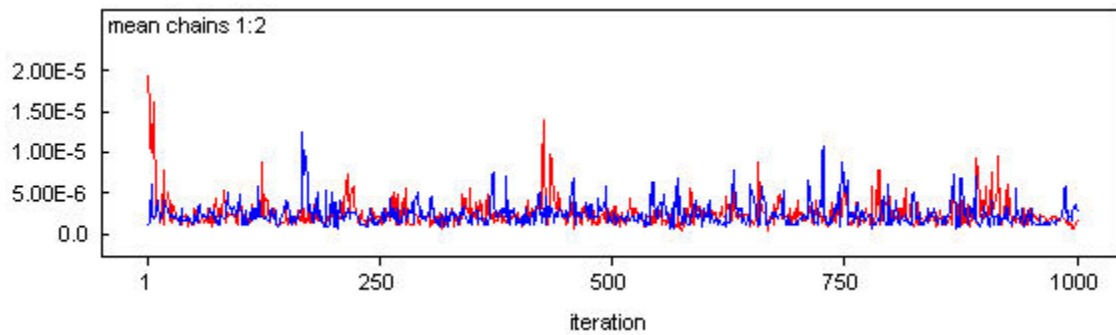
- Figure 4-54 Plot of first 1,000 values of mean parameter, indicating convergence.
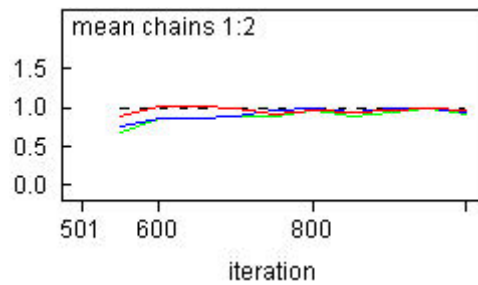
- Figure 4-55. BGR plot for mean parameter, indicating convergence.
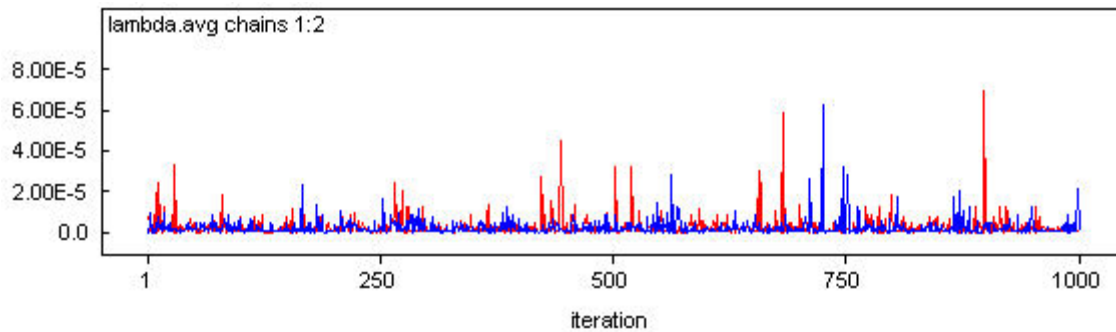
- Figure 4-56. Plot of first 1,000 values of lambda.avg, indicating convergence.
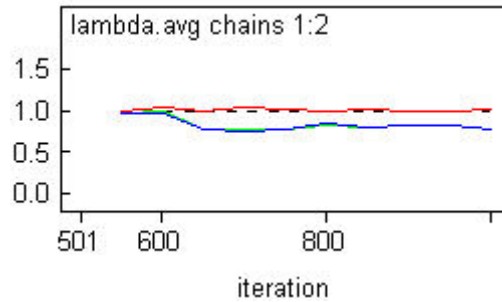
- Figure 4-57. BGR plot for lambda.avg, indicating convergence.

The sampling is much faster with a gamma first-stage prior distribution than for a beta distribution, so WinBUGS runs much faster than it did above when we were estimating a hierarchical model for p in the binomial distribution. Running 100,000 iterations gives a posterior mean for lambda.avg of 3.2×10^{-6}/hour, with a 90% credible interval of $(1.7 \times 10^{-9}, 1.2 \times 10^{-5})$.

Note that the distribution of node lambda.avg is not a gamma distribution, despite what one might be led to believe by looking at Script 25. It is a weighted average of gamma distributions, with the weights provided by the posterior distribution of the hyperparameters, CV and the mean. Thus, it cannot be written in closed form. However, if desired, it can be approximated by a distribution of known form to aid in further analysis, although this is not necessary in WinBUGS. To update the average PVC with additional data, add another line to the data block, change N to 13, and rerun the script. Then, monitoring lambda[13] will give the posterior distribution for the circuit board in the ATCS.

The p.value for this model, which captures source-to-source variability in lambda, is 0.48, compared with 0.002 for the model that pools the data. Thus, the population variability model is **much better** able to replicate the data for the 12 sources under consideration.

4.5.3 Population Variability Models for CCF Parameters
In Example 7, we estimated the alpha-factors and MGL parameters for homogeneous failure data. The assumptions from that calculation were:
- ☑ There are multiple (1 to i =3) possible outcomes (single failure, two failures, three failures).
- ☑ There is a constant probability of the ith outcome.
- ☑ The outcomes of earlier trials do not influence the outcomes of later trials.
- ☑ The prior information is homogeneous and is known with certainty. In this section, we will **relax** the assumption of homogeneity.
- ☑ The observed data are homogeneous and are known with certainty. In this section, we will **relax** the assumption of homogeneity.

In this section, we examine the case where is variability from source to source. Analogous to the treatment of the binomial and Poisson cases above, we will start with a first-stage prior that describes the variability in the CCF parameters (the alpha factors) from source to source. We will then use a second-stage prior (hyperprior) to describe the uncertainty in the parameters of the first-stage prior. An average PVC will then be calculated for the CCF parameters.

• Example 19. Estimating CCF parameters with population variability included.

> The data in the table below are failure counts for a component group of size 3 from 10 sources. Find the average PVC for the alpha factors and MGL parameters.

Source #	n_1	n_2	n_3
1	95	12	1
2	86	4	13
3	75	21	0
4	81	3	0
5	80	12	1
6	79	6	18
7	79	5	20
8	75	3	0
9	78	21	1
10	97	9	2

The WinBUGS Script 26 is used to carry out the analysis. The first-stage Dirichlet prior has three parameters (hyperparameters). Independent diffuse second-stage priors (hyperpriors) are placed on each of these three hyperparameters. Running this script in the usual way, being careful to check for convergence, gives the following results for the average alpha factors and MGL parameters.

	mean	sd	MC_error	val5.0pc[1]	val95.0pc
alpha.avg[1]	0.8507	0.08525	3.023E-4	0.6914	0.9614
alpha.avg[2]	0.1074	0.07363	2.616E-4	0.01793	0.2476
alpha.avg[3]	0.04185	0.04788	1.605E-4	5.612E-4	0.1357
beta.avg	0.1493	0.08525	3.023E-4	0.03856	0.3086
gamma.avg	0.2801	0.2374	8.653E-4	0.00548	0.7538

---

[1] "val5.0pc" is the 5th percentile while "val95.0pc" is the 95th percentile.

```
model   {
     for(i in 1:K) {
          n[i,1:group.size] ~ dmulti(alpha[i,1:group.size ], N[i])
          alpha[i, 1:group.size] ~ ddirch(theta[1:group.size])
          N[i] <- sum(n[i, 1:group.size])

          #Calculate MGL parameters (Table A-2 in NUREG/CR-5485)
          beta[i] <- alpha[i,2] + alpha[i,3]
          gamma[i] <- alpha[i,3]/(alpha[i,2] + alpha[i,3])
     }
theta[1] ~ dunif(0,100)
theta[2] ~ dunif(0,100)
theta[3] ~ dunif(0,100)
alpha.avg[1:group.size] ~ ddirch(theta[1:group.size])     #Average PVC
beta.avg <- alpha.avg[2] + alpha.avg[3]  #Monitor these nodes for average MGL values
gamma.avg <- alpha.avg[3]/(alpha.avg[2] + alpha.avg[3])
 }
data
list(K=10, group.size=3)
n[,1]  n[,2]  n[,3]
95    12    1
86    4     13
75    21    0
81    3     0
80    12    1
79    6     18
79    5     20
75    3     0
78    21    1
97    9     2
END
inits
list(theta=c(1,1,1))
list(theta=c(2,2,2))
```

• Script 26. WinBUGS script for modeling population variability in CCF parameters.
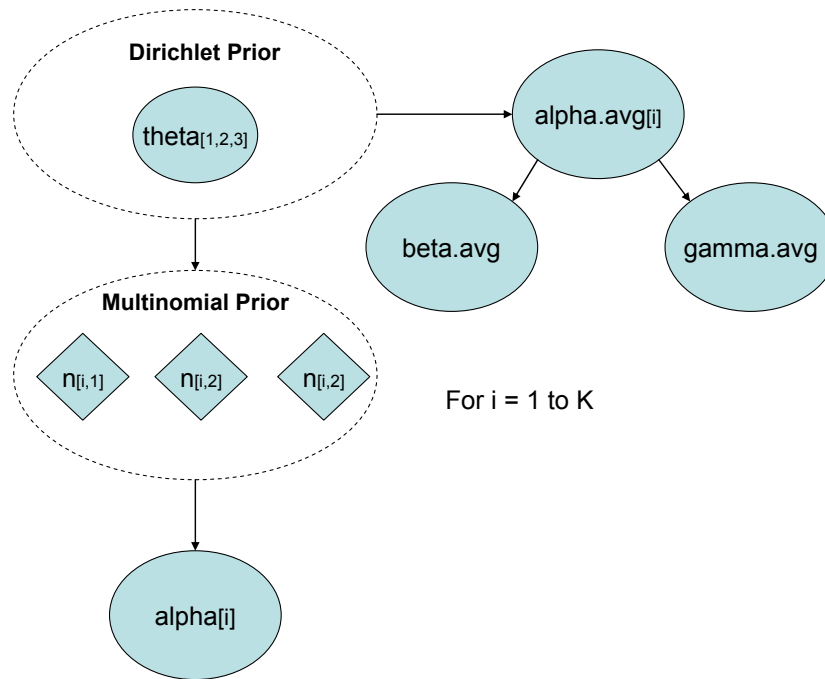
- Figure 4-58. DAG representing Script 26.

4.6 Modeling Time-to-Failure or Other Durations

There are cases in which the simplest model for time duration, the exponential distribution, is not adequate in the sense that the Bayesian inference model with an exponential likelihood function cannot replicate the observed data well. In these cases, more complicated aleatory models may need to be used. The mathematics of the Bayesian inference become more complex, but again tools like WinBUGS makes the analysis straightforward.

The assumptions from the exponential analysis were:
- ☑   The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval. The constant of proportionality is denoted by lambda.
- ☑   The probability of simultaneous events in a short interval of time is approximately zero.
- ☑   The occurrence of an event in one time interval does not affect the probability of occurrence in another, non-overlapping time interval.
- ☑   The random event that is observed is the time to an event.

- Example 20. Test for appropriateness of exponential model for time durations.

> The following times (in minutes) to suppress a fire in a vehicle assembly building have been collected:
>
> 1.6, 1.8, 13.8, 17.2, 19.2, 24.4, 30.2, 39.1, 49.1, 61.2.
>
> Is the exponential distribution a good choice as a likelihood function?

Solution

We cannot use Script 19 to do a posterior predictive check unless we have an informative prior for lambda. Running Script 20 gives a p.value of 0.54, suggesting that an exponential aleatory model is

adequate. However, another approach is to examine certain alternative distributions which reduce to the exponential distribution if their shape parameter equals one. Therefore, we can perform Bayesian inference for one of these alternatives, and if the posterior distribution for the shape parameter is centered at one, we can conclude that the exponential model is adequate.

4.6.1 Alternatives to Exponential Distribution for Random Durations
There are three commonly used distributions that are alternative aleatory models to the exponential distribution:
1. Gamma
2. Weibull
3. Lognormal

There are other less commonly used possibilities, which we will not treat here.

4.6.1.1 Gamma Distribution as Likelihood for Random Durations—As we saw earlier when we used the gamma distribution as a prior for lambda, two parameters are required to specify the distribution, so now we must perform Bayesian inference for both alpha and beta.

> **In this section, the gamma and lognormal distributions are used as likelihood functions (i.e., aleatory models) for an observed time, instead of as prior distributions for a parameter in an aleatory model.**

We illustrate replacing the exponential model with a gamma model by using the times from Example 20. In Script 27, we have specified a **gamma** distribution as the **aleatory** model (i.e., likelihood) for each fire suppression time, and we have placed independent diffuse priors on alpha and beta; if information about alpha or beta were available, an informative prior could be used. Changed assumptions when using the gamma model (instead of exponential) are:

☑ The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval, but the constant of proportionality is a function of time.

```
model  {
    for(i in 1:n)  {
        time[i] ~ dgamma(alpha, beta)          # Gamma likelihood function for n times
    }
alpha ~ dgamma(0.0001, 0.0001)                # Diffuse priors for alpha and beta
beta ~ dgamma(0.0001, 0.0001)  }

data
list(time=c(1.6, 1.8, 13.8, 17.2, 19.2, 24.4, 30.2, 39.1, 49.1, 61.2), n=10)

inits
list(alpha=1, beta=0.1)
list(alpha=0.5, beta=1)
```

• Script 27. WinBUGS script for Bayesian inference of random durations using gamma likelihood.
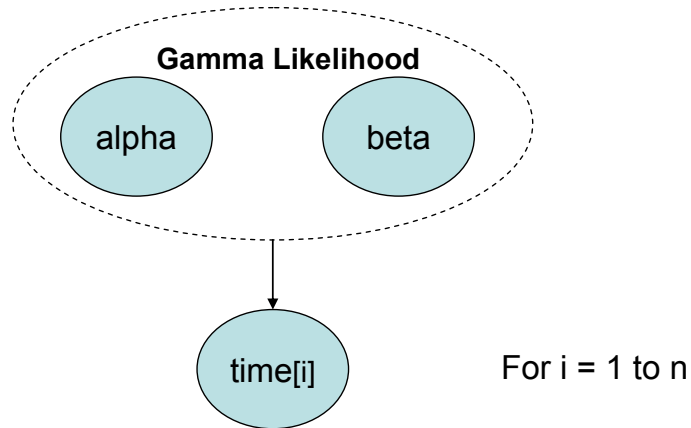
- Figure 4-59. DAG representing Script 27.

Running 1,000 iterations and checking alpha and beta for convergence using the history and BGR plots as we have done in earlier sections, we see that the model converges very rapidly. We discard the first 1,000 iterations and run another 100,000 iterations to estimate parameter values, obtaining the posterior distribution for alpha shown below. If alpha = 1, then the gamma distribution reduces to the exponential distribution, so this graph, with a median of about 1, suggests the gamma distribution does not provide a better model than the exponential distribution, agreeing with the conclusion based on the p.value from Script 20.

> **With relatively few observed times, it can be difficult to discriminate between the exponential distribution and alternative models.**
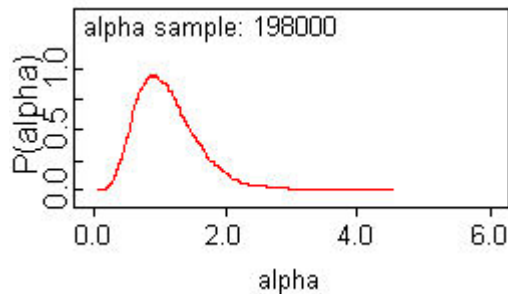
- Figure 4-60. Posterior distribution for alpha indicates that values near 1 are likely, in which case gamma distribution reduces to exponential distribution.

- Example 21. Inference for gamma distribution as aleatory model for circulating pump failure times in ATCS system

> Consider now the following set of repair times for the circulating pump in the ATCS (in hours):
> 0.8, 1.7, 0.5, 11.4, 0.1, 5.7, 3.0, 1.1, 2.5, 0.04.

Solution
Running Script 20 with the data gives a p.value of 0.62, which suggests that an exponential model may be adequate. We will use Script 27 to model a gamma likelihood function for these repair times. Proceeding in the usual way, we obtain the posterior distribution for the shape parameter (alpha) shown in Figure 4-61.

As can be seen, most of this distribution is to the **left** of 1 indicating that an exponential distribution may not be a good aleatory model for this set of data. We will consider how to decide between these conflicting indications in Section 4.6.2.
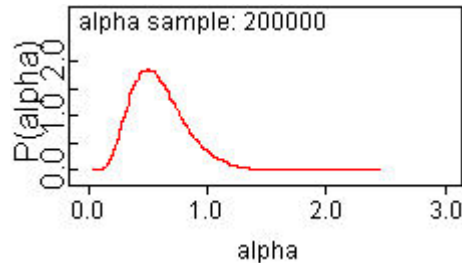
- Figure 4-61. Posterior distribution of alpha for Example 21.

When alpha is less than one, as is the case in this example, the rate at which repair is occurring is a decreasing function of time; the longer one goes without repair, the lower the conditional probability of successful repair in the next time interval. Conversely, when alpha is greater than one, the repair rate increases with time. If the observed variable were a time to failure, then these same conclusions would apply to the failure rate. Thus, the gamma distribution allows us to relax the assumption of constant rate of occurrence that was one of the assumptions behind the exponential distribution.

4.6.1.2 Weibull Distribution as Likelihood for Random Durations—The Weibull distribution is another two-parameter aleatory model for random durations. Like the gamma distribution, it has a shape parameter, which we will denote as alpha in this section. If alpha = 1, the Weibull distribution reduces to the exponential distribution, just as did the gamma distribution in the previous section. When alpha is less than (greater than) one, the rate of occurrence is decreasing (increasing) with time, analogously to the gamma distribution. Changed assumptions when using the Weibull model (instead of exponential) are:

☑ The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval, but the constant of proportionality is a function of time.

- Example 22. Inference for Weibull distribution as aleatory model for circulating pump failure times in the ATCS.

> We will carry out Bayesian inference for the times in Example 21 under the assumption that the aleatory model is a Weibull (instead of exponential) distribution with parameters called alpha and scale. The WinBUGS Script 28 is used for this analysis with diffuse priors on the Weibull parameters.

```
model  {
    for(i in 1:n)  {
        time[i] ~ dweib(alpha, scale)      # Weibull likelihood function for n times
    }
alpha ~ dgamma(0.0001, 0.0001)          # Diffuse priors for alpha and beta
scale ~ dgamma(0.0001, 0.0001)
}
data
list(time=c(0.8, 1.7, 0.5, 11.4, 0.1, 5.7, 3.0, 1.1, 2.5, 0.04), n=10)
inits
list(alpha=1, scale=1)
list(alpha=0.5, scale=10)
```

- Script 28. WinBUGS script for Bayesian inference of random durations using Weibull likelihood.
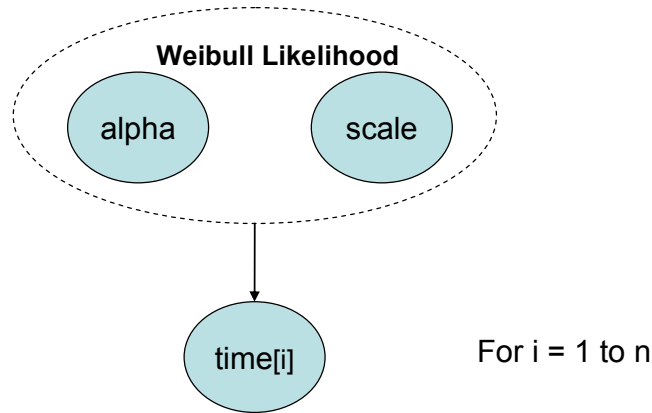
- Figure 4-62. DAG representing Script 28.

Running the script in the usual way produces the posterior distribution for alpha shown below. As with the gamma model example, the posterior distribution indicates that values of alpha less than one are most likely, suggesting that the repair rate is decreasing with time, and that the exponential model (with constant repair rate) may not be adequate.
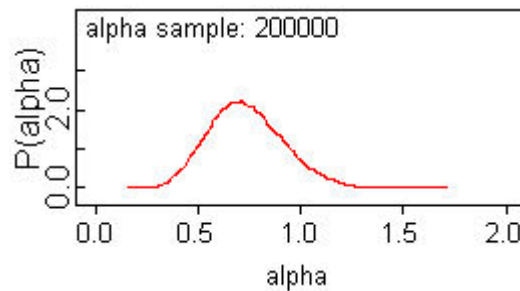
- Figure 4-63. Posterior distribution of alpha for Example 22.

4.6.1.3 Lognormal Distribution as Likelihood for Random Durations—In addition to its use as a nonconjugate prior, the lognormal distribution is also a popular aleatory model for random durations. It has the interesting property that the rate of the process initially increases, and then decreases monotonically. The period over which the rate increases is often very short, so that it can be used as an alternative to the gamma or Weibull distribution with shape parameter less than one. It has a heavier tail than the gamma or Weibull distribution, and may therefore produce more conservative results. The parameters of the lognormal distribution (mu and tau) cannot be interpreted in the manner of the gamma and Weibull shape parameter, and the lognormal distribution does not reduce to the exponential distribution for certain parameter values, as do the gamma and Weibull distribution. Changed assumptions when using the lognormal model (instead of exponential) are:

- ☑ The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval, but the constant of proportionality is a function of time.

- Example 23. Inference for lognormal distribution as aleatory model for circulating pump failure times in the ATCS.

> We will carry out Bayesian inference for the times in Example 21 under the assumption that the aleatory model is a lognormal distribution with parameters called mu and tau. The WinBUGS Script 29 is used for this analysis with diffuse priors on the lognormal parameters.

```
model  {
    for(i in 1:n) {
        time[i] ~ dlnorm(mu, tau)        # Lognormal likelihood function for n times
    }
mu ~ dflat()                             # Diffuse priors for lognormal parameters
sigma ~ dunif(0, 10)
tau <- pow(sigma, -2)
}


data
list(time=c(0.8, 1.7, 0.5, 11.4, 0.1, 5.7, 3.0, 1.1, 2.5, 0.04), n=10)


inits
list(mu=1, sigma=1)
list(mu=0.5, sigma=0.5)
```

- Script 29. WinBUGS script for Bayesian inference of random durations using lognormal likelihood.
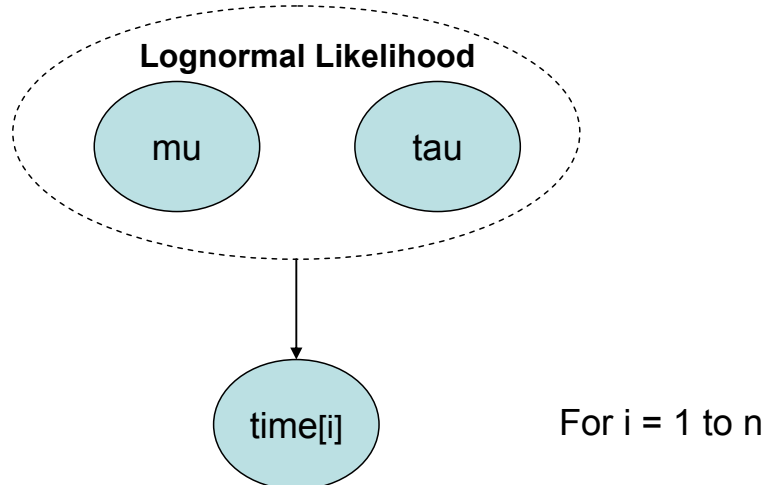
- Figure 4-64. DAG representing Script 29.

Running Script 29 in the usual way produces the posterior mu of 3.6×10^{-2} with a 90% interval of -1.05 to 1.12 and tau of 0.29 with a 90% interval of 0.10 to 0.57. The posterior distributions on these two parameters are shown in Figure 4-65 and Figure 4-66. As stated above, it is not possible to interpret these distributions in terms of evidence for or against an exponential model – however the next section does provide one method of quantitative interpretation.
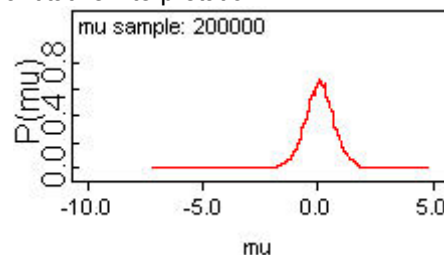
- Figure 4-65. Posterior distribution of mu in Example 23.
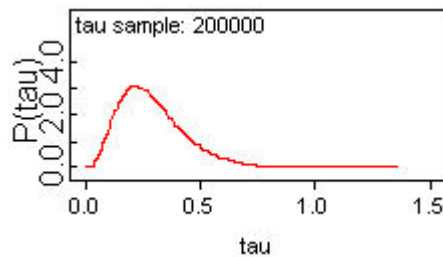
- Figure 4-66. Posterior distribution of tau in Example 23.

4.6.2 Choosing Among Alternative Distributions—Deviance Information Criterion (DIC)

In the examples from the previous section, we have developed four alternative aleatory models for the repair times given in Example 21: exponential, gamma, Weibull, and lognormal. By looking at the posterior distributions for the gamma and Weibull shape parameters, we have concluded that the exponential model may not be adequate, as it appears that the repair rate is a decreasing function of time. **How can we select among these models?** WinBUGS provides a quantitative measure called the *deviance information criterion* or DIC that can help in making this selection. The DIC measures how well a model replicates the observed data, but penalizes models with more parameters, to avoid over-fitting, which can produce a model that is poor at extrapolating beyond the observed data.

> **To avoid getting an invalid DIC, do not calculate DIC until parameter values have been estimated. This will ensure simulation has converged.**

The model with the **smallest DIC is the preferred model**, although differences in DIC less than about five may not be significant. Note that DIC can be negative in some cases; the model with the smallest DIC is still preferred. As an example of this, if there were three models under consideration, with DICs of 10, -3, and -9, the model with a DIC of -9 would be the preferred model. Note that DIC is a measure of *relative* goodness of fit; the model with the smallest DIC may still not be an adequate model.

4.6.2.1 Calculating DIC in WinBUGS—First note that DIC must be calculated on the same data set for each of the alternative models. Also, DIC should not be estimated until the sampling has converged to the posterior distribution. Here are the steps for calculating DIC in WinBUGS:

1. Estimate parameter values in the usual way.
2. Set DIC as shown below (Inference -> DIC -> Click "Set")
3. Run iterations to calculate DIC
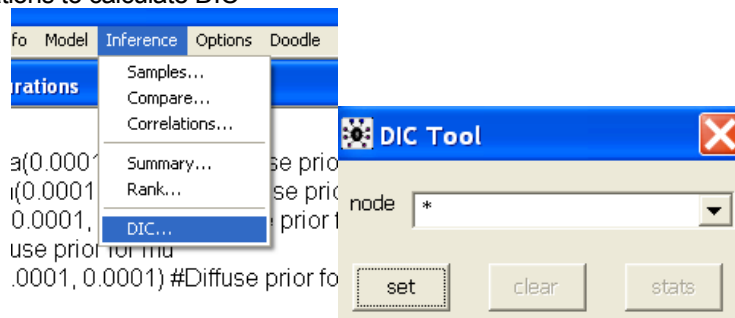
We illustrate this option by calculating DIC for each of the four models using the data from Example 21. The results are listed below.

Exponential model:

	Dbar	Dhat	DIC	pD
time	40.8	39.8	41.8	1.02
total	40.8	39.8	41.8	1.02

Minimum deviance = 39.8

Gamma model:

	Dbar	Dhat	DIC	pD
time	40.4	38.2	42.5	2.12
total	40.4	38.2	42.5	2.12

Minimum deviance = 38.2

Weibull model:

	Dbar	Dhat	DIC	pD
time	40.2	38.2	42.3	2.05
total	40.2	38.2	42.3	2.05

Minimum deviance = 38.2

Lognormal model:

	Dbar	Dhat	DIC	pD
time	41.8	40.1	43.6	1.75
total	41.8	40.1	43.6	1.75

Minimum deviance = 39.2

Caution: the value for pD in the DIC results reported by WinBUGS must be positive in order for DIC to be used to compare models.

Perhaps surprisingly, the exponential model has the lowest DIC, although the differences among the four models are small. The choice of the exponential model based on DIC also agrees with the p.value of 0.62 from Script 20. With only 10 observed times, the data is relatively sparse, and the penalty paid for introducing a second parameter offsets the improved ability of the model to replicate the observed data.

4.7 Analyzing Failure Time Data from Repairable Systems

In Sections 4.3 and 4.6, we analyzed times to occurrence of an event of interest. Such an event could be

> **Analyzing data from repairable systems can be complicated and may require expert consultation.**
>
> **Likewise, incorporating results into a PRA may not be straightforward.**

failure of a component or system. If the failure is **not repaired**, and the component or system is replaced following failure, then the earlier analysis methods are applicable. However, in this section, we consider the case in which the failed component or system is repaired and placed back into service. Analysis in this situation is more complicated. The details will depend principally upon the state of the system or component after repair. We will consider two cases:[1]

1. Repair leaves the component or system the same as **new** ("reincarnated"),
2. Repair leaves the component or system the same as **old** ("resuscitated").

Both of these cases are modeling assumptions that an analyst must make, and they lead to different aleatory models for the failure time. We will provide some qualitative guidance for when each assumption might be appropriate, along with some qualitative and quantitative model checks that can be used to validate the assumption.

In all of the following discussion, we assume that we can ignore the time it takes to actually repair a component or system that has failed. This allows us to treat the failure process as a simple point process. This assumption is typically valid either because repair time is short with respect to operational time, or because we are only concerned with operational time, so time out for repair is accounted for through component or system maintenance unavailability estimates.

4.7.1 Repair Same as New—Renewal Process

In this (same as new) case, repair leaves the failed component or system in the same state as a new

> **Repair to as good as new may be plausible when a component is replaced with a new one or completely overhauled.**

component or system. The times between failures are thus independent and will be assumed to be identically distributed. If the times between failures are exponentially distributed, the methods of Section 4.3 can be applied. If the times between failures are not exponentially distributed (e.g., Weibull or lognormal), then the methods of Section 4.6 can be applied. Note in both cases that it is the times *between* failures that are analyzed, not the *cumulative* failure times.

The assumption of repair same as new is plausible when the entire component or system is replaced or completely overhauled following failure. Examples would be replacement of a failed circuit board or rewinding a motor. The assumptions applicable to this section are:
- ☑ Repair restores a component to a new (not degraded) state.
- ☑ The random event that is observed is the time to an event and has a likelihood based on a shared (i.e., common) renewal distribution.

---

[1] Intermediate cases, in which repair leaves the component in an intermediate state between new and old, can also be modeled, along with imperfect repair, which leaves a component worse than old. Such more general models are still an area of research and thus practical guidelines are difficult to give, so they are not included herein.

4.7.1.1 Qualitative Check for Failure Rate in Renewal Process—If one assumes a renewal process, then a qualitative check on whether the failure rate is constant can be done using the times between failures. If the failure rate is constant, then the times between failures are exponentially distributed. If one plots the ranked times between failures on the x-axis, and $1/n_t$ on the y-axis, where n_t is the number of components still operating at time t, the result should be approximately a straight line due to the assumption of a constant failure rate (failures are not increasing or decreasing in time).

> **A cumulative hazard plot is only useful if repair is same as new, that is, failures are described by a renewal process.**

If the slope is increasing (decreasing) with time, this suggests a renewal process whose failure rate is likewise increasing (decreasing) with time, thereby invalidating one of the assumptions in this section. Such a plot is referred to as a *cumulative hazard plot*.

- Example 24. Servo motor failure example, with replacements.

Consider the following 25 cumulative times of failure (in days) for a servo motor. Assume that the motor is replaced with a new one each time it fails, so the assumption of a renewal process seems reasonable. Use a cumulative hazard plot to decide if the failure rate appears to be increasing or decreasing with time.

Cumulative time (days)	Time between failures (days)	Sorted time between failures
127.4920	127.492	27.1964
154.6884	27.1964	74.28
330.4580	175.7696	76.22
739.9158	409.4578	87.037
1153.074	413.1582	123.803
1470.720	317.646	127.492
1809.616	338.896	141.964
2118.147	308.531	160.479
2289.570	171.423	171.423
2365.790	76.22	175.7696
2757.970	392.18	193.382
3154.409	396.439	201.649
3448.874	294.465	258.093
3941.777	492.903	294.465
4143.426	201.649	308.531
4217.706	74.28	317.646
4359.670	141.964	338.896
4483.473	123.803	352.491
4570.51	87.037	392.18
4763.892	193.382	396.439
4924.371	160.479	409.4578
5360.967	436.596	413.1582
5619.06	258.093	436.596
5971.551	352.491	476.511
6448.062	476.511	492.903

There are 25 times so the cumulative hazard plot increases by 1/25 at 27.1964, by 1/24 at 74.28, etc. The cumulative hazard plot is shown below, and appears to indicate an increasing failure rate with time.
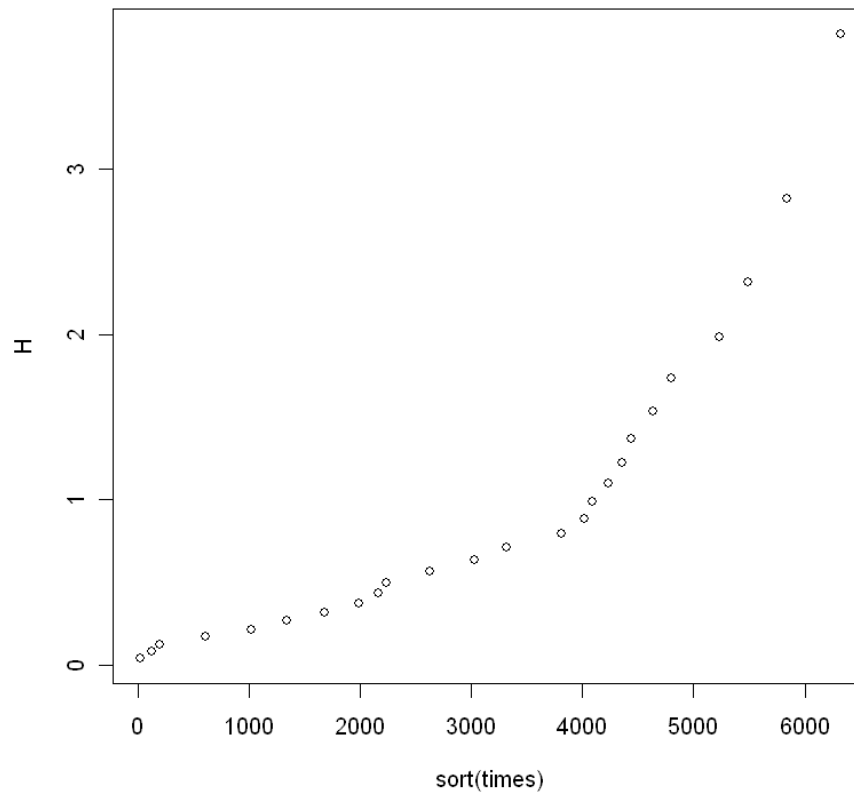
- Figure 4-67. Cumulative hazard plot for Example 24, suggesting increasing failure rate with operating time.

Quantitative analysis of these times can be carried out using the methods in Sections 4.3 and 4.6. Another quantitative check will be described below, after we have discussed the other extreme: repair same as old.

4.7.2 Repair Same as Old—Nonhomogeneous Poisson Process (NHPP)

In the case where repair only leaves the component in the condition it was in immediately preceding failure, then the times between failures may not be independent. For example, if the component is wearing out over time (**aging**), then later times between failures will tend to be shorter than earlier times, and conversely if the component is experiencing **reliability growth**. In these two cases, the times between failures will also fail to meet the assumption of being identically distributed. The assumptions applicable to this section are:

> **If repair is not to the same state as a new component, then the methods of Secs. 4.3 and 4.6 cannot be applied.**

☑ Repair restores a component to an old (not new) state.
☑ Times between failures are not independent and are not identically distributed.

Repair to a state the same as old is a good default assumption for most components in a risk assessment, because a typical component is composed of subcomponents. When failure occurs, only a portion of the component (one or more subcomponents) is typically repaired, so the majority of the subcomponents are left in the condition they were in at the time of failure.

4.7.2.1 Qualitative Check for Trend When Repair is Same as Old—If the rate of occurrence of failures (ROCOF) is constant with time, then the times between failures will not tend to get shorter (aging) or longer (reliability growth) over time. If one plots cumulative number of failures on the y-axis versus cumulative failure time on the x-axis, the resulting plot will be approximately a straight line if ROCOF is constant. If aging is occurring, the slope will increase with time, as the times between failures get shorter. If reliability growth is occurring, the slope will decrease with time, as the times between failures get longer.

> **The cumulative failure plot is only useful if repair is same as old, that is, failures are not described by a renewal process.**

- Example 25. Cooling unit failure times.

Consider the following 25 cumulative times in standby at which a cooling unit failed. Because the cooling unit consists of a large number of subcomponents, and only one or two of these were replaced at each failure, assume repair leaves the cooling unit in the state it was in immediately prior to failure. Plot cumulative number of failures versus cumulative failure time to check if there appears to be a time trend in the ROCOF for the cooling unit.

Cumulative time (days)
116.0454
420.8451
523.1398
538.3135
585.581
591.5301
772.365
868.7294
912.3777
1031.021
1031.133
1086.673
1096.476
1103.463
1165.640
1257.554
1375.917
1385.808
1421.459
1456.259
1484.755
1496.982
1523.915
1526.050
1530.836

The plot is shown below. The slope appears to be increasing with time, suggesting that the ROCOF for the cooling unit is increasing as a function of calendar time.
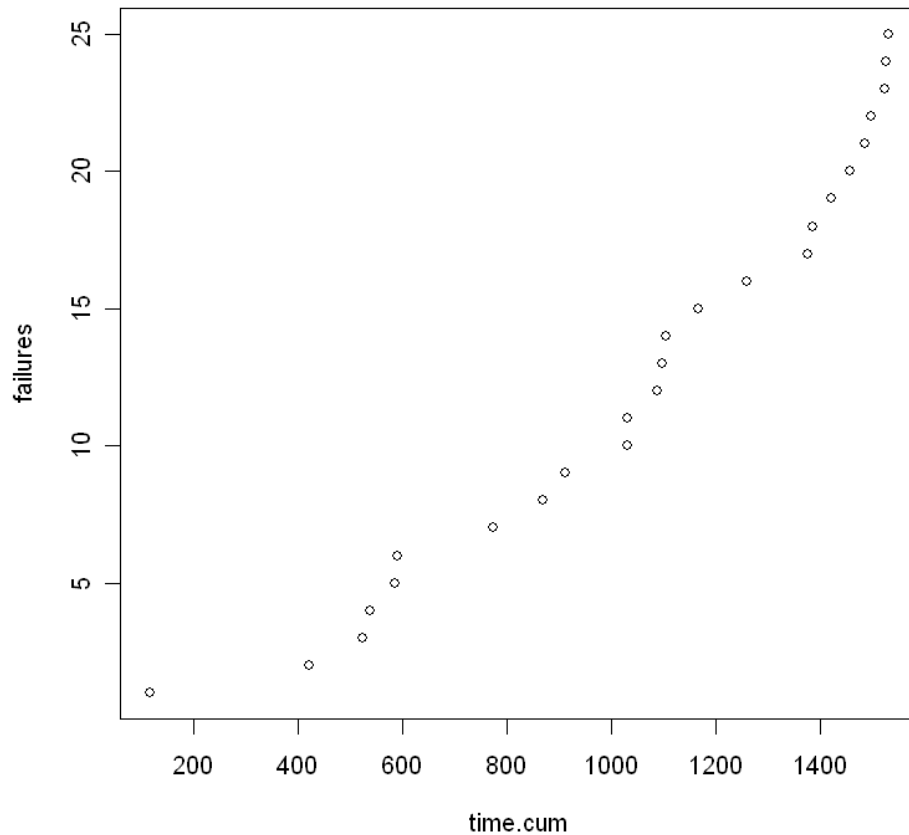
• Figure 4-68. Cumulative failure plot for Example 25, suggesting increasing ROCOF over time.

An interesting exercise is to construct a cumulative failure plot for the data in Example 24, where the failure rate appeared to be an increasing function of operating time under the assumption that repair was same as new, a renewal process. The plot, shown in Figure 4-69, does **not** suggest an increasing ROCOF under the assumption of repair same as old. This plot illustrates a subtle point in analyzing repairable systems. If repair is same as new after each failure, then times between failures will not exhibit a trend over calendar time. Aging or reliability growth only occurs over the time between one failure and the next, because the system returns to new, and the clock is reset, after each failure. On the other hand, when repair is same as old after each failure, then aging or reliability growth occurs over calendar time and one can then expect to see a trend in the slope of cumulative failures versus time. Therefore, absence of a trend in the cumulative failure plot may suggest no aging or reliability growth under the assumption of repair same as old, but there still may be aging or reliability growth between each failure under the assumption of repair same as new. The cumulative hazard plot shown earlier can be used to check for that possibility.
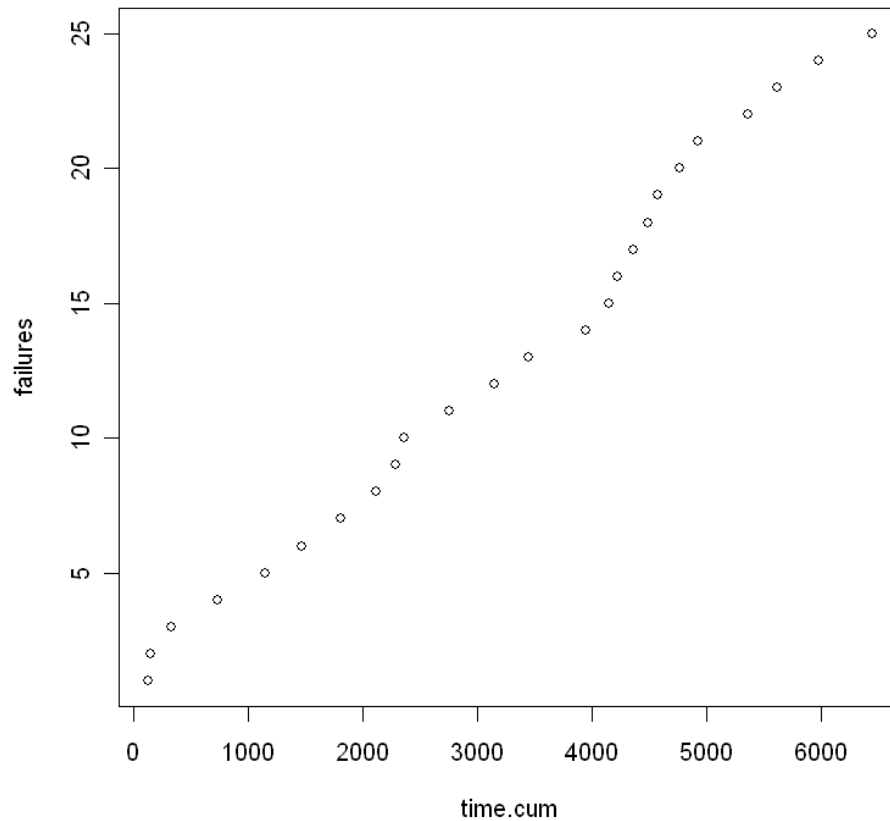
- Figure 4-69. Cumulative failure plot for data in Example 24, showing lack of trend in slope over calendar time.

The plots below (Figure 4-70 to Figure 4-73 and Figure 4-75) show cumulative failures versus cumulative time for 1,000 simulated failure times from two different renewal processes, one in which reliability growth is occurring between each failure, the other where aging takes place between each failure. Note in both cases that the cumulative failure plot produces a straight line, reinforcing the conclusion that this plot is useful for checking for aging or reliability growth under the same-as-old assumption for repair, but it cannot detect a time-dependent failure rate under the same-as-new repair assumption. The corresponding cumulative hazard plots shown below are useful for this purpose when repair is same as new.
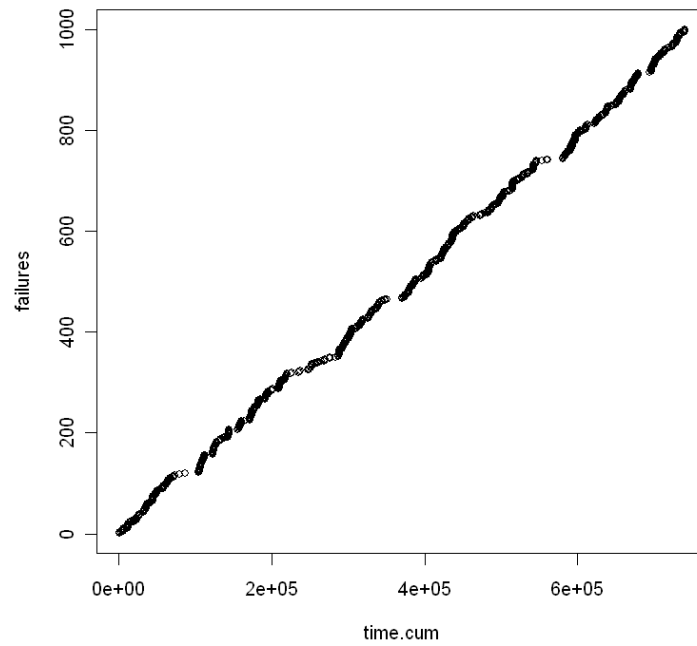
• Figure 4-70. Cumulative failure plot for 1,000 simulated failure times from renewal process with decreasing failure rate.
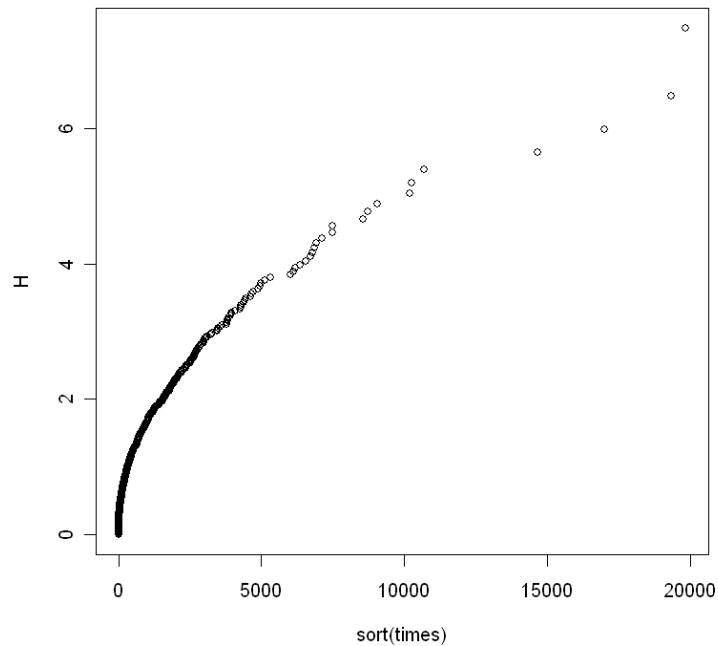
• Figure 4-71. Cumulative hazard plot for 1,000 simulated failure times from renewal process with decreasing failure rate.
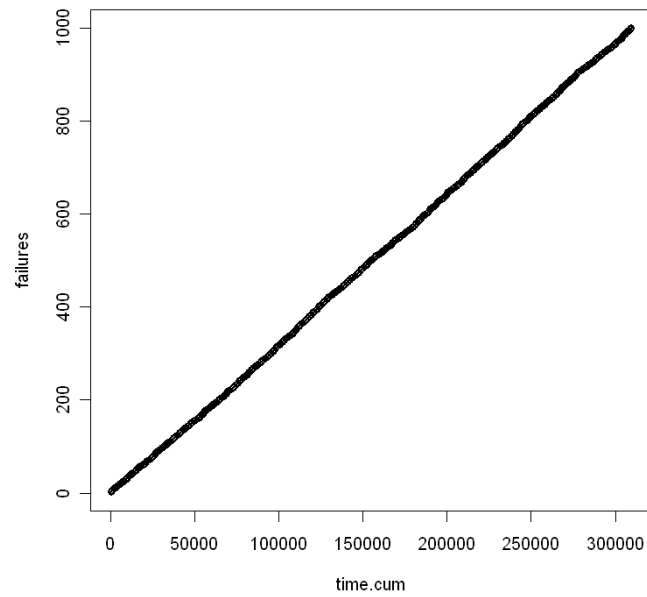
- Figure 4-72. Cumulative failure plot for 1,000 simulated failure times from renewal process with increasing failure rate.
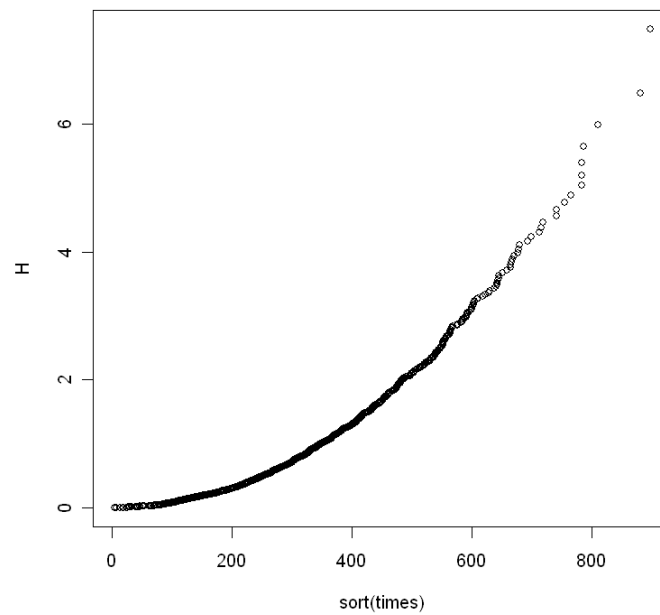
- Figure 4-73. Cumulative hazard plot for 1,000 simulated failure times from renewal process with increasing failure rate.

4.7.2.2 Quantitative Analysis under Same-as-Old Repair Assumption—As stated above, if there is an increasing or decreasing trend in the ROCOF over time, then the times between failures will not be independently and identically distributed, and thus the methods of Sections 4.3 and 4.6 cannot be applied (and invalidating one of the assumptions of this section).

In particular, one **cannot** simply fit a Weibull, gamma, etc., distribution to the **cumulative** failure times or the times between failures. Instead, the likelihood function must be constructed using the fact that each failure time, after the first, is dependent upon the preceding failure time. One must also specify a functional form for the ROCOF.

We will assume a power-law form for our analysis here.[1]

$$\Phi[j] = \ln(\beta) - \beta * \ln(\alpha) + (\beta - 1)*\ln[t(j)] - ([t(M) / \alpha]^{\beta} / M)$$

In this form of the power-law function, there are two parameters to estimate, which we denote as alpha (α) and beta (β). Beta determines how the ROCOF changes over time, and alpha sets the units with which time is measured.

If beta is **less** than one, reliability growth is occurring, if it is **greater** than one, aging is taking place, and if beta equals one, there is no trend over time. The WinBUGS script shown below is used to estimate alpha and beta. It also provides a Bayesian p-value to check the validity of the model.[2]

---

[1] The power-law ROCOF is also referred to as the Crow-AMSAA model in the reliability community.

[2] In Sec. 4.6, we used alpha for the shape parameter of the Weibull distribution. We use beta in this section to conform to standard notation in the literature.

```
Modeling NHPP (repair "as bad as old")
Failure-truncated data
model   {
    for(i in 1:M) {
        zeros[i] <- 0
        zeros[i] ~ dgeneric(phi[i])
        #phi[i] = log(likelihood)
    }
#Power-law model (failure-truncated)
    for(j in 1:M) {
        phi[j] <- log(beta) - beta*log(alpha) + (beta-1)*log(t[j]) - pow(t[M]/alpha, beta)/M
    }
#Model validation section
    for(j in 1:M) {
        z.obs[j] <- pow(t[j]/alpha, beta)
    }
z.inc.obs[1] <- z.obs[1]
    for(k in 2:M) {
        z.inc.obs[k] <- z.obs[k] - z.obs[k-1]
    }
for(j in 1:M) {
    z.inc.rep[j] ~ dexp(1)
    z.rep.ranked[j] <- ranked(z.inc.rep[], j)
    z.obs.ranked[j] <- ranked(z.inc.obs[], j)
    F.obs[j] <- 1 - exp(-z.obs.ranked[j])
    F.rep[j] <- 1 - exp(-z.rep.ranked[j])
    diff.obs[j] <- pow(F.obs[j] - (2*j-1)/(2*M), 2)
    diff.rep[j] <- pow(F.rep[j] - (2*j-1)/(2*M), 2)
    }
CVM.obs <- sum(diff.obs[])
CVM.rep <- sum(diff.rep[])
p.value <- step(CVM.rep - CVM.obs)
alpha ~ dgamma(0.0001, 0.0001)
beta ~ dgamma(0.0001, 0.0001)
}

data
list( t=c(116.0454, 420.8451, 523.1398, 538.3135, 585.581, 591.5301, 772.365,
    868.7294, 912.3777, 1031.021, 1031.133, 1086.673, 1096.476, 1103.463,
    1165.640, 1257.554, 1375.917, 1385.808, 1421.459, 1456.259, 1484.755,
    1496.982, 1523.915, 1526.050, 1530.836))
list(M=25)

inits
list(alpha = 1, beta = 0.1)
list(alpha = 0.5, beta = 1)
```

• Script 30. WinBUGS script for analyzing data under same-as-old repair assumption (power-law process).
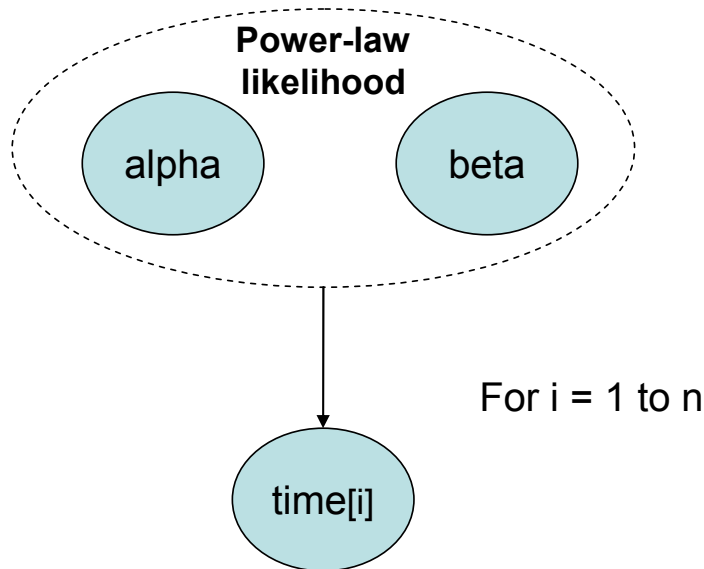
• Figure 4-74. DAG representing Script 30 (omits model validation portion)

Let us apply this script to the data given in Example 25. Recall that Figure 4-68 suggested an increasing trend in ROCOF with time, corresponding to beta greater than one. We will run two chains, one starting with an initial value of beta less than one, the other with an initial beta greater than one. The initial values of the scale parameter (alpha) are not crucial; we will vary alpha across the chains. Convergence to the joint posterior distribution appears to occur within the first 1,000 samples, so we discard the first 1,000 samples for burn in. We run another 10,000 samples to estimate parameter values, obtaining a posterior mean for beta of 1.94 with a 90% credible interval of (1.35, 2.65). The posterior probability that beta is greater than one is near unity, suggesting a ROCOF that is increasing with time, corresponding to aging. The Bayesian p-value is 0.57, suggesting a model that is good at replicating the observed data.[1]

4.7.3 Impact of Assumption Regarding Repair

The assumption made regarding repair (same as old versus same as new) is crucial to the analysis. Consider times to failure being produced by a process with increasing ROCOF, corresponding to aging with repair same as old. As (calendar) time progresses, times between failure will tend to decrease, and there will be a preponderance of short times between failures in a sample of failure times. If the process is assumed to be a renewal process, with times between failures described by a Weibull distribution, as in Section 4.6, the preponderance of short times between failures will cause the Weibull shape parameter to be less than one, corresponding to apparent reliability growth between each failure, the opposite of what is actually happening.

This can be illustrated by simulation. First, we generated 1,000 cumulative failure times for a system whose repair is same-as-old, described by a power-law process with shape parameter of 2 and scale parameter of 350. The cumulative failure plot below shows the increasing trend in ROCOF with time.

---

[1] For reference, the data in Example 25 were simulated from a power-law process with beta = 2 and alpha = 350.
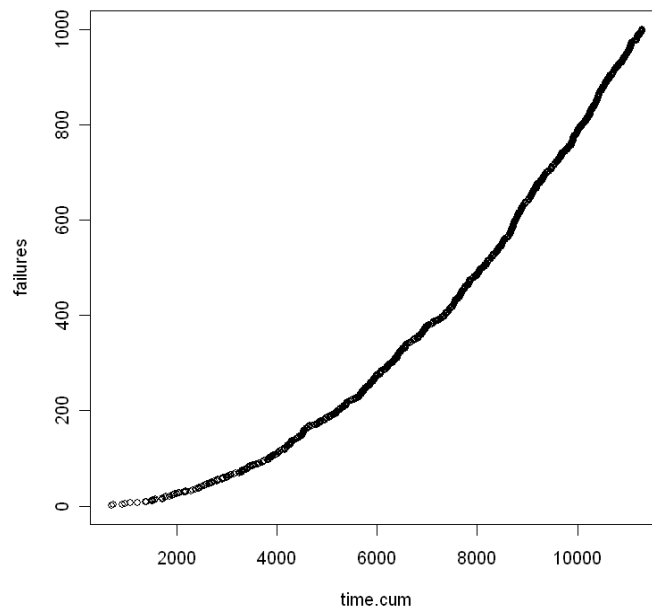
• Figure 4-75 Cumulative failure plot for 1,000 times simulated from power-law process with shape parameter of 2, illustrating increasing ROCOF

The histogram in Figure 4-76 of the times between failures shows the preponderance of short times between failures caused by the increasing ROCOF.
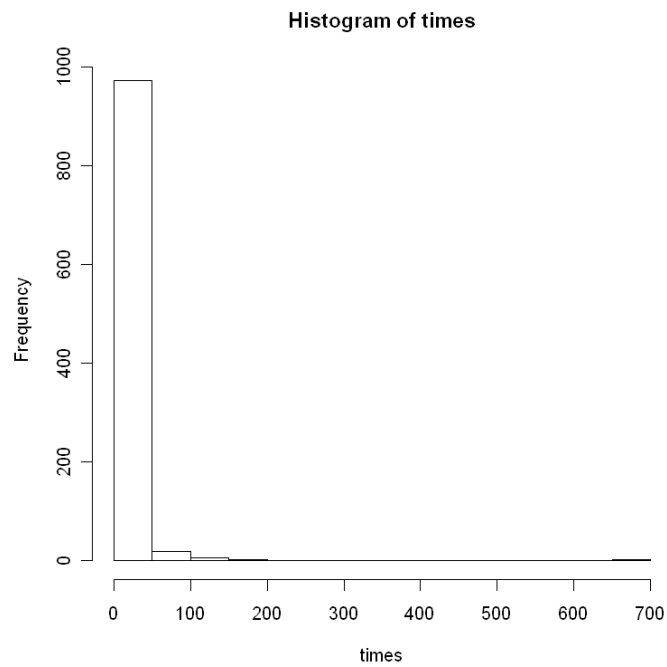
• Figure 4-76. Histogram of times between failures for simulated failure times from power-law process with increasing ROCOF.

If we had assumed the repair is same-as-new and fitted a Weibull distribution to these times between failures using the techniques of Section 4.6, one estimates a Weibull shape parameter of about 0.8, which would suggest reliability growth over time between each failure. This result is caused by the fact that times between failures are tending to become shorter due to aging. Consequently, treating the times **between** failures as independent and identically distributed leads to an erroneous conclusion about the process. Unfortunately, the Bayesian p-value may not help much in deciding which model is better, because both models can replicate the observed data quite well.

If the repair is same-as-new, and the failure rate increases with operating time or time in standby (whichever is being modeled), an assumption of same-as-old repair will, as suggested by Figure 4-72, lead to an estimate near one for the shape parameter of the power-law process. In this case, the Bayesian p-value can be helpful, as a power-law process with shape parameter near one cannot replicate data from a renewal process with increasing failure rate very well.

4.8 Treatment of Uncertain Data

So far we have only analyzed cases where the observed data were **known** with certainty and completeness. Reality is messier in a number of ways with respect to observed data. For example, the number of binomial demands may not have been recorded and may have to be estimated. Similarly, the exposure time in the Poisson distribution may have to be estimated. One may not always be able to tell the exact number of failures that have occurred, because of imprecision in the failure criterion. For example, one of the ATCS component failure modes might be plugging of the radiators. The analyst may have an established failure criterion expressed in terms of differential pressure across the radiator. However, in examining failure records, the analyst may not be able to tell in all cases whether a putative plugging event has met the differential pressure failure criterion, leading to epistemic uncertainty as to the actual number of observed failures.

When observing times at which failures occur (i.e., random durations), various types of *censoring* can occur. For example, a number of components may be placed in test, but the test is terminated before all the components have failed. This produces a set of observed data consisting of the recorded failure times for those components that have failed. For the components that did not fail before the test was terminated, all we know is that the failure time was longer than the duration of the test. As another example, in recording fire suppression times, the exact time of suppression may not be known; in some cases, all that may be available is an interval estimate (e.g., between 10 and 20 minutes).

Finally, there is the issue of *heritage*[1] or *legacy* data, in which data from an earlier design is available, but the designer has made changes intended to eliminate some early failure modes. Thus, there is uncertainty as to the applicability of the earlier data to the latest revision of the design.

All of these cases can be treated within the Bayesian framework, and MCMC approaches make the analysis straightforward.

4.8.1 Uncertainty in Binomial Demands or Poisson Exposure Time

A common approach for this situation is to input a distribution for the number of demands or the exposure time. This distribution represents the analyst's epistemic uncertainty as to the actual number

---

[1] From [NASA 2007], "heritage" refers to the original manufacturer's level of quality and reliability that is built into parts and which has been proven by (1) time in service, (2) number of units in service, (3) mean time between failure performance, and (4) number of use cycles.

of demands or exposure time. Often this will be a uniform distribution between known lower and upper bounds. The posterior distribution for the parameter of interest is then averaged over this distribution. Assumptions in the section include the usual binomial or Poisson assumptions with the following exception:

    ☑ The observed data (specifically the number of demands or the exposure time) are not known with certainty.

- Example 26. Modeling uncertain demands in the binomial distribution.

> Assume in Example 1 that the number of demands is not known to be 285 but instead is only known to be between 100 and 500.
>
> Find the posterior mean and 90% credible interval for p using a uniform distribution between 100 and 500 to represent the uncertainty in the number of demands.

Solution

The prior distribution for p in Example 1 was a beta distribution with parameters 1.24 and 189,075. The beta prior was conjugate to the binomial likelihood in that example, and WinBUGS was not needed to carry out the analysis. Now, with a distribution on the number of demands, the answer cannot be written down by inspection, and a tool like WinBUGS or numerical integration is needed. The script below shows the change that is needed to analyze this problem. Note that this approach uses what is known as posterior-averaging, where (in this case) **each possible** posterior for the n number of demands is weighted equally. This topic is addressed in more detail in Section 4.8.2.2.

```
model  {                         # Model defined between { } symbols
x ~ dbin(p, n)                   # Binomial distribution for number of failures in n
demands
n ~ dunif(lower, upper)          # Uniform distribution for number of demands
p ~ dbeta(alpha.prior, beta.prior)       # Conjugate beta prior distribution for p
}

data
list(x=2, lower=100, upper=500)          # Data for Example 1
list(alpha.prior=1.24, beta.prior=189075)        # Prior parameters for Example 1
```

- Script 31. WinBUGS script for modeling uncertainty in binomial demands.
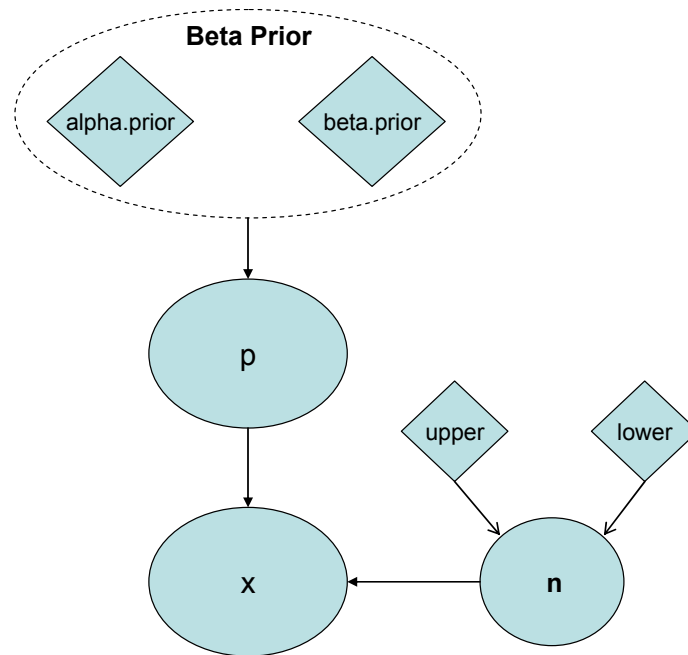
- Figure 4-77. DAG representing Script 31

Running Script 31 in the usual way, we find the posterior mean of p to be 1.7×10^{-5} and the 90% interval is $(4.9 \times 10^{-6}, 3.5 \times 10^{-5})$. These are the same results obtained in Example 1, where the number of demands was known to be exactly 285. Recall that the prior is highly inconsistent with 2 failures in so few demands, even as many as 500, and the relatively sparse data are having little influence on the posterior results. Thus, accounting for uncertainty in the demand count has little impact in this example.

- Example 27. Modeling uncertain exposure time in the Poisson distribution.

> In Example 3, the prior distribution was gamma with parameters 1.6 and 365,000 hours. The observed data were 0 failures in 200 days, giving a posterior mean and 90% credible interval of 4.3×10^{-6}/hour and $(5.6 \times 10^{-7}, 1.1 \times 10^{-5})$.
>
> Suppose 200 days was actually a lower bound on the exposure time and that it could have been as long as 300 days. How does this epistemic uncertainty in the exposure time affect the posterior mean and 90% credible interval?

Solution
We will use a uniform distribution between 200 and 300 days to represent the epistemic uncertainty in the exposure time. Because of this, the analysis is no longer conjugate; a tool like WinBUGS or numerical integration is needed. The script shown below was used to carry out the analysis.

```
model  {
x ~ dpois(mean.poisson)                 # Poisson distribution for number of events
mean.poisson <- lambda*time.hr          # Poisson parameter
time.hr ~ dunif(lower.hr, upper.hr)     # Uniform distribution for exposure time
lower.hr <- lower*24 #Convert to hours
upper.hr <- upper*24
lambda ~ dgamma(alpha.prior, beta.prior)        # Gamma prior distribution for lambda
}

data
list(x=0, lower=200, upper=300)
list(alpha.prior=1.6, beta.prior=365000)

inits
list(lambda=0.000001, time.hr=6000)
```

• Script 32. WinBUGS script for modeling uncertainty in Poisson exposure time.
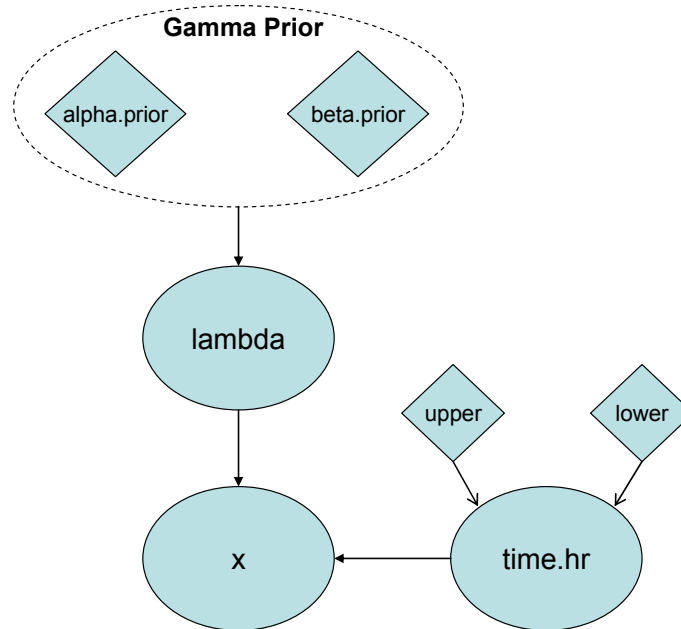
• Figure 4-78. DAG representing Script 32.

The results of running this script show essentially no difference from the case where the exposure time was taken to be exactly 200 days. Again, the difference could be significant in other problems; however, one often finds the results of the inference to not be very sensitive to epistemic uncertainties in binomial demand counts and Poisson exposure times. Nonetheless, if uncertainties are present, it is straightforward to include them in the inference via WinBUGS.

4.8.2 Uncertainty in Binomial or Poisson Failure Counts
Assumptions in the section include the usual binomial or Poisson assumptions with the following exception:
    ☑   The observed data (specifically the number of failures) are not known with certainty.

In this instance it is the observed number of events that is not known with certainty. There are two cases to consider:
- The analyst knows that the failure count is in a particular interval, but does not have information from which to develop a subjective probability distribution for the actual failure count. This will be referred to as the *interval*-censored case.
- Information is available to be able to develop a distribution for the failure count.

Bayesian analysis of the interval-censored case is straightforward. For the second case, there are a variety of approaches that have been suggested. We will illustrate three of these approaches, and list some salient insights about each.

4.8.2.1 Interval-Censored Failure Counts—In this case, the available information allows the analyst to say with certainty that the actual failure count is in a particular interval, but is insufficient to use in developing a subjective distribution expressing the analyst's belief as to the probability of the actual count. Consequently, the likelihood function becomes the probability that the failure count is in the specified interval; it is a sum of binomial or Poisson probabilities (depending on the failure model), conditional upon the unknown parameter (p or lambda, respectively).

- Example 28. Modeling uncertainty in binomial failure counts for the ATCS mixing valve. Suppose an analyst is estimating the probability that the mixing valve in the ATCS fails to change position in response to a signal from the avionics controller.

  Assume the prior distribution of the failure probability is lognormal with a median value of 0.001 and an error factor of 5. Records for this valve indicate there has been 1 failure in 187 demands. However, personnel responsible for maintaining the system indicate that there may have been two other failures which were not recorded properly.

  Find the posterior mean and 90% credible interval for p assuming:
  - There was exactly 1 failure in 187 demands
  - There may have been as many as 3 failures in 187 demands.

Solution
The first part is solvable using existing binomial/lognormal script, modified for this example as shown in Script 33. Running this script in the usual way gives a posterior mean of 2.3×10^{-3} and a 90% credible interval of (4.1×10^{-4}, 6.3×10^{-3}).

```
model  {
x ~ dbin(p, n)                          # Binomial distribution for number of failures
p ~ dlnorm(mu, tau)                     # Lognormal prior distribution for p
tau <- 1/pow(log(prior.EF)/1.645, 2)    # Calculate tau from lognormal error factor
# Calculate mu from lognormal prior median and error factor
mu <- log(prior.median)
}
data
list(x=1,n=187, prior.median=0.001, prior.EF=5)
```

- Script 33. WinBUGS script for first part of Example 28.
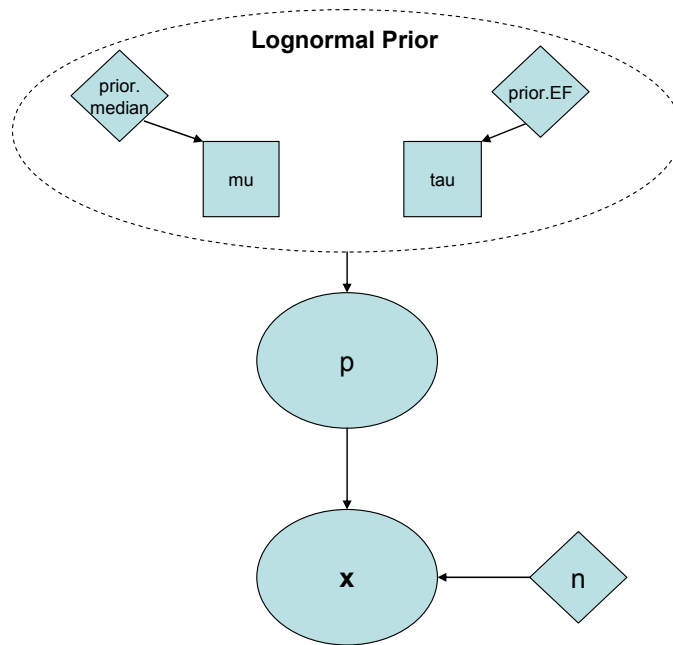
- Figure 4-79. DAG representing Script 33.

The WinBUGS script below implements the likelihood-based approach for the second part of the example, in which the analyst is not required to develop a discrete distribution representing uncertainty in the failure count. Running this script in the usual way gives a posterior mean for p of 2.9×10^{-3} and a 90% credible interval of (4.6×10^{-4}, 8.1×10^{-3}).

```
model   {
x ~ dbin(p, n)C(lower, upper)          # Binomial distribution for number of failures
p ~ dlnorm(mu, tau)                     # Lognormal prior distribution for p
tau <- 1/pow(log(prior.EF)/1.645, 2)    # Calculate tau from lognormal error factor
# Calculate mu from lognormal prior median and error factor
mu <- log(prior.median)
}
data
list(x=NA, lower=1, upper=3, n=187, prior.median=0.001, prior.EF=5)
```

- Script 34. WinBUGS script for incorporating uncertainty in binomial failure count via likelihood-based approach.

4.8.2.2 Uncertain Failure Count with a Subjective Probability Distribution—When information is available (e.g., from equipment records) that the analyst can use to make a judgment as to the likelihood that each possible failure count is the true value, Bayesian inference is less straightforward, as there is a lack of consensus as to the appropriate approach. We will discuss three approaches that have been proposed. We will begin with a simple example, which is helpful for illustrating some salient properties of each method. We will then return to the ATCS mixing-valve example to compare the practical differences among the three approaches.

Approach 1: Posterior Averaging
In this approach, a posterior distribution is found for each possible failure count, and then a weighted-average posterior distribution is constructed by averaging over the analyst's subjective probability for each failure count.

Approach 2: Likelihood Averaging
In this approach, the analyst's subjective probability for each failure count is used to construct a weighted-average likelihood function, which is then used to update the prior distribution in the usual manner. Note that there is only one update performed in this approach.

Approach 3: Weighted Likelihood
In this approach, the analyst's subjective probability for each failure count is used to weight the likelihood of that failure count, with the weight appearing in the exponent of the likelihood function. The effect of this approach is to discount the failure and demand counts by the weighting factor. This approach always leads to a unimodal posterior distribution.

Example Calculation
We begin our discussion of these three approaches with a simple example: a single demand in which the outcome is either success (X = 0) with probability 1 – p or failure (X = 1) with probability p. Let the prior distribution for p be uniform(0, 1), which is a beta(1, 1) distribution. Therefore, the posterior distribution of p will be a beta(x + 1, 2 – x) distribution due to the beta distribution being a conjugate to the Bernoulli model.

If we were to treat this example as being interval-censored, all we would know is that the value of X is either 0 or 1; in other words, we have gained no information, because this was the original constraint on X. We would thus expect the posterior distribution to be unchanged from the prior distribution. The likelihood function is the probability that X = 0 plus the probability that X = 1, conditional upon a value of p. Thus, the likelihood function is

$$f(0\,|\,p) + f(1\,|\,p) = 1 - p + p = 1.$$

The prior distribution for p is uniform, so the posterior is also uniform, as expected since a uniform prior is conjugate with the Bernoulli model. In fact, the posterior will be the same as the prior for any prior distribution.

In the posterior-averaging approach, where we weight the two possible posteriors [g(p|0) and g(p|1)] with weight *w*, we can calculate the marginal posterior distribution for p by averaging over the possible values of X (which in this case has only two possible values, 0 and 1). The averaging yields

$$g(p) = w\,g(p\,|\,0) + (1 - w)g(p\,|\,1).$$

When the two values are judged equally likely (*w* = 0.5) and when x = 0, we have

$$g(p\,|\,0) = beta(1, 2) = 2(1 - p).$$

Alternatively, when x = 1, we have

$$g(p\,|\,1) = beta(2, 1) = 2p\ .$$

Therefore, the marginal (weighted-average) posterior is given by

$$g(p) = \frac{2(1 - p) + 2p}{2} = 1$$

So the marginal posterior distribution is a uniform(0, 1) distribution, just as we obtained in the interval-censored case. With unequal weights, the marginal posterior distribution in the posterior averaging approach will be given by

$$g(p) = \frac{w(1 - p) + (1 - w)p}{2} = \frac{w + (1 - 2w)p}{2}$$

In the average-likelihood approach, the likelihood function used to update the prior is a weighted average

$$\langle f(x \mid p) \rangle = w f(0 \mid p) + (1 - w) f(1 \mid p) = w + p(1 - 2w)$$

In the case that the values are equally likely (w = 0.5) the likelihood function reduces to 0.5. In this case, the posterior distribution is equal to the prior, which is uniform(0, 1). If the values are not equally likely, then the posterior will be given by *g(p) = [w + p(1 − 2w)] / k*, where

$$k = \int_0^1 [w + (1 - 2w)p]\,dp = \frac{1}{2}$$

In the weighted-likelihood approach, the subjective probabilities developed by the analyst are used as exponential weighting factors for the likelihood function. In the case here of a single trial of the Bernoulli model, the result is

$$f(x \mid p) = \prod_{i=1}^2 [p^{x_i}(1 - p)^{1 - x_i}]^{w_i}$$

The effect of this likelihood function is as if there are $(w_i \cdot x_i)$ failures in $w_i(1 - x_i)$ demands. In the case of a single trial with a uniform prior on p, this gives a beta(1.5, 1.5) posterior distribution. Thus, the posterior mean is 0.5, as with the other approaches, but the 90% interval is (0.1, 0.9), narrower than in the other approaches. The posterior density obtained from the weighted-likelihood approach is shown in Figure 4-80, and differs quite markedly from the uniform posterior density obtained in the other approaches.
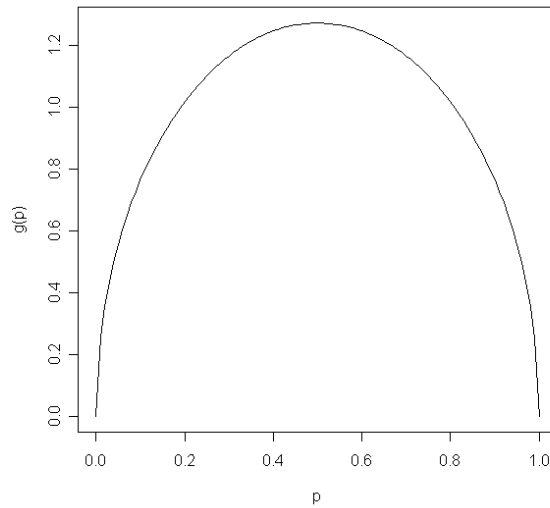
- Figure 4-80. Posterior density for p in the weighted-likelihood approach for a single trial with a uniform prior on p and equal weights for X = 0 and X = 1

Let us now consider the simple Bernoulli example, but with an informative prior for p. We will examine what happens when the prior distribution favors small values of p. We will use a beta(0.63, 123) distribution to illustrate this case. The prior mean is 5.1E-3, with a 90% interval of (5.9E-5, 1.8E-2).

With equal weights on the possible outcomes of a single trial for the Bernoulli model (i.e., 0 or 1), the interval-censoring and average-likelihood approaches give the same outcome, which is just the prior distribution. The posterior-averaging approach gives a different result, because the weights are retained. The posterior mean increases to 9.1E-3, and the 90% interval shifts to (1.7E-4, 2.8E-2). The weighted-

likelihood approach gives a similar posterior mean, 9.0E-3, but a narrower 90% interval, (6.4E-4, 2.6E-2). In the interval-censoring approach, the probability of an outcome is either p or 1 – p, so the weights are irrelevant. In the average-likelihood approach, the equal weights cancel in the numerator and denominator of Bayes' Theorem, giving the same effect. Thus, if an analyst's reason for assigning equal weights is because he is expressing indifference to the two possible values of X, then the average-likelihood approach will reflect this indifference, while the posterior-averaging approach will not.

Now consider the case of unequal weights. Recall that the beta(0.63, 123) prior distribution weights small values of p heavily, and so the most likely outcome of a trial is X = 0 (probability of this outcome is 0.995). Imagine that one such trial is performed, and the actual outcome is not certain, but information is available such that the analyst judges the outcome to be X = 1 with probability 0.9. In this case, we would expect the posterior distribution for p to be close to what we would get by updating the prior with X = 1. The **posterior-averaging** approach behaves as we would expect, giving a posterior mean of 0.012, compared to the value of 0.013 we would have obtained had the outcome been X = 1 with certainty. The 90% interval is (1.1E-3, 3.2E-2). The outcome of the **weighted-likelihood** approach is similar, with a posterior mean of 0.012 and a somewhat narrower 90% interval of (1.5E-3, 3.1E-2). In contrast, the **average-likelihood** approach gives a posterior mean of 0.005; *the posterior distribution is essentially unchanged by the trial, even though the analyst's probability that the outcome was X = 1 was quite high*. This aspect of the average-likelihood approach, its tendency to over-rule the data and allow the prior distribution to dominate the outcome, should be kept in mind if it is to be used with relatively sparse data.

We now return to our Example 1 with a total of 187 binomial demands on the ATCS mixing valve, and an uncertain failure count. Earlier we treated the uncertain count as interval-censored between 1 and 3. Now we will illustrate the other three approaches. We will first consider the case of equal weights for each of the possible outcomes. The WinBUGS scripts for each approach are shown below, and the results are summarized in Table 4. As expected from the simple Bernoulli example, the average-likelihood approach with equal weights is equivalent to interval censoring, and the posterior-averaging and weighted-likelihood approaches give similar numerical results, with the weighted-likelihood approach capturing less of the uncertainty present in the problem.

```
model  {
    for(i in 1:K) {
        x[i] ~ dbin(p[i], n)        # Binomial distribution for number of failures
        p[i] ~ dlnorm(mu, tau)    # Lognormal prior distribution for p
    }
p.avg <- p[r]                       # Monitor this node
r ~ dcat(p.analyst[])
    for(j in 1:3) {
        p.analyst[j] <- 1/3
    }
# Calculate tau from lognormal error factor
tau <- 1/pow(log(prior.EF)/1.645, 2)
# Calculate mu from lognormal prior median and error factor
mu <- log(prior.median)
}
data
list(x=c(1,2,3) ,n=187, prior.median=0.001, prior.EF=5, K=3)
```

• Script 35. The WinBUGS script for the posterior-averaging approach.

```
model   {
   for(i in 1:K) {
   phi[i] <- w[i]*exp(logfact(n) - logfact(x[i]) - logfact(n-x[i]))*pow(p, x[i])*pow(1-p, n-x[i])
   w[i] <- 1/3
   }
phi.sum <- sum(phi[])
log.phi.sum <- log(phi.sum)
zero <- 0
zero ~ dgeneric(log.phi.sum)
p ~ dlnorm(mu, tau)              #Lognormal prior distribution for p
# Calculate tau from lognormal error factor
tau <- 1/pow(log(prior.EF)/1.645, 2)
# Calculate mu from lognormal prior median and error factor
mu <- log(prior.median)
}
data
list(x=c(1,2,3) ,n=187, prior.median=0.001, prior.EF=5, K=3)
```

• Script 36. The WinBUGS script for average-likelihood approach.

```
model   {
   for(i in 1:K)    {
      zeros[i] <- 0
      zeros[i] ~ dgeneric(phi[i])
      #Phi is log-likelihood
      phi[i] <- w[i]*(logfact(n) - logfact(x[i]) - logfact(n-x[i]) + x[i]*log(p) + (n-x[i])*log(1-p))
      w[i] <- 1/K
   }
p ~ dlnorm(mu, tau) #Lognormal prior distribution for p
# Calculate tau from lognormal error factor
tau <- 1/pow(log(prior.EF)/1.645, 2)
# Calculate mu from lognormal prior median and error factor
mu <- log(prior.median)
}
data
list(x=c(1,2,3), n=187, prior.median=0.001, prior.EF=5, K=3)
```

• Script 37. The WinBUGS script for weighted-likelihood approach.

• Table 4. Summary of results for ATCS mixing valve.

Approach	Posterior Mean	90% Interval
Interval censoring	2.9E-3	(4.6E-4, 8.1E-3)
Posterior averaging	4.4E-3	(6.4E-4, 1.2E-2)
Average likelihood	2.8E-3	(4.6E-4, 8.0E-3)
Weighted likelihood	4.2E-3	(8.7E-4, 1.0E-2)

With unequal weights developed by the analyst, the interval-censoring approach is no longer applicable, as it does not use weights developed by the analyst. Let us assume in our mixing valve example that the analyst has developed the following subjective probabilities for the possible failure counts:

- $P(X = 1) = 0.1$
- $P(X = 2) = 0.2$
- $P(X = 3) = 0.7$

Note that the analyst has put a high probability on $X = 3$, which is a very unlikely outcome with the specified lognormal prior (probability = 0.01). We enter these weights and rerun the scripts shown above to obtain the results shown in Table 5. The average-likelihood results are somewhat nonconservative, and the weighted likelihood results are what would be obtained by updating the lognormal prior with the weighted-average failure count (2.6) and 187 demands, illustrating the equivalence of the weighted-likelihood approach and data-discounting. When the possible outcomes are more consistent with the prior distribution, the posterior-average and average-likelihood approaches will generally be similar.

- Table 5. Summary of results for ATCS mixing valve with unequal weights.

Approach	Posterior Mean	90% Interval
Posterior averaging	5.8E-3	(1.0E-3, 1.4E-2)
Average likelihood	3.7E-3	(5.3E-4, 1.E-2)
Weighted likelihood	5.6E-3	(1.3E-3, 1.3E-2)

Summary

When event counts are uncertain, there are various approaches that can be taken to Bayesian inference. In some limited cases, the outcomes of all the approaches will coincide; however in most cases they will not. The remainder of the examples in the Guidebook that involve uncertainty in event counts will be solved using the posterior-averaging approach. The reader should keep in mind that other approaches are available, and there is as yet no consensus on a single approach. We summarize, in Table 6, some key aspects of each of the approaches.

- Table 6. Summary of four approaches to Bayesian inference with uncertain event counts.

Approach	Salient Features
Interval censoring	1. Appropriate when data are known to lie in an interval, but information is not available to develop probability distribution for "true" event count
Posterior-averaging	2. Can lead to multimodal posterior 3. Not equivalent to interval censoring when weights are equal
Average likelihood	4. Equivalent to interval censoring when all weights are equal 5. Weights effectively ignored when data are sparse, leading to domination by the prior distribution 6. Can lead to multimodal posterior
Weighted likelihood	7. Always leads to unimodal posterior 8. Under-estimates uncertainty 9. Equivalent to data-discounting

As a demonstration of the posterior-averaging approach, we will revisit the ATCS radiator plugging failure mode via an example.

- Example 29. Modeling uncertainty in Poisson failure counts for radiator plugging in ATCS system.

> Consider plugging of the radiator in the ATCS.
>
> Assume that over the past 11,000 hours of operation, there have been 7 plugging events. Two of these events clearly met the failure criterion in terms of excess differential pressure across the radiator. However, the event narratives for the other five events are less clear as to whether the failure criterion was met. The analyst is pretty sure that three of the events met the criterion, and is also pretty sure that one of the events did not. For the other three events, she simply cannot tell.
>
> Assuming a prior on the radiator plugging rate that is a lognormal distribution with a mean of 10^{-5}/hour and an error factor of 7.6, find the posterior mean and 90% credible interval for the plugging rate
> - Using 2 failures in 11,000 hours.
> - Factoring in uncertainty in the failure count.

Solution

The first part is straightforward, using the existing Poisson/lognormal script, we have to modify the script somewhat for the specifics of this example. The modified script is shown below. Running this script with our prior and observed data, we find a posterior mean of 5.6×10^{-5}/hour and a 90% credible interval of $(7.7 \times 10^{-6}, 1.6 \times 10^{-4})$.

```
model   {
x ~ dpois(mean.poisson)                 # Poisson distribution for number of events
mean.poisson <- lambda*time.hr          # Poisson parameter
lambda ~ dlnorm(mu, tau)                # Lognormal prior distribution for lambda
# Calculate tau from lognormal error factor
tau <- 1/pow( log(prior.EF)/1.645, 2)
# Calculate mu from lognormal mean and error factor
mu <- log(prior.mean) - pow(log(prior.EF)/1.645, 2)/2
}

data
list(x=2, time.hr=11000, prior.mean=0.00001, prior.EF=7.6)
```

- Script 38. WinBUGS script for first part of Example 29.
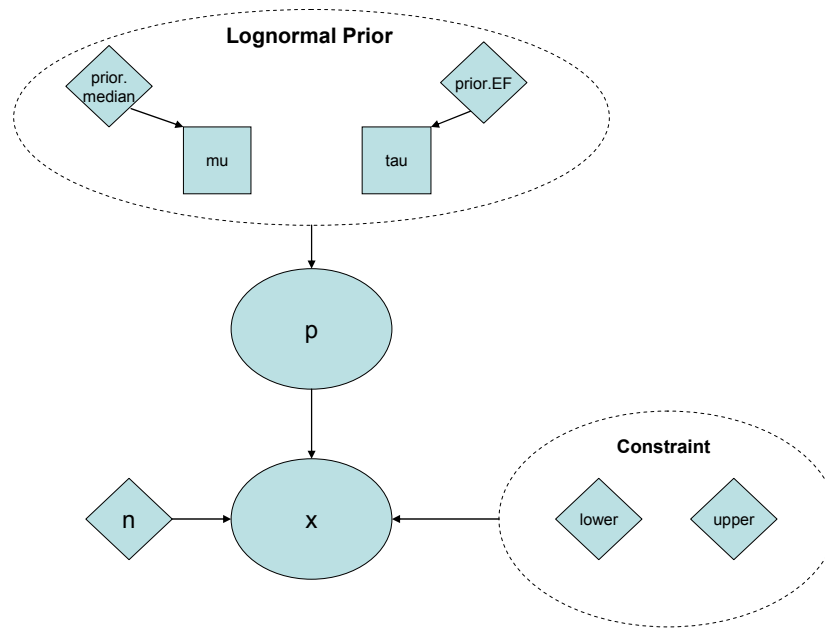
• Figure 4-81. DAG representing Script 38

To factor in the uncertainty in the number of radiator plugging events, the analyst must specify a probability for each possible failure count from 2 to 7. One convenient way of doing this is to work on a scale where 100 is very likely, 50 is indeterminate, and 10 is unlikely. We can then normalize by dividing each possibility by the total point count, which in this case is 360. Thus, in this example, 2 and 3 failures would be assigned 100 points, because the analyst was pretty sure that three of the events met the PRA failure criterion. Four, five, and six failures would each be assigned 50 points, because the analyst couldn't tell, and seven failures would be assigned 10 points, because the analyst was pretty sure that one of the events had not met the failure criterion. In the WinBUGS script below, the required normalization is carried out as part of the script.

```
model  {
    for(i in 1:N) {
        x[i] ~ dpois(mean.poisson[i])     # Poisson distribution for number of events
        mean.poisson[i] <- lambda[i]*time.hr      # Poisson parameter
        lambda[i] ~ dlnorm(mu, tau)                # Lognormal prior distribution for lambda
    }
lambda.avg <- lambda[r]          # Monitor this node for posterior average distribution

r ~ dcat(p.analyst[])
p.analyst[1] <- 100/360               # Probability of 2 failures
p.analyst[2] <- 100/360               # Probability of 3 failures
p.analyst[3] <- 50/360                # Probability of 4 failures
p.analyst[4] <- 50/360                # Probability of 5 failures
p.analyst[5] <- 50/360                # Probability of 6 failures
p.analyst[6] <- 10/360                # Probability of 7 failures

# Calculate tau from lognormal error factor
tau <- 1/pow( log(prior.EF)/1.645, 2)

# Calculate mu from lognormal mean and error factor
mu <- log(prior.mean) - pow(log(prior.EF)/1.645, 2)/2
}
data
list(x=c(2,3,4,5,6,7), time.hr=11000, prior.mean=0.00001, prior.EF=7.6, N=6)
```

- Script 39. WinBUGS script for incorporating uncertainty into Poisson event count.
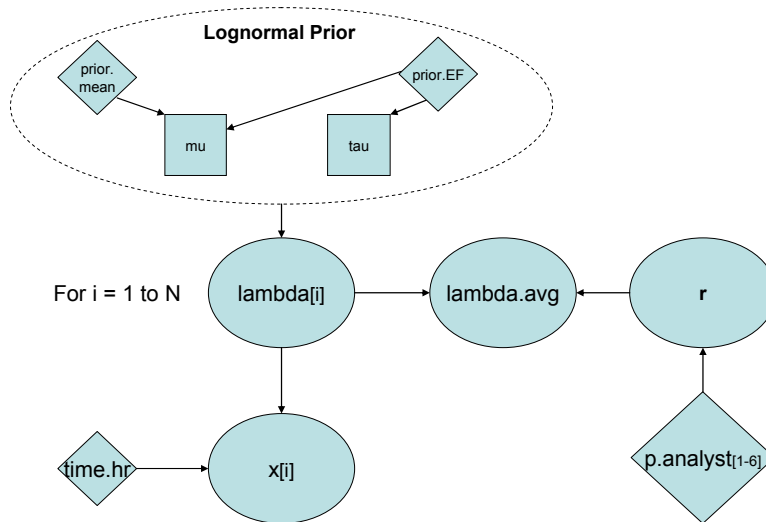
- Figure 4-82. DAG representing Script 39.

Note that the node **lambda.avg** in Script 39 reflects the posterior distribution for lambda, averaged over the analyst's subjective distribution for the failure count. In effect, WinBUGS estimates **six** different posterior distributions, corresponding to the possible failures (two to seven), and then averages these distributions over node **p.analyst**, which encodes the analyst's epistemic uncertainty as to the actual number of failures.

Running this script in the usual way gives a posterior mean for **lambda.avg** of 1.5×10^{-4}/hour and a 90% credible interval of (1.5×10^{-5}, 4.3×10^{-4}), a significant difference from the results above where the failure count was taken to be exactly two.

4.8.3 Censored Data for Random Durations

Consider the following example. A programmable logic controller (PLC) in the avionics package for the ATCS is being tested. Ten PLCs are placed in test, and each test is to be run for 1,000 hours. If a PLC fails before the end of the test, its failure time is recorded. Assume that two of the PLCs failed during the test, at 395 and 982 hours. The other eight PLCs were still operating when the test was terminated at 1,000 hours. How can we use this information to carry out Bayesian inference for the PLC failure rate, lambda, assuming that the time to failure can be described by an exponential distribution?

The key to dealing with censored times, for any likelihood distribution, is the **C(lower, upper)** construct in WinBUGS.[1] In cases where a failure time was not recorded, this tells WinBUGS to impute a failure time from the specified aleatory distribution, between the bounds specified by lower and upper in the construct. For our example above, lower would be equal to 1,000 hours, and upper would be omitted, so we would have the following line in the WinBUGS script: **t[i] ~ dexp(lambda)C(1000,)**.

In the data portion of the script, a value of **NA** is entered for each censored failure time (time not recorded). Because "lower" is a vector, the failure time is entered if a failure time is observed, otherwise **NA** is entered. The script to analyze this example is shown below.

```
model  {
    for(i in 1:N) {
        # Define likelihood function with Type I censoring
        t[i] ~ dexp(lambda)C(lower[i],)
    }
lambda ~ dgamma(0.0001, 0.0001)        # Jeffreys prior for lambda
}

data
list(t=c(982,394.7,NA,NA,NA,NA,NA,NA,NA,NA),
lower=c(982,394.7,1000,1000,1000,1000,1000,1000,1000,1000), N=10)

Inits
list(lambda=0.001)
list(lambda=0.0001)
```

• Script 40. WinBUGS script for modeling censored random durations.

---

[1] It is also possible to code this as I(lower, upper).
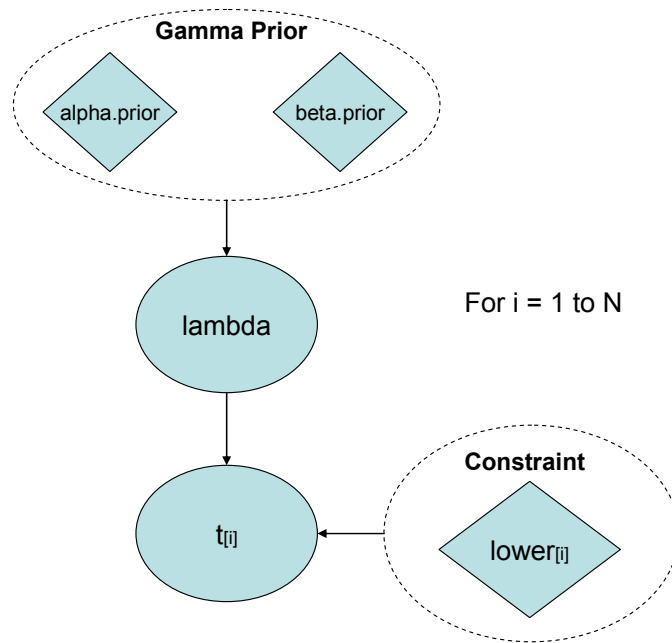
- Figure 4-83. DAG representing Script 40.

Running this script in the usual way gives a posterior mean for lambda of 2.1×10^{-4}/hour and a 90% credible interval of (3.8×10^{-5}, 5.1×10^{-4}).

- Example 30. Modeling uncertainty in fire suppression times (censored data).
  > Consider the following data on time to suppress a fire (in minutes) in a critical area: < 1, 5, < 10
  > 15, 4, 20, 30, 3, 30-60, 25. Find the probability that a fire in this area will last longer than 20
  > minutes, the assumed time to damage for critical components in the area.

Solution
We start by specifying an aleatory model for the time to suppress the fire. The simplest model is the
exponential distribution, so we will start with that, and examine alternative models later. We will use a
Jeffreys prior distribution for lambda, the suppression rate in the exponential distribution. The
WinBUGS script below illustrates how to encode the censored data for suppression time. Node
time.rep estimates the suppression time, unconditional upon lambda, and node **prob.fail** estimates the
probability that the fire will last for at least 20 minutes without being suppressed.

```
model {
    for (i in 1 : N) {
        # Exponential dist. for suppression times
        time.supp[i] ~ dexp(lambda)C(lower[i], upper[i])
    }
time.rep ~ dexp(lambda)
lambda ~ dgamma(0.0001,0.0001)        # Diffuse prior for exponential parameter
prob.fail <- exp(-lambda*time.crit)        # Probability of non-suppression by time.crit
time.crit <- 20
}

data
list(time.supp=c(NA, 5, NA, 15, 4, NA, 3, NA, 25), N=9)
list(lower=c(0, 5, 0, 15, 4, 20, 3, 30, 25), upper=c(1, 5, 10, 15, 4, 30, 3, 60, 25))

inits
list(lambda = 0.1) #initial values
```

• Script 41. WinBUGS script for Example 30, interval-censored random durations with exponential likelihood.
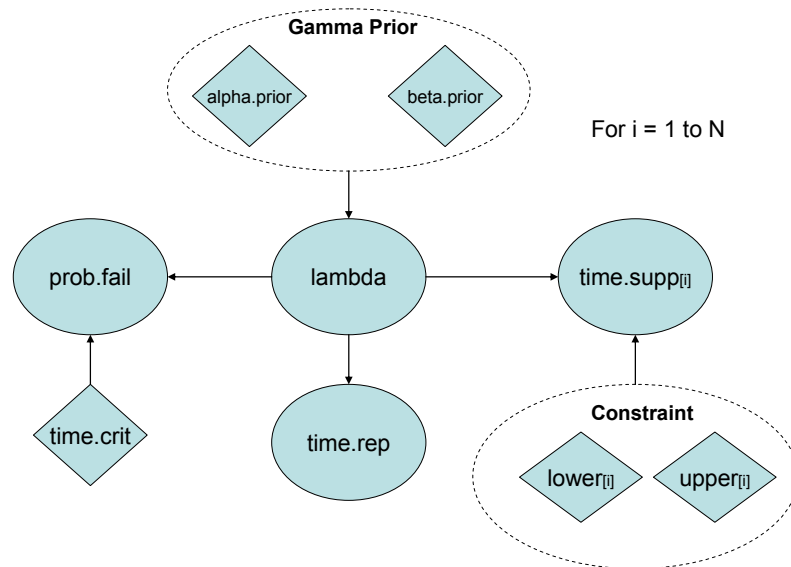
• Figure 4-84. DAG representing Script 41.

Running this script in the usual way gives a mean non-suppression probability at 20 minutes of 0.25, with a 90% interval of 0.09, 0.46). For later comparison with an alternative model, the DIC was found to be 51.3.

The Weibull distribution is an alternative aleatory model to the exponential distribution, as described in Sec. 4.6.1.2. The script below shows how to encode a Weibull likelihood function and estimate the probability of non-suppression at 20 minutes. Recall that if the shape parameter, alpha, equals one, then the Weibull distribution reduces to the exponential distribution. Running the script below in the usual way, gives the following posterior distribution for the Weibull shape parameter.
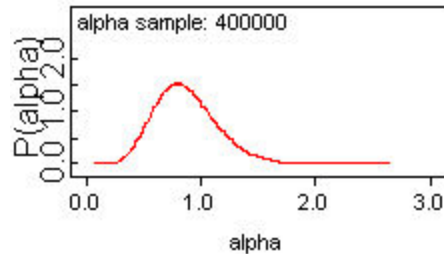
- Figure 4-85. Posterior distribution for alpha in Weibull distribution.

As this distribution is centered about one, the exponential distribution is probably adequate. The probability of non-suppression with the Weibull model has a posterior mean of 0.26 and a 90% credible interval of (0.09, 0.47). These are essentially the same results as obtained with the simpler exponential model above. Finally, the DIC for the Weibull model is 52.5, slightly larger than the exponential model, allowing us to conclude that the exponential is a better model than the Weibull distribution.

```
model {
    for (i in 1 : N) {
        # Weibull distribution for suppression times
        time.supp[i] ~ dweib(alpha, scale)C(lower[i], upper[i])
    }
time.rep ~ dweib(alpha, scale)
alpha ~ dgamma(0.0001,0.0001)          # Diffuse prior for exponential parameter
scale ~ dgamma(0.0001,0.0001)
prob.fail <- exp(-scale*pow(time.crit, alpha)) # Probability of non-suppression by time.crit
time.crit <- 20
}

data
list(time.supp=c(NA, 5, NA, 15, 4, NA, 3, NA, 25), N=9)
list(lower=c(0, 5, 0, 15, 4, 20, 3, 30, 25), upper=c(1, 5, 10, 15, 4, 30, 3, 60, 25))

inits
list(alpha = 1, scale=10) #initial values
list(alpha=0.5, scale=15)
```

- Script 42. WinBUGS script for Example 30, interval-censored random durations with Weibull likelihood.
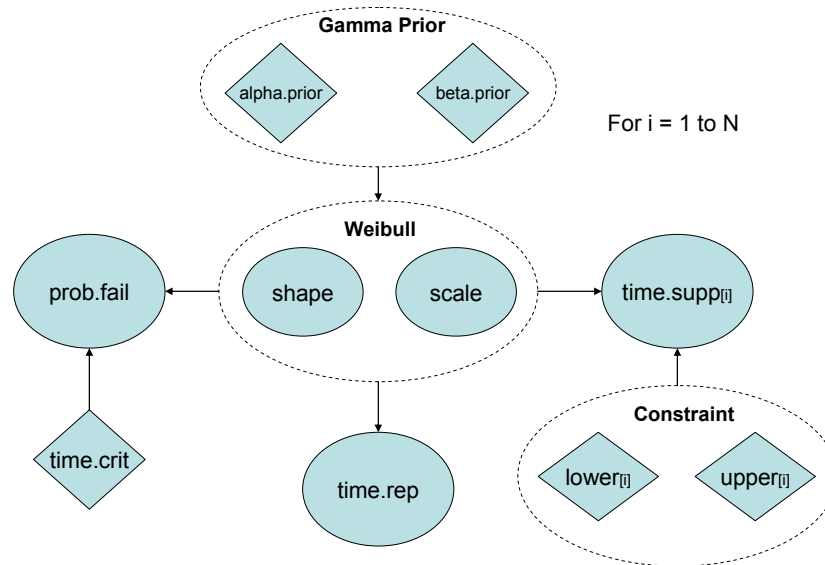
- Figure 4-86. DAG representing Script 42.

4.8.4 Legacy and Heritage Data

In this section we show how Bayesian inference methods can be used in reliability assessment of systems under development. The methodology provides the flexibility for engineers to incorporate reliability considerations into the design or redesign of systems, even when data on the actual system under design is lacking or very limited in quality and quantity. For instance, in many practical situations, systems are not designed from scratch, but evolve in a series of evolutionary designs and modifications. While the reliability of a system at different stages in such design evolutions is not necessarily identical, earlier designs typically provide a meaningful indication of the reliability behavior of future generations, on the basis that their design, manufacturing, and operation are largely the same. The reliability behavior observed for existing systems can therefore well serve as a useful source of information, as long as the differences between system designs are taken into account.

The various types of information that might be available for assessing the reliability of a system during its life cycle include:

Stage I: Initial Design: During the conceptual and early design phase of a new system reliability assessment can be informed by any combination of the following types of information:
- Expert Opinion: various forms of expert opinions are discussed in Section 4.11
- Reliability Analysis: estimates based on system modeling and probabilistic analysis of physics
- Generic Data: estimates or data from similar systems, other applications
- Legacy or Heritage Data: field and/or test data from the most representative previous system designs. These systems are called *legacy or heritage* systems**.**

Stage II: Development Phase: This Phase normally involves several rounds of design modifications, tests, and fixes. The tests may include accelerated-life tests, and the entire process constitutes what is commonly known as the *reliability growth program*. The types of information include:
- Design credit,
- Failure mode fix effectiveness,
- Scaling of life data collected from accelerated-life testing.

Stage III. Field Operation: Actual operational data collected during actual system mission and field use constitutes the most relevant data for assessing and updating the actual reliability of the system.

This section focuses on Stages I and II in the system life-cycle; inference during Stage III has been addressed in previous sections.

4.8.4.1 Inference for Initial Design (Stage I) —In developing "baseline" estimates from legacy system information the analyst needs to account for any differences in the designs to establish the degree of relevance or degree of applicability of the legacy information to the new system design.

- Example 31. Developing failure probability for new parachute design from legacy data.

> An evolutionary parachute design has been developed for a new mission. Legacy data for parachute failure are available from the Apollo program and the Russian Soyuz program.
>
> In the Apollo program there was 1 failure of a parachute in 15 missions (3 parachutes are used for each mission). The Soyuz program likewise experienced 1 parachute failure in 93 missions (1 parachute for each mission).
>
> Use this legacy data to develop a prior distribution for p, the probability of parachute failure for the new design.

Solution

For both Apollo and Soyuz, we will assume that the number of parachute failures can be described by a binomial distribution. We will first obtain the posterior distribution for p in each of these legacy programs by updating a Jeffreys prior. Our prior for the new parachute design will then be a weighted average (as described in Section 4.8.2) of these legacy posterior distributions, with the weights representing the analyst's judgment as to the applicability of the information from each of the two legacy programs.

For this example, we assume that the judgment is that the Apollo information is more applicable, so it will have a weight of 0.7 (indicating higher applicability), giving a weight of 0.3 to the Soyuz information. The WinBUGS script shown below implements this analysis.

```
subscript 1 = Soyuz: 1 failure in 93 demands
subscript 2 = Apollo: 1 failure in 45 demands (15 missions, 3 parachutes per mission)
model {
    for(i in 1:2) {
        x[i] ~ dbin(p[i], n[i])          # Binomial likelihood for each legacy program
        p[i] ~ dbeta(0.5, 0.5)           # Jeffreys prior for p
  }
p.new <- p[r]                            # Prior distribution for new design is weighted average
                                         # of posteriors from legacy programs

r ~ dcat(w[])
}


data
# Weights and observed failures c(Soyuz failures, Apollo failures)
list(w=c(0.3,0.7), x=c(1,1), n=c(93, 45))
```

- Script 43. WinBUGS script to develop prior distribution as weighted average of legacy information.
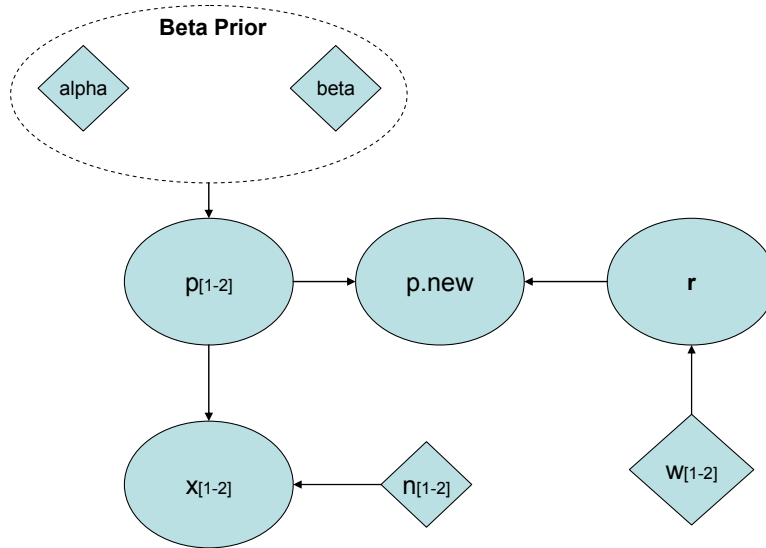
• Figure 4-87. DAG representing Script 43.

Running this script in the usual way gives a prior mean for the new design of 0.028 with a 90% credible interval of $(2.9 \times 10^{-3}, 0.076)$. Note that the mean is the weighted average of the mean of each legacy posterior distribution.

If desired, uncertainty about the weighting factors can be introduced simply by assigning a distribution to each weight. For example, assume the analyst felt that the weighting factor for the Apollo information could be anywhere between 0.5 and 0.9. The WinBUGS Script 44 encodes this belief. Running the script in the usual way gives a prior mean for the new design of 0.028 with a 90% credible interval of $(2.9 \times 10^{-3}, 0.076)$. Note that the results have not changed significantly.

```
subscript 1 = Soyuz: 1 failure in 93 demands
subscript 2 = Apollo: 1 failure in 45 demands (15 missions, 3 parachutes per mission)
model {
    for(i in 1:2) {
        x[i] ~ dbin(p[i], n[i])        # Binomial likelihood for each legacy program
        p[i] ~ dbeta(0.5, 0.5)         # Jeffreys prior for p
  }
p.new <- p[r]                          # Prior distribution for new design is weighted average
                                       # of posteriors from legacy programs
r ~ dcat(w[])
w[2] ~ dunif(0.5, 0.9)                 # Uncertainty in Apollo weighting factor
w[1] <- 1 - w[2]                       # Weights must sum to 1
}

data
# Weights and observed failures c(Soyuz failures, Apollo failures)
list(x=c(1,1), n=c(93, 45))
```

• Script 44. WinBUGS script to develop prior distribution as weighted average of legacy information with uncertainty as to weighting factors.
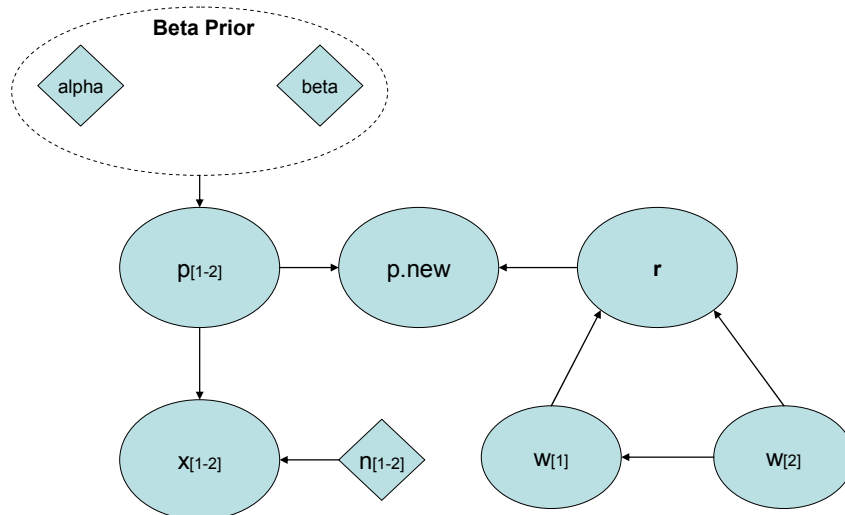
- Figure 4-88. DAG representing Script 44.

4.8.4.2 Inference During Development Phase (Stage II) —This phase normally involves several rounds of design modifications, tests, and fixes. The tests may include accelerated-life tests, and the entire process constitutes what is commonly known as the *reliability growth program*. As problems in the design are encountered during testing, they are fixed, so that data may not be strictly applicable. Various *ad hoc* approaches to the problem exist, such as "failure-discounting[1]" and the so-called STRATCOM[2] method, in which the designer specifies an anticipated lower bound on system reliability. In implementing either of these approaches, it is important to factor in uncertainty about the discounting factor or the lower bound in the STRATCOM approach in order to avoid nonconservatively low estimates of PRA parameters such as p. We illustrate this by revisiting Example 31.

- Example 31. continued, incorporating prototype test data.

> Assume that test data are available for a prototype of the new parachute design.
> There have been 3 failures of the prototype in 13 tests. The analyst judges that a discount factor of 0.1 is appropriate. Find the posterior mean and 90% credible interval for the prototype using the prior developed in the second part of the example above, which accounted for uncertainty in the analyst's weighting factors for the legacy information from Apollo and Soyuz.

Solution
The WinBUGS script below is used to analyze this problem, with no uncertainty applied to the discounting factor. Running this script in the usual way gives a posterior mean of 0.027 and a 90% credible interval of ((3.7 × 10^{-3}, 0.067, nearly the same as the prior mean and 90% interval obtained earlier. This is expected as the prototype data are sparse. Note that had no discounting factor been applied, the posterior mean would have been 0.074 and the 90% credible interval would have been (0.026, 0.14).

---

[1] Failure discounting is the process by which an integer number of failures is reduced to a lower, possibly non-integer, value.
[2] See, for example, Guikema 2005.

```
subscript 1 = Soyuz: 1 failure in 93 demands
subscript 2 = Apollo: 1 failure in 45 demands (15 missions, 3 parachutes per mission)
model {
    for(i in 1:2) {
        x[i] ~ dbin(p[i], n[i])       # Binomial likelihood for each legacy program
        p[i] ~ dbeta(0.5, 0.5)        # Jeffreys prior for p
    }
p.proto <- p[r]                       # Prior distribution for new design is weighted average
                                      # of posteriors from legacy programs

r ~ dcat(w[])
w[2] ~ dunif(0.5, 0.9)                # Uncertainty in Apollo weighting factor
w[1] <- 1 - w[2]                      # Weights must sum to 1
x.disc <- x.proto*f.disc             # Apply discounting factor to prototype failures
x.disc ~ dbin(p.proto, n.proto)      # Binomial distribution for discounted prototype failures
}


data
# Weights and observed failures c(Soyuz failures, Apollo failures)
list(x=c(1,1), n=c(93, 45))
list(x.proto=3, n.proto=13, f.disc=0.1)
```

• Script 45. WinBUGS script for discounting prototype failure data using discounting factor.
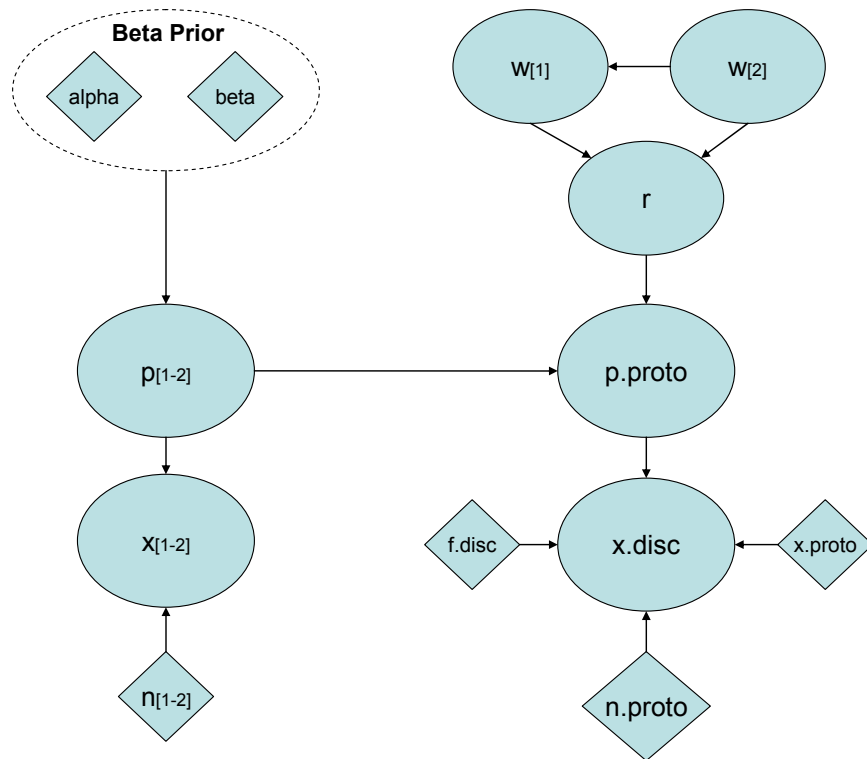
• Figure 4-89. DAG representing Script 45.

The "failure-discounting" approach has been widely criticized within the PRA community, on a variety of grounds, including the **non-conservative** estimates it tends to produce, and the failure to account for

uncertainty in the discounting factor, which is really the same criticism restated. It is not possible to account for uncertainty in the discounting factor by assigning it a distribution, as WinBUGS will have conflicting information about the aleatory model for the observed prototype failures: one model will be the observed number of failures (3 in our example) multiplied by the distribution for the discounting factor and the other will be a binomial distribution.

We can avoid the problem of conflicting information by treating the observed prototype failures as uncertain, as described in Section 4.8.2. The analyst must develop a discrete distribution for the number of prototype failures. In this example, the possible failure counts are 0, 1, 2, or 3. To simplify the scripting required in WinBUGS, we will replace the weighted-average prior from the Apollo and Soyuz legacy information with a lognormal distribution, preserving the mean and error factor. In the WinBUGS script below, a probability of 0.6 has been assigned to 0 failures, 0.2 to 1 failure, and 0.1 to 2 and 3 failures. The values for prior.mean, upper, and lower were obtained using the results from Script 45. Running this script in the usual way gives a posterior mean for the prototype failure probability of 0.036, with a 90% credible interval of $(3.7 \times 10^{-3}, 0.12)$.

```
subscript 1 = Soyuz: 1 failure in 93 demands
subscript 2 = Apollo: 1 failure in 45 demands (15 missions, 3 parachutes per mission)

model {
    for(j in 1:K) {
        x.proto[j] ~ dbin(p.proto[j], n.proto)      # Binomial dist. for prototype failures
        p.proto[j] ~ dlnorm(mu, tau)      # Lognormal approximation to prototype prior
    }
p.proto.avg <- p.proto[s]                          # Monitor this node
s ~ dcat(disc[])
mu <- log(prior.mean) - pow(log(sqrt(upper/lower))/1.645, 2)/2
tau <- pow(log(sqrt(upper/lower))/1.645, -2)
prior.mean <- 0.027
upper <- 0.075
lower <- 0.003
}

data
list(x.proto=c(0,1,2,3), n.proto=13, K=4, disc=c(0.6, 0.2, 0.1, 0.1))
```

• Script 46. WinBUGS script for treating prototype failures as uncertain data.

For j = 1 to K
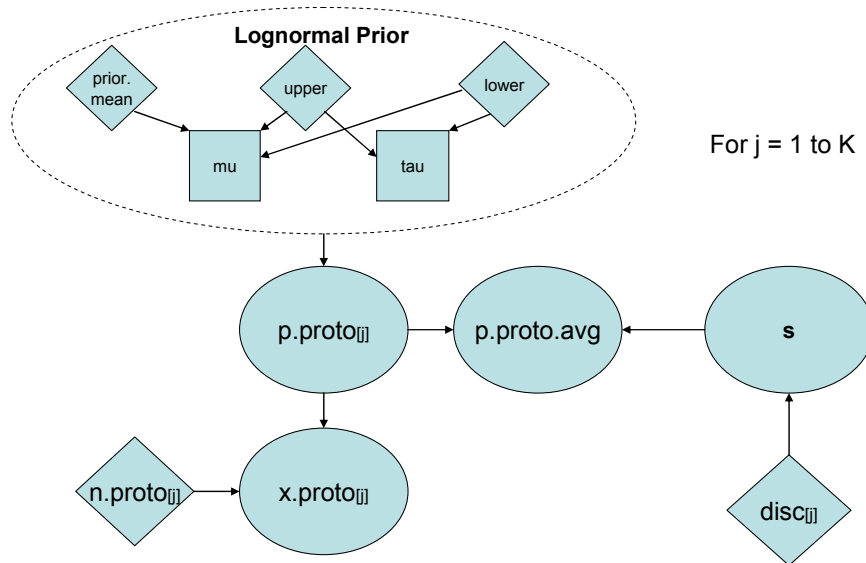
• Figure 4-90. DAG representing Script 46

Uncertainty in the weighting factors can be added to the script above if desired. The most straightforward way of doing this is by assigning a Dirichlet distribution to the weights. The marginal distribution of each weight will then be a beta distribution with parameters $alpha_i$ and $alpha_{tot} - alpha_i$, where $alpha_{tot}$ is just the sum of each $alpha_i$. The analyst can choose a value for $alpha_{tot}$, with smaller values leading to larger uncertainties for each weight; a value of 10 should work if there are only a few weights, as in this example. We choose $alpha_i$ so that the mean of each weight is equal to the point estimate in Script 46 above. In our example, we will have the inputs shown below.

• Table 7. Alpha weighting factor parameter values for Example 31.

Component of alpha	Mean weight	5th percentile	95th percentile
6	0.6	0.34	0.83
2	0.2	0.04	0.43
1	0.1	0.006	0.28
1	0.1	0.006	0.28

To simplify the scripting required in WinBUGS, we will replace the weighted-average prior from the Apollo and Soyuz legacy information with a lognormal distribution, preserving the mean and error factor. This produces the WinBUGS script shown below. Running this script in the usual manner gives a posterior failure probability for the prototype of 0.036, with a 90% credible interval of $(3.7 \times 10^{-3}, 0.12)$. Thus, in this example, the uncertainty for the weights has had essentially no impact on the results.

```
model {
    for(j in 1:K) {
        x.proto[j] ~ dbin(p.proto[j], n.proto)      # Binomial dist. for prototype failures
        p.proto[j] ~ dlnorm(mu, tau)      # Lognormal approximation to prototype prior
    }
p.proto.avg <- p.proto[s]                           # Monitor this node
s ~ dcat(disc[])
disc[1:4] ~ ddirch(alpha[])
#Replace legacy prior with lognormal distribution to simplify script
mu <- log(prior.mean) - pow(log(sqrt(upper/lower))/1.645, 2)/2
tau <- pow(log(sqrt(upper/lower))/1.645, -2)
prior.mean <- 0.027
upper <- 0.075
lower <- 0.003
}

data
list(x.proto=c(0,1,2,3), n.proto=13, K=4, alpha=c(6,2,1,1))
```

- Script 47. WinBUGS script for treating prototype failures as uncertain data, including uncertainty in weighting factors.
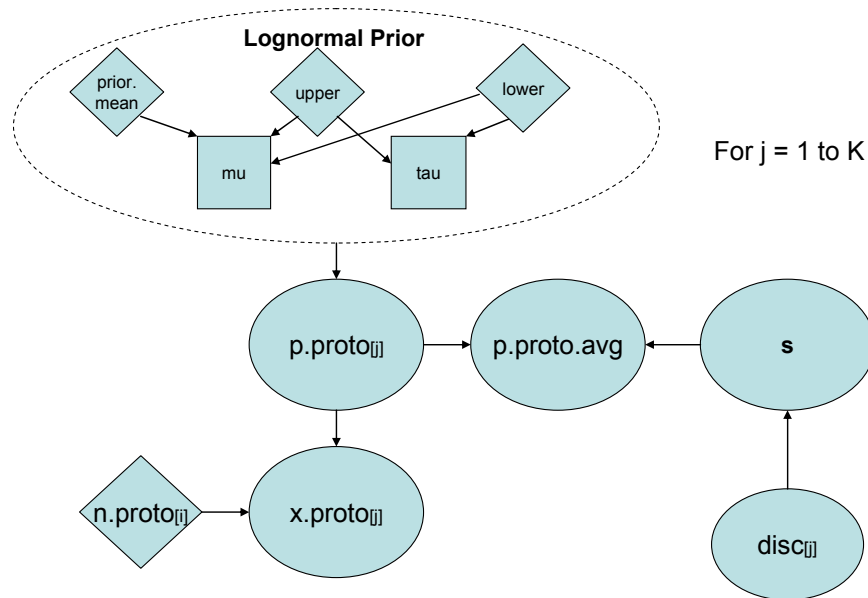
- Figure 4-91. DAG representing Script 47

4.9 Bayesian Regression Models

Sometimes a parameter in an aleatory model, such as p or lambda, can be described by observable quantities such as pressure, mass, and temperature. For example, in the case of a pressure vessel, very high pressure and high temperature may be leading indicators of failures. In such cases, information about the explanatory variables can be used in the Bayesian inference paradigm to estimate p or lambda. We illustrate the possibility of a Bayesian regression approach with an example that estimates the probability of O-ring failure in the solid-rocket booster motors of the space shuttle.

- Example 32. Modeling O-ring distress probability as a function of leak test pressure and launch temperature.

  Table 8 shows data on O-ring thermal stress collected during launches prior to the 1986 launch of the Challenger that led to disastrous failure of the O-rings. Each shuttle has three primary and three secondary O-rings, with one of each type having to fail to cause a disaster such as the Challenger.

- Table 8. Space shuttle field O-ring thermal distress data.

Flight	Distress[1]	Temp (^{o}F.)	Press (psig)
1	0	66	50
2	1	70	50
3	0	69	50
5	0	68	50
6	0	67	50
7	0	72	50
8	0	73	100
9	0	70	100
41-B	1	57	200
41-C	1	63	200
41-D	1	70	200
41-G	0	78	200
51-A	0	67	200
51-C	2	53	200
51-D	0	67	200
51-B	0	75	200
51-G	0	70	200
51-F	0	81	200
51-I	0	76	200
51-J	0	79	200
61-A	2	75	200
61-B	0	76	200
61-C	1	58	200

[1]Thermal distress is defined to be erosion of the O-ring or blow-by of hot gases. The table shows the number of distress events for each launch. There are six field O-rings on the shuttle, so the number of distress events is an integer in the interval [0, 6].

Our aleatory model for O-ring distress during launch will be a binomial distribution with n = 6, since there are 6 O-rings on the shuttle. The unknown parameter of interest is **p**, the probability of distress of an O-ring, assumed to be the same for each of the six O-rings.

Engineering knowledge leads us to believe that p will vary as pressure and temperature vary (i.e., the leading indicators mentioned earlier), so we construct a model for p with pressure and temperature as

explanatory variables. We consider the potential explanatory model [Dalal et al, 1989]: logit(p) = a + b*temp + c*press.

The WinBUGS script (Script 48) shows how the regression model is encoded. This script also predicts the probability of O-ring distress at the Challenger launch temperature of 31° F and pressure of 200 psig, information that presumably would have been of great value in the Challenger launch decision.

```
model  {
    for(i in 1:K) {
        distress[i] ~ dbin(p[i], 6)
        # Regression model by Dalal et al with temp and pressure
        logit(p[i]) <- a + b*temp[i] + c*press[i]
        distress.rep[i] ~ dbin(p[i], 6)      # Replicate values for model validation
        diff.obs[i] <- pow(distress[i] - 6*p[i], 2)/(6*p[i]*(1-p[i]))
        diff.rep[i] <- pow(distress.rep[i] - 6*p[i], 2)/(6*p[i]*(1-p[i]))
    }
chisq.obs <- sum(diff.obs[])
chisq.rep <- sum(diff.rep[])
p.value <- step(chisq.rep - chisq.obs)
distress.31 ~ dbin(p.31, 6) # Predicted number of distress events for launch 61-L
logit(p.31) <- a + b*31 + c*200     # Regression model with temp and pressure
                                    # from day of the accident

# Prior distributions
a ~ dnorm(0, 0.000001)
b ~ dnorm(0, 0.000001)
c ~ dnorm(0, 0.000001)
}
data
list(
   distress=c(0,1,0,0,0,0,0,0,0,1,1,1,0,0,2,0,0,0,0,0,0,2,0,1),
   temp=c(66,70,69,68,67,72,73,70,57,63,70,78,67,53,67,75,70,81,76,79,75,76,58),

press=c(50,50,50,50,50,50,100,100,200,200,200,200,200,200,200,200,200,200,200,2
00,200,200,200),
   K=23
   )

inits
list(a=5, b=0, c=0)
list(a=1, b=-0.1, c=0.1)
```

• Script 48. WinBUGS script for logistic regression model of O-ring distress probability with pressure and temperature as explanatory variables.
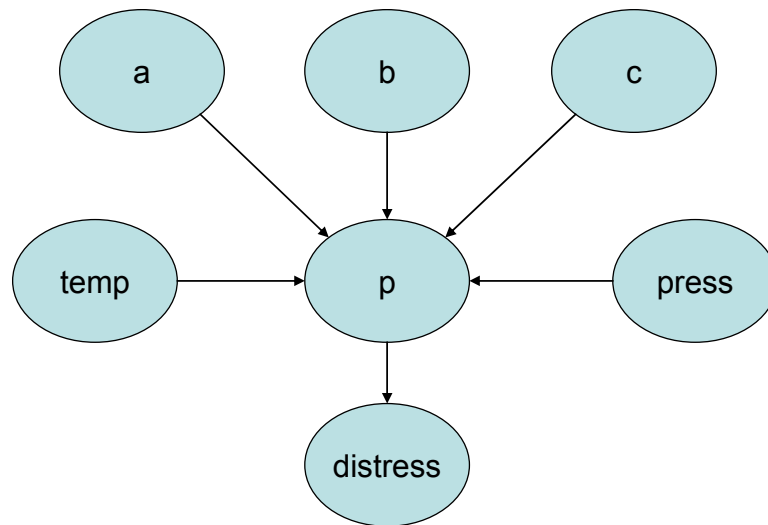
• Figure 4-92. DAG representing Script 48.

Running this script in the usual way gives the results in Table 9. The expected number of **distress** events at 31° F is about four (indicating the distress of four of six O-rings) and the probability that one of the six O-rings experiences thermal stress during a launch at that temperature is about 0.71.

• Table 9. Results for shuttle O-ring distress model with temperature and pressure as explanatory variables.

Parameter	Mean	95% Interval
a (intercept)	2.22	(-4.78, 9.91)
b (temperature coeff.)	-0.10	(-0.20, -0.02)
c (pressure coeff.)	0.01	(-0.004, 0.03)
p.31	0.71	(0.14, 0.99)

4.10 Using Higher-Level Data to Estimate Lower-Level Parameters

This section examines Bayesian inference in the case where we are interested in failure rates and probabilities for components that make up a system, but the observed data are at multiple "levels," including the subsystem and system level (rather than just at the component level like in earlier sections). We will use the system fault tree and the resulting expression of system failure probability in terms of constituent component failure probabilities to carry out the analysis in WinBUGS. Consider the simple fault tree shown in Fig. 6-9 of the NASA PRA Procedures Guide, reproduced in Figure 4-93.
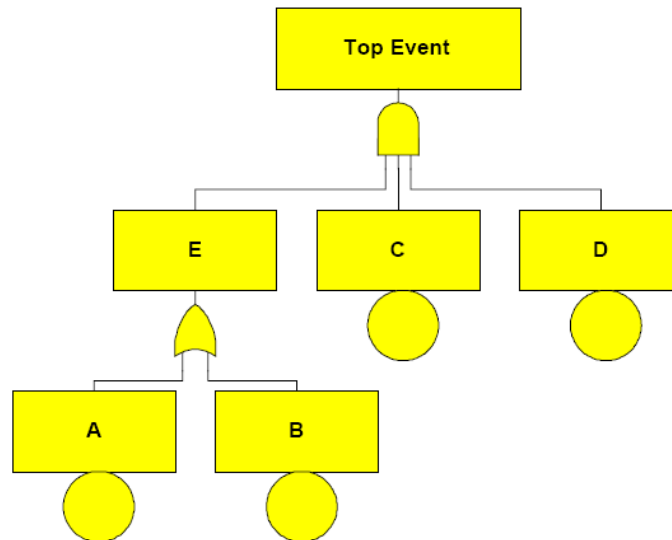
Figure 6-9: Typical Fault Tree Structure

- Figure 4-93. Example fault tree from NASA PRA Procedures Guide.

Assume we have data for Gate E (subsystem) and the Top Event (system). Specifically, assume there have been:

- Three failures at Gate E in 20 demands (of this subsystem, not the overall system)
- One failure of the Top Event in 13 demands of the system. Note that the subsystem (Gate E) was demanded during the 13 system demands (since the only way to have a system failure includes a component failure from this subsystem).

Further assume that basic events "A" and "B" represent a standby component, which must change state upon a demand. Assume that "A" represents failure to start of this standby component and "B" represents failure to run for the required time, which we will take to be 100 hours. Assume basic events "C" and "D" represent normally operating components. We will assume the information about the basic event parameters as shown in Table 10.

• Table 10. Basic event parameters for the example fault tree.

Basic Event	Parameters of Interest	Prior Distribution
A	p	Lognormal, mean=0.001, EF=5
B	Lambda Mission time=100 hours	Lognormal, median=0.005/hr, EF=5
C	Lambda Mission time = 100 hours	Lognormal, mean=0.0005/hr, EF=10
D	Lambda Mission time = 100 hours	Lognormal, mean=0.0005/hr, EF=10

We are using the same distribution to represent epistemic uncertainty in the parameters for events C, and D, so we will have to account for this state-of-knowledge dependence in the Bayesian inference. The WinBUGS Script 49 (an associated DAG shown in Figure 4-94) will be used to find posterior distributions for p and lambda in this model, using the available operational data (specified on the previous page) at the subsystem and system level.

```
model  {
# This is system (fault tree top) observable (x.TE number of failures)
x.TE ~ dbin(p.TE, n.TE)
x.Gate.E ~ dbin(p.Gate.E, n.Gate.E)
p.TE <- p.Gate.E *p.C*p.D          # Probability of Top Event (from fault tree)
p.Gate.E <- p.A + p.B - p.A*p.B   # Probability of Gate E from fault tree
p.C <- p.ftr                       # Account for state-of-knowledge dependence between C & D
p.D <- p.ftr                       # by setting both to the same event
p.B <- 1 - exp(-lambda.B*time.miss)
p.ftr <- 1 - exp(-lambda.ftr*time.miss)

# Priors on basic event parameters
p.A ~ dlnorm(mu.A, tau.A)
mu.A <- log(mean.A) - pow(log(EF.A)/1.645, 2)/2
tau.A <- pow(log(EF.A)/1.645, -2)
lambda.B ~ dlnorm(mu.B, tau.B)
mu.B <- log(median.B)
tau.B <- pow(log(EF.B)/1.645, -2)
lambda.ftr ~ dlnorm(mu.ftr, tau.ftr)
mu.ftr <- log(mean.ftr) - pow(log(EF.ftr)/1.645, 2)/2
tau.ftr <- pow(log(EF.ftr)/1.645, -2)
}

data
list(x.TE=1, n.TE=13, x.Gate.E=3, n.Gate.E=20, time.miss=100)
list(mean.A=0.001, EF.A=5, median.B=0.005, EF.B=5, mean.ftr=0.0005, EF.ftr=10)
```

• Script 49. WinBUGS script for using higher-level information for fault tree shown in Figure 4-93.
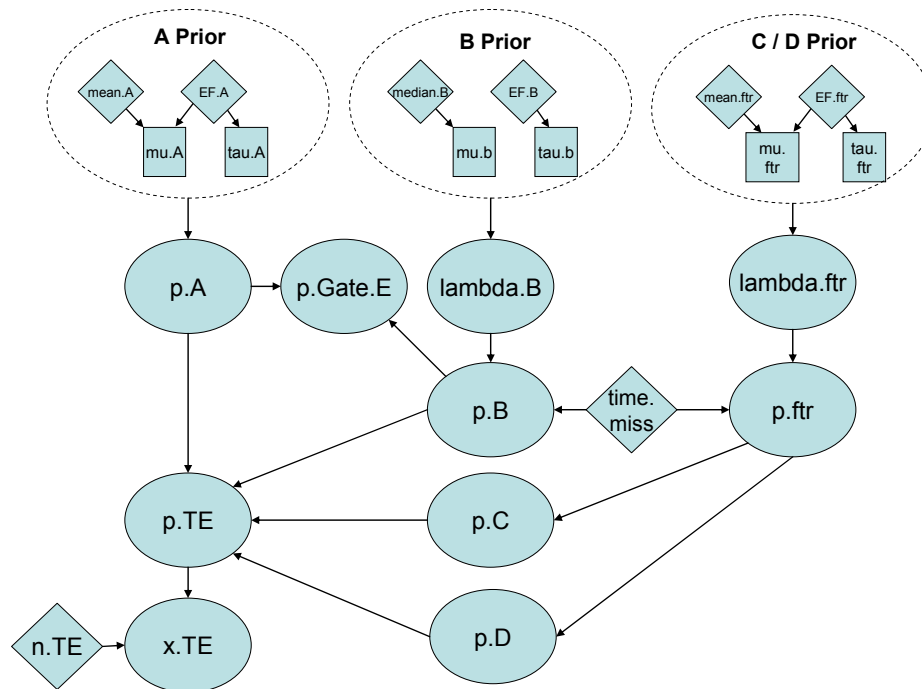
- Figure 4-94. DAG representing Script 49.

Running Script 49 in the usual way gives the results in Table 11, which are compared to the prior means. These results (see Figure 4-95) show that the observed data has significantly **increased** the shared failure rate used for basic events C and D. The estimate of the failure rate of component B has decreased from its prior mean value.

- Table 11. Analysis results for example fault tree model.

Parameter	Prior Mean	Posterior Mean	90% Interval
p.A	1.0×10^{-3}	1.0×10^{-3}	$(1.2 \times 10^{-4}, 3.0 \times 10^{-3})$
lambda.B	8.1×10^{-3}/hr	2.4×10^{-3}/hr	$(1.1 \times 10^{-3}, 4.3 \times 10^{-3})$
lambda.ftr (shared by event C and D)	5×10^{-4}/hr	3.7×10^{-3}/hr	$(4.5 \times 10^{-4}, 1.0 \times 10^{-2})$
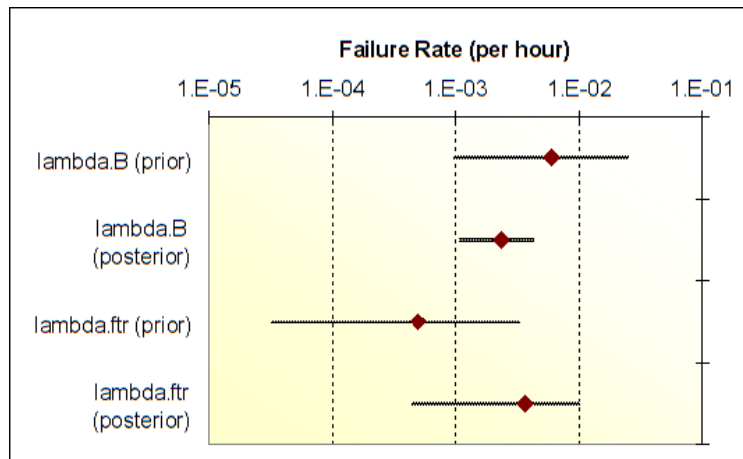
- Figure 4-95. Whisker plot of the prior and posterior failure rates for basic events B (lambda.B) and C/D (lambda.ftr).

4.11 Using Information Elicited from Experts

The focus of this section is on methods for using information obtained from experts. Whether one has information from one or several experts, one would usually need to develop a representative estimate for use in the analysis. When the opinions of several experts are elicited, methods are needed to form the "aggregated" or "consensus" opinion. The formulation is quite simple conceptually. Expert opinion is simply treated as information about the unknown parameter of interest. The information is then used to update the analyst's own (prior) estimate through Bayes theorem. We will describe some of the basic techniques for a number of important classes of problems, but the coverage will not be exhaustive as the techniques for certain classes of problems tend to become very complicated without any assurance of significant improvement in the resulting estimates.

4.11.1 Using Information from a Single Expert

In this case the expert provides an estimate for an unknown parameter of interest, e.g., lambda in the Poisson distribution. To use this information to update the analyst's prior distribution for lambda via Bayes' Theorem, a likelihood function is needed for the information obtained from the expert. When the epistemic uncertainty in parameter values spans several orders of magnitude, as is common in PRA, a lognormal distribution is a convenient likelihood function. The tau parameter (logarithmic precision) in the lognormal distribution will represent the analyst's assessment of the expert's expertise: small values of tau correspond to low confidence (high uncertainty) and vice versa. A bias factor can also be introduced if desired, with bias less than one meaning the analyst believes the expert tends to underestimate the true value of lambda, and a factor greater than one means the expert tends to overestimate the true value.

• Example 33. Developing a prior for level sensor failure probability based on information from a single expert.

> The analyst's prior estimate for the failure of a level sensor in the ATCS is 10^{-6}/hour. The analyst uses a lognormal distribution with this estimate as the median, and an error factor of 10 to describe the uncertainty. The level sensor vendor provides an estimate of the mean time to failure (MTTF) for the level sensor. The vendor's estimate for the MTTF is 500,000 hours.
> Develop a posterior distribution for the failure rate of the level sensor using these two sources of information.

Solution

The first step is to convert the MTTF estimate provided by the vendor into an estimate of the failure rate. This is done by taking the reciprocal of the MTTF estimate. The analyst must assess an uncertainty factor on the vendor's estimate, representing their confidence in the estimate provided by the vendor. Assume that the analyst is not very confident in the vendor's estimate, and assesses an error factor of 10. He also believes that the expert tends to overestimate the MTTF, that is, he underestimates the failure rate, so a bias factor of 0.75 is assessed by the analyst. The WinBUGS script below is used to analyze this problem. Running the script in the usual way gives a posterior mean for lambda of 2.7×10^{-6}/hour with a 90% credible interval of $(3.2 \times 10^{-7}, 8.4 \times 10^{-6})$. If the analyst thought the vendor tended to underestimate the MTTF, that is, overestimate lambda, he would use a bias factor greater than one. Changing the bias factor to 5, for example, changes the posterior mean to 1.0×10^{-6}/hour with a 90% credible interval of $(1.3 \times 10^{-7}, 3.2 \times 10^{-6})$.

```
model {
lambda.star ~ dlnorm(mu, tau)                    # Lognormal likelihood for information
from expert
lambda.star <- 1/MTTF
mu <-log( lambda*bias)
tau <- pow(log(EF.expert)/1.645, -2)
lambda ~ dlnorm(mu.analyst, tau.analyst)         # Analyst's lognormal prior for lambda
mu.analyst <- log(prior.median)
tau.analyst <- pow(log(prior.EF)/1.645, -2)
}
data
list(MTTF=500000, bias=0.75, EF.expert=10, prior.median=0.000001, prior.EF=10)
```

• Script 50. WinBUGS script for lognormal (multiplicative error) model for information from single expert.
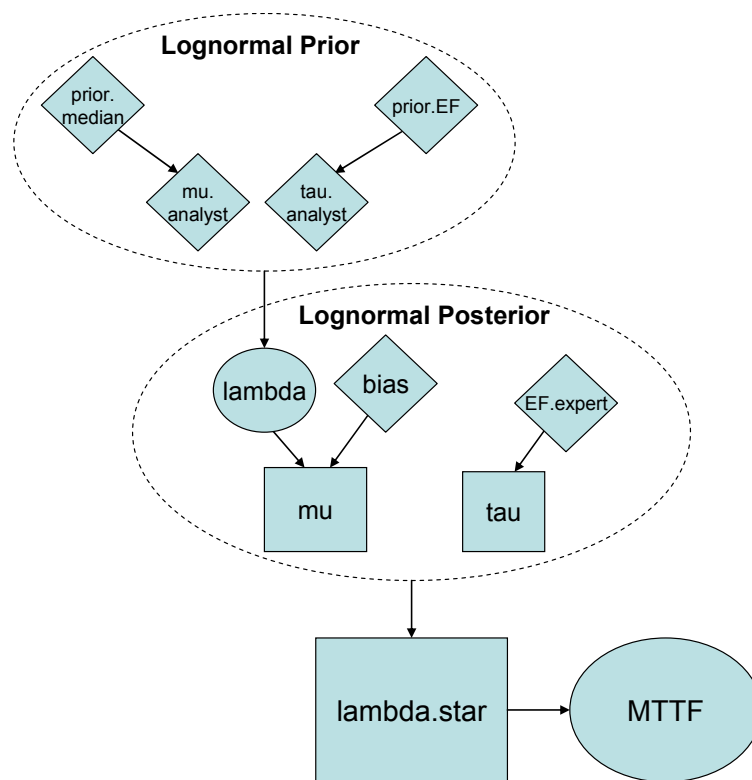
• Figure 4-96. DAG representing Script 50.

4.11.2 Using Information from Multiple Experts

Cases encountered in practice often involve more than one expert. When multiple experts are involved the main question concerns the method of aggregation or pooling to form a single representative or aggregate estimate from the multiple expert estimates. A number of *ad hoc* approaches have been used for combining information from multiple experts, such as taking the geometric average (arithmetic average of the logarithms) and taking the low and high estimates as the 5th and 95th percentiles of a

lognormal distribution. A justification commonly given for these *ad hoc* approaches is that analytical techniques need not be more sophisticated than the pool of estimates (experts' opinions) to which they are applied. Therefore, a simple averaging technique (equal weights) has often been judged satisfactory as well as efficient, especially when the quantity of information collected is large.

The Bayesian approach of the previous section can be expanded to include multiple experts. Basically, the methods of Section 4.5 can be used to develop a prior distribution representing the variability among the experts. While mathematically cumbersome, this is straightforward to encode in WinBUGS, as the following example illustrates.

• Example 34. Developing a prior for pressure transmitter failure using information from multiple experts.

> Six estimates are available for failure rate of pressure transmitters. These estimates along with the assigned measure of confidence are listed below. The analyst wishes to aggregate these estimates into a single distribution that captures the variability among the experts.

Expert	Estimate (per hour)	Confidence Measure (Error Factor)
1	3.0×10^{-6}	3
2	2.5×10^{-5}	3
3	1.0×10^{-5}	5
4	6.8×10^{-6}	5
5	2.0×10^{-6}	5
6	8.8×10^{-7}	10

As in the previous section, the likelihood function for each expert will be assumed to be lognormal. A diffuse prior is placed on lambda (actually on the logarithm of lambda). The WinBUGS script below is used to develop a distribution for lambda, accounting for the variability among the six experts. Running the script in the usual way gives a posterior mean for lambda of 6.5×10^{-6}/hour, with a 90% credible interval of (3.4×10^{-6}, 1.1×10^{-5}).

```
model  {
    for(i in 1:N) {
        lambda.star[i] ~ dlnorm(mu, tau[i])
        tau[i] <- pow(log(EF[i])/1.645, -2)
    }
mu ~ dflat()              # Diffuse prior on mu
lambda <- exp(mu)         # Monitor this node for aggregate distribution
}

data
list(lambda.star=c(3.E-6, 2.5E-5, 1.E-5, 6.8E-6, 2.E-6, 8.8E-7), EF=c(3,3,5,5,5,10), N=6)
inits
list(mu=-10)
list(mu=-5)
```

• Script 51. WinBUGS script for combining information from multiple experts using multiplicative error model (lognormal likelihood).
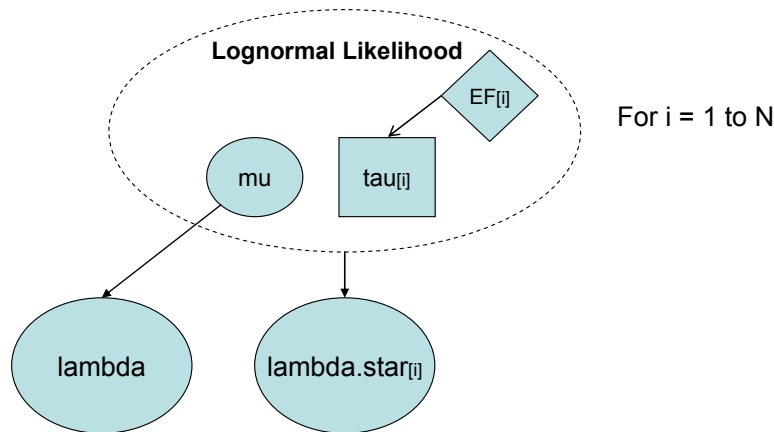
- Figure 4-97. DAG representing Script 51.

4.11.3 Bayesian Inference from Heritage Data

Often, heritage data from prior hardware use is used to predict the performance of the hardware in a new mission. The heritage data can be from prior missions, prototype testing, manufacturer's claims, or terrestrial use. The following example covers the process of predicting a failure rate of a hardware item using flight data from other missions.

Assume mission experts determine that the hardware item's reliability is sensitive to three attributes:

1. Manufacturer
2. Design process
3. Hardware Qualification program

A review of flight records identified four missions where the spacecraft flew the hardware item and had at least one attribute in common with the mission for which the PRA is being performed. Data from other missions that the hardware item was used on are not considered. Two of the missions are considered similar enough to call siblings and can be pooled (e.g. the two Mars Exploration Rovers). The mission experts deliberate as a team and determine the following:

- Mission 1 is completely applicable to the new mission. The same manufacturer, design, and qualification program were used.
- Mission 2 is partially applicable to the new mission in that there was a different manufacturer and design process, but the qualification program was identical to the current mission.
- Missions 3 and 4 are partially applicable to the new mission in that the same manufacturer and design process were used, but the qualification program was substantially different.

The probabilities that the evidence from the four heritage missions apply to the new mission are elicited from the experts. The hardware performance and probability that the evidence applies are summarized in Table 12.

- Table 12. Heritage component flight evidence and the probability that it applies to the new mission under consideration.

Mission	Flight Evidence		Probability Evidence Applies
	Failures	Operating Hours	
1	0	6.41×10^5	100%
2	1	5.98×10^6	40%
3	0	2.73×10^4	60%
4	0	2.73×10^4	60%

There are sixteen evidence sets resulting from the four missions. Each evidence set has a unique applicability to the current mission. The following rules are agreed upon by the experts:

- Mission 1 is completely applicable, therefore any evidence set that excludes Mission 1 is not considered.
- Mission 2 is partially applicable and all sets in which it is included or excluded can be considered.
- Missions 3 and 4 are pool-able information and partially applicable, therefore only evidence sets in which both are either included or excluded are considered.

Using these rules, Table 13 summarizes the evidence sets and their calculated combined probabilities.

- Table 13. Evidence set applicability.

Evidence Sets (which Missions Apply)				Evidence Set
Mission 1	Mission 2	Mission 3	Mission 4	Probability
Yes	Yes	Yes	Yes	24%
Yes	Yes	Yes	No	0%
Yes	Yes	No	Yes	0%
Yes	No	Yes	Yes	36%
No	Yes	Yes	Yes	0%
Yes	Yes	No	No	16%
Yes	No	Yes	No	0%
No	Yes	Yes	No	0%
Yes	No	No	Yes	0%
No	Yes	No	Yes	0%
No	No	Yes	Yes	0%
Yes	No	No	No	24%
No	Yes	No	No	0%
No	No	Yes	No	0%
No	No	No	Yes	0%
No	No	No	No	0%

The WinBUGS script below is used to determine the posterior failure rate based on the evidence set probabilities developed by the experts. A Jeffreys prior is updated for each evidence set and the evidence set probabilities are used to average the resulting four posterior distributions.
Running the script in the usual way gives a posterior mean for lambda of 5.4×10^{-7}/hour, with a 90% credible interval of 7.1×10^{-9}/hr to 2.2×10^{-6}/hr.

Four possibilities considered:
1. **All four mission applicable, probability = 0.24**
2. **Second mission not applicable, probability = 0.36**
3. **Linked missions (3 and 4) not applicable, probability = 0.16**
4. **Only first mission applicable, probability = 0.24**

```
model  {
    for(i in 1:4)  {
        x[i] ~ dpois(mean.pois[i])
        mean.pois[i] <- lambda[i]*time[i]
        lambda[i] ~ dgamma(0.5, 0.0001)          # Jeffreys prior
    }
lambda.avg <- lambda[r]                          # Average posterior distribution
r ~ dcat(p[])
# Formulate datasets
x[1] <- sum(flight.failure[])
time[1] <- sum(flight.time[])
x[2] <- flight.failure[1] + sum(flight.failure[3:4])
time[2] <- flight.time[1] + sum(flight.time[3:4])
x[3] <- sum(flight.failure[1:2])
time[3] <- sum(flight.time[1:2])
x[4] <- flight.failure[1]
time[4] <- flight.time[1]
}

data
list(flight.failure=c(0, 1, 0, 0), flight.time=c(6.41E+5, 5.98E+6, 2.73E+4, 2.73E+4), p=c(0.24,
0.36, 0.16, 0.24))
```

- Script 52. Posterior averaged predicted failure rate using heritage data.
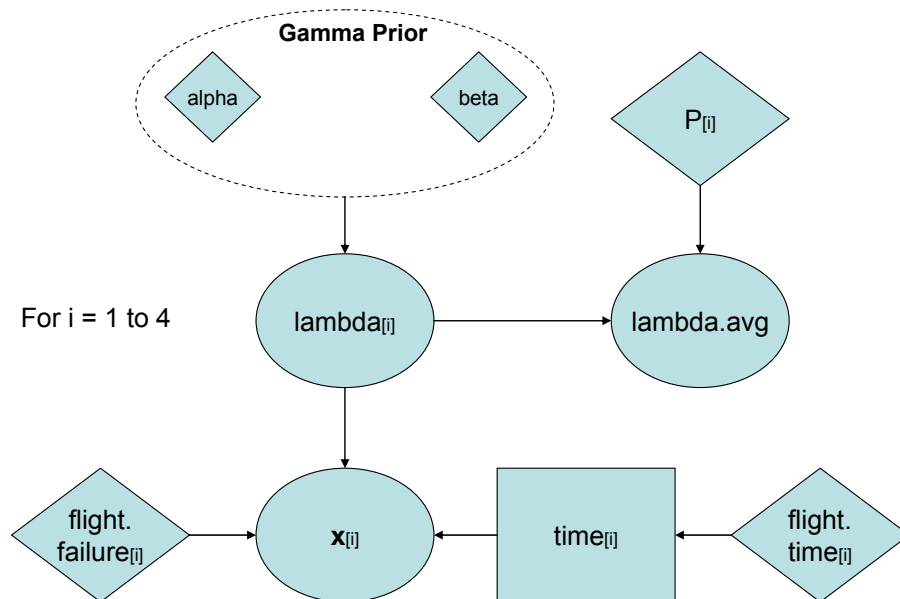
- Figure 4-98. DAG representing Script 52.

4.12 Case Study

Earlier, in Chapter 3, we described a thermal control system to be used as a case study. This ATCS architecture is shown in Figure 4-99. In this section, we will follow the system through its hypothetical life cycle, as defined in NPR 7120.5D "NASA Space Flight Program and Project Management Requirements," in order to further demonstrate the methods described in Section 4.

- Approval (initial design)
- Implementation (final design and modifications)
- Integration (final testing)
- Operation
- Decommissioning
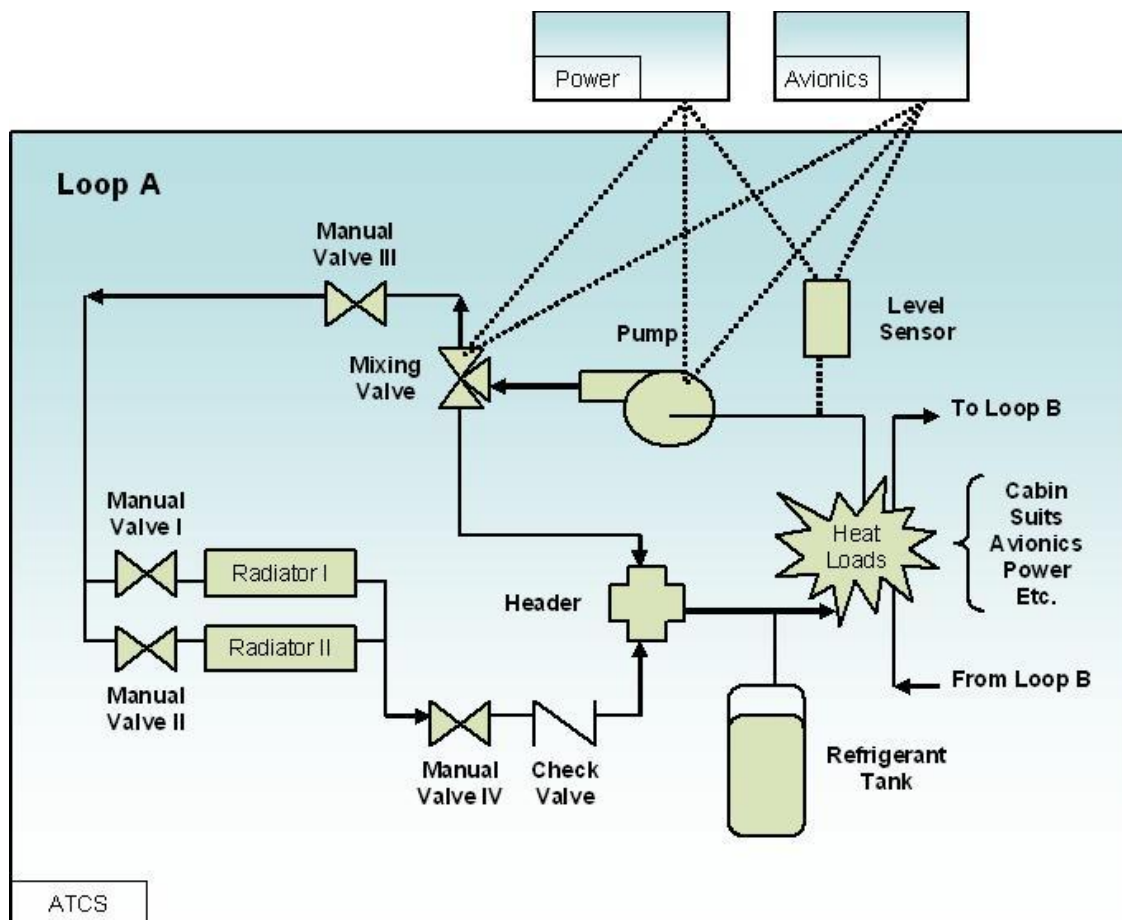
- Figure 4-99. Active thermal control system diagram.

4.12.1 Case Study Initial Design Phase

During the initial formulation of even a relatively simple system such as the ATCS, a variety of information is known to different degrees. For example, we may know that we need "a" system to remove heat loads, but we may initially have only a vague idea as to specifics of the system. Conversely, at the same time, we may know that the heat removal system will have one or more radiators – we may have years of experience applicable to radiator performance. In any case though, we need to make inferences on the ATCS as it evolves from design to operation and these inferences should reflect what is known (and, by definition, what is not known).

Our initial design specification may simply reflect a set of minimal conditions for successful functionality of the ATCS, for example:

> The ATCS should be a dual-loop system capable of removing expected heat loads in the crew module.

Given this limited specification and general knowledge of the design and operation of cooling systems, one might propose modeling the ATCS at the initial design phase as shown in Figure 4-100.
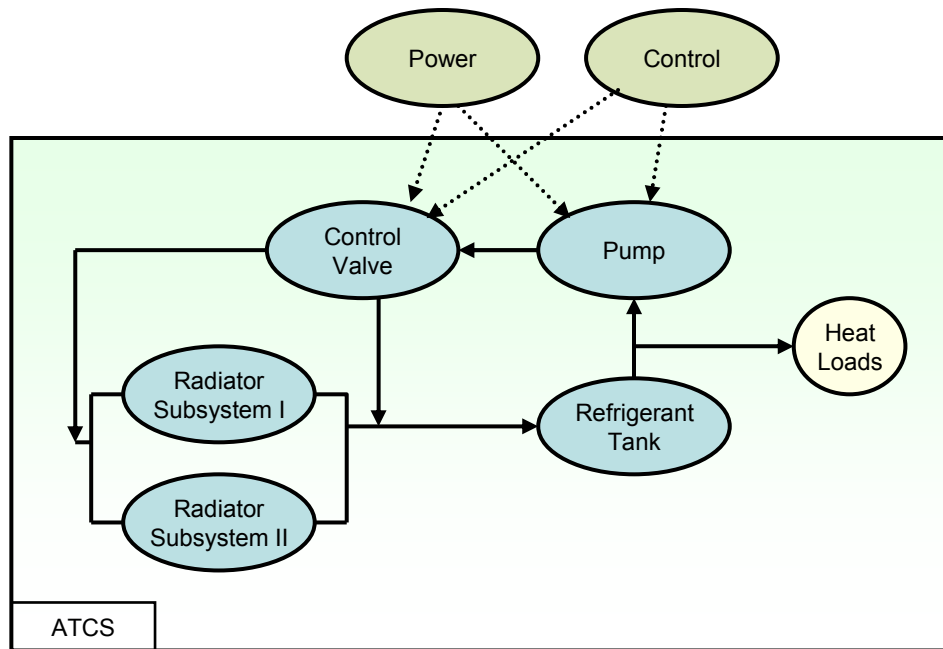
• Figure 4-100. Initial simplified diagram for the ATCS

In this design, we have two supporting systems (power and control) and five components. For each of these components or supporting systems, we need three types of information:

- Engineering information
- Aleatory models
- Data

We will discuss a variety of differing degrees of information for these three categories as we evaluate each support system and component.

4.12.1.1 DC Power Subsystem—The unreliability of the DC power subsystem affects the unreliability of the ATCS. This subsystem may eventually be modeled as a more detailed system itself, but initially we do not have detailed information or design requirements for this subsystem. Consequently, we will represent power failure as a single failure entity (an undeveloped event in a fault tree of the ATCS) and collect information on the subcomponents most likely used in a full design.

The failure taxonomy for the DC power subsystem is:

Mission Directorate: **Exploration Systems Mission Directorate**
 └ Program: **IntraSolar Exploration**
  └ Project: **New Manned Vehicle (NMV)**
   └ Subsystem: **Crew Module**
    └ Subsystem: **Active Thermal Control**
     └ Subsystem: **DC Power**
      └ Failure Mode: **Fails to Operate**
       └ Failure Mechanism: **Any Active Operational Failure in Time**
        └ Failure Cause: **Any**

If we return to Figure 4-99 for guidance, we see that the component type is "operating" and the failure type is "fails to operate." This path through the flow chart is shown in Figure 4-101. We are up to the question of data type. For the DC power subsystem, we do not have data available. One may believe that, at this point, we are finished with the inference process. However, following the process described in this report, even in the case of no data, proves useful for the following reasons:

- Knowledge of the types of data for future inference (in this case either failure times or failure counts and operating time) will help to guide data collection specific to this component.
- Knowledge of the types of aleatory models (in this case either Poisson or exponential) will help to guide the probabilistic modeling of the DC power system in the ATCS within the context of a PRA.
- Knowledge of the prior distribution will provide the epistemic uncertainty applicable to the PRA probabilistic modeling for the DC power subsystem.

Not having data results in a Bayesian inference process where the posterior is **equivalent** to the prior. This posterior, though, is what is used in the PRA for the initial design.
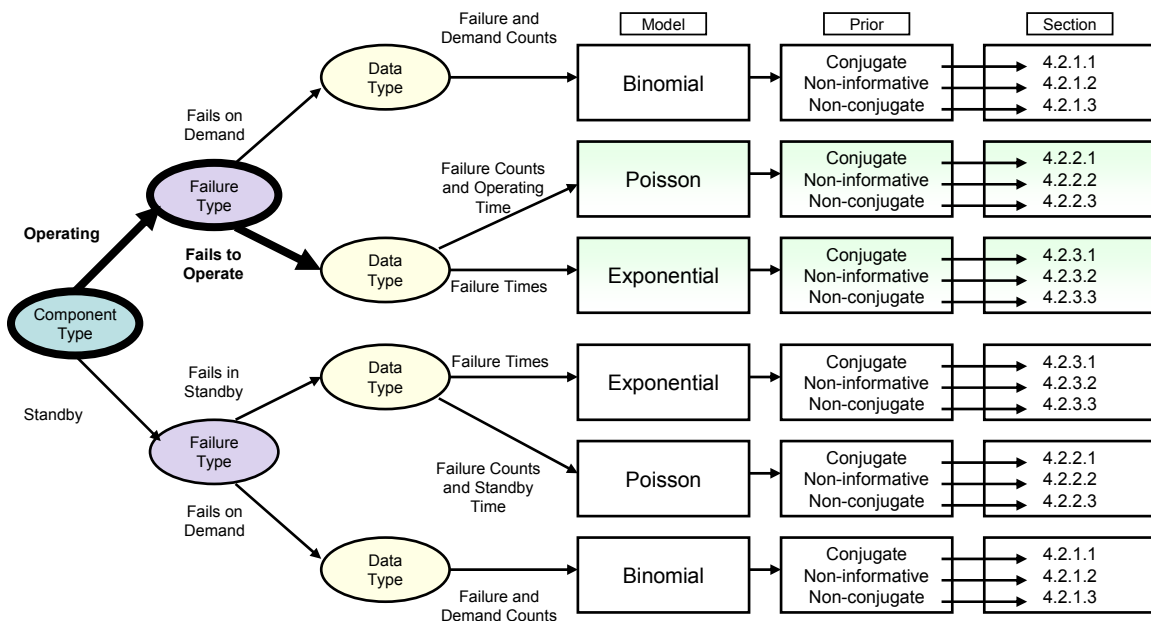
• Figure 4-101. Inference flow diagram path for the DC power subsystem components inference process.

Step 1: The prior
We gather information related to DC power systems. We know that a DC power system consists of parts such as fuses, wiring, buses, batteries, and breakers. Engineering information is collected and includes:

- Fuse failures from GSFC database indicate 2 failures in 64,163 hours of satellite operation.
- Fuse failures from the NUCLARR database indicate 6 failures in 48,577,000 hours of energized operation. The estimates provided by these two sources are quite different, so the information on the fuses cannot simply be pooled.
- GSFC AC bus data indicates zero failures in 64,163 hours of satellite operation.
- Low voltage (<600 volt) AC bus fails to operate, NUCLARR, zero failures in 2,170,000 hours of energized operation. With zero observed failures, the bus information can be pooled.
- DC power circuit breaker spurious transfer open, NUCLARR, zero failures in 916,000 hours of energized operation.
- DC power circuit breaker (indoor), IEEE-Std 500-1984, failures in 10^6 hours (low = 0.02, recommended = 0.14, high = 0.65). Since this is a standard, the information can be interpreted as a failure rate with the values of mean = 1.4E-7 per hour, lower bound = 2.0E-8 per hour and upper bound = 6.5E-7 per hour.
- Dry cell battery failure, IEEE-Std 500-1984, failures in 10^6 hours (recommended = 27.0, high = 31.0). Again, this is a standard which can be interpreted as a failure rate, this time omitting the zero value lower bound (mean = 2.70E-5 per hour, upper bound = 3.10E-5).
- Lithium battery failure, NASA non-proprietary data, 1.5×10^{-5} per hour.

Step 2: The data
Operational data on the DC power system is not available. Consequently, the posterior result for the failure-to-operate inference process will be equivalent to the prior information.

Step 3: Bayesian calculation to estimate DC power component failure rates
- ☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) contains a single unknown parameter.
- ☑ The prior information is heterogeneous for the fuses and homogeneous for the buses. It is considered known with certainty for the fuses and buses but there is epistemic uncertainty for the circuit breaker and battery failure counts.
- ☑ The fuse and the bus sets of information are both presented as failures over total operating hours. Script 53 shows the fuse analysis. Running this script for the fuses in the usual manner produces the results shown in Figure 4-103.

The categorical distribution is used to give a weighted value to the priors based on perceived applicability to the current model. Note that the result for lambda.avg is biased towards lambda[1] by giving the GSFC satellite information a 0.85 weight when compared to the NUCLARR information. This weighting was chosen due to the more applicable environment of the in-orbit operation. The NUCLARR information is weighted at 0.15 and the sum of the weights must equal 1.0.

```
Component    : ATCS DC Power System Fuses
Model        : Poisson
Prior        : Noninformative
subscript 1 = Fuses in GSFC, 2 failures in 64,163 hr
subscript 2 = Fuses in NUCLARR, 6 failures in 48,577,000 hr

model  {
for (i in 1:2)     {
x[i] ~ dpois(mean.pois[i])
mean.pois[i] <- lambda[i] * time[i]        # Poisson parameter
lambda[i] ~ dgamma(0.5, 0.0001)            # Jeffreys Priors
}
lambda.avg <- lambda[r]                     # Assigning weights to information
r ~ dcat(p[])
}

data
list(x= c(2, 6), time= c(64163, 48577000), p= c(0.85, 0.15))
```

- Script 53. WinBUGS script for two weighted priors with failures in time of operation.
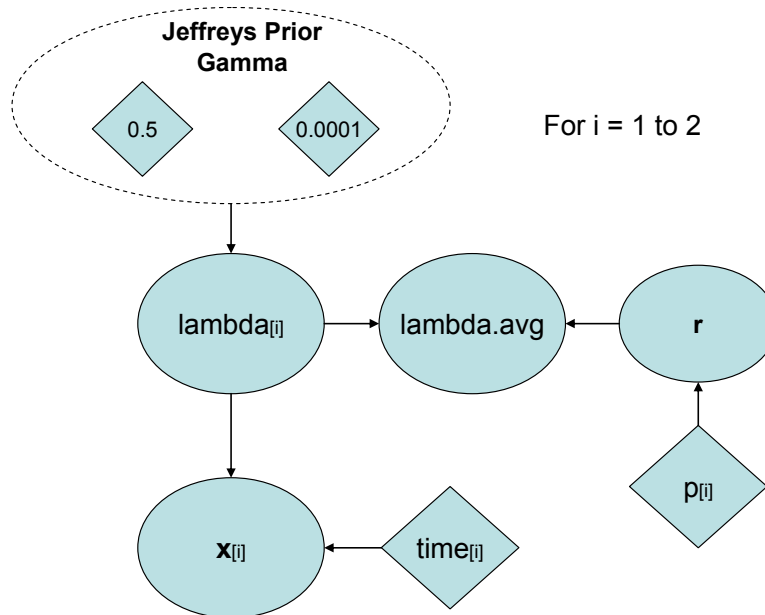
- Figure 4-102. DAG representing Script 53.

	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
lambda[1]	3.896E-5	2.465E-5	7.973E-8	9.028E-6	3.389E-5	8.609E-5	1001	100000
lambda[2]	1.335E-7	5.268E-8	1.782E-10	6.032E-8	1.267E-7	2.303E-7	1001	100000
lambda.avg	3.313E-5	2.665E-5	9.058E-8	1.068E-7	2.904E-5	8.287E-5	1001	100000

- Figure 4-103. ATCS fuse nodes results.

Even though both sources of bus information contain zero failures, and thus could be pooled, NUCLARR contains much more information than does the GSFC data. Pooling the data may lead to an unrealistically low estimate and an overly narrow uncertainty range if the NUCLARR experience is not very representative of in-orbit operation. Thus, the analyst may still wish to weight these two information sources in the same manner as for the fuses above. In this case, Script 53 can be used for the bus component as well. Changing the information to the bus data provides a mean of 6.7×10^{-6} per hour with a 90% credible interval of 1.0×10^{-8} to 2.8×10^{-5} per hour. This is compared to a mean of 2.2×10^{-7} per hour and 90% credible interval of (8.8×10^{-10}, 8.6×10^{-7}) from simply pooling the two sources.

The circuit breaker information is both in the form of a number of failures in a time period and a failure rate with a mean and an upper and lower bound. Script 54 shows the circuit breaker analysis with a 0.60 weight given to the NUCLARR database over the standard due to the actual runtime of the NUCLARR information. Running this script in the usual manner results in a lambda.avg of 4.1×10^{-7} per hour with a 90% credible interval of 6.0×10^{-9} to 1.7×10^{-6} per hour.

```
Component      : ATCS DC Power System Circuit Breakers
Model          : Poisson / Lognormal
Prior          : Informative, Uncertain
    subscript 1 = DC Power Circuit Breaker,NUCLARR, 0 failures in 916,000 hr. operation
    subscript 2 = DC Power Circuit Breaker IEEE-Std 500-1984 (lower = 2.0E-8 per hour,
              high = 6.5E-7 per hour)
model  {
x ~ dpois(mean.pois)
mean.pois <- lambda[1] * time            # Poisson parameter
lambda[1] ~ dgamma(0.5, 0.0001)          # Jeffreys priors for Poisson
lambda[2] ~ dlnorm(mu, tau)              # Lognormal prior for given upper and lower limit
mu <- log( sqrt(upper * lower))
tau <- pow(log(sqrt(upper/lower))/1.645, -2)
lambda.avg <- lambda[r]                   # Assignment of weights
r ~ dcat(p[])
}

data
list(p=c(0.60, 0.40), lower= 2.0E-8, upper = 6.5E-7, x= 0, time= 916000)
```

• Script 54. WinBUGS script for failures in a time period mixed with a lognormal of a given mean with upper and lower bounds.
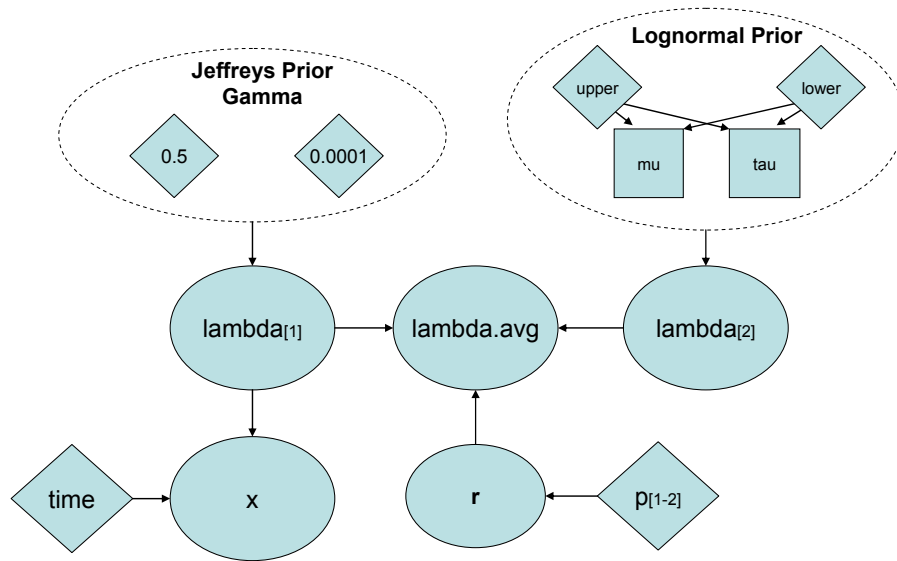
- Figure 4-104. DAG representing Script 54.

The battery has two information sources: one is presented as a failure rate with an assumed lognormal error factor of 10, and the other as a mean failure rate with an upper bound. Script 55 shows the analysis for the batteries, giving a 75% weight to the NASA information over the IEEE standard. Running the script in the usual manner results in a mean value for lambda.avg of 1.2×10^{-5} per hour with a 90% credible interval of 1.0×10^{-8} to 3.0×10^{-5} per hour.

```
Component    : ATCS DC Power System Batteries
Model        : Lognormal
Prior        : Informative, Uncertain
    subscript 1 = Dry cell battery IEEE-Std 500-1984 (recommended = 2.70E-5 per hour,
                  high = 3.10E-5 per hour)
    subscript 2 = Lithium battery failure, NASA non-proprietary data 1.5E-5 per hour.

model  {
lambda[1] ~ dlnorm(mu[1], tau[1])        # Lognormal prior with mean and upper limit
mu[1] <- log(mean[1]) - pow(sigma, 2)/2
tau[1] <- pow(sigma, -2)
sigma <- (2 * 1.645 + sqrt(4*pow(1.645,2) + 8*log(mean[1]/upper)))/2
lambda[2] ~ dlnorm(mu[2], tau[2])        # Lognormal prior with mean and error factor
mu[2] <- log(mean[2]) - pow(log(EF/1.645),2)/2
tau[2] <- pow(log(EF)/1.645,-2)
lambda.avg <- lambda[r]                  # Assignment of weights
r ~ dcat(p[])
}

data
list(p=c(0.25, 0.75), mean=c(2.70E-5, 1.5E-5), upper = 3.10E-5, EF = 10.0)
```

- Script 55. WinBUGS script for failures in a time period and a lognormal (mean and upper bound) given.
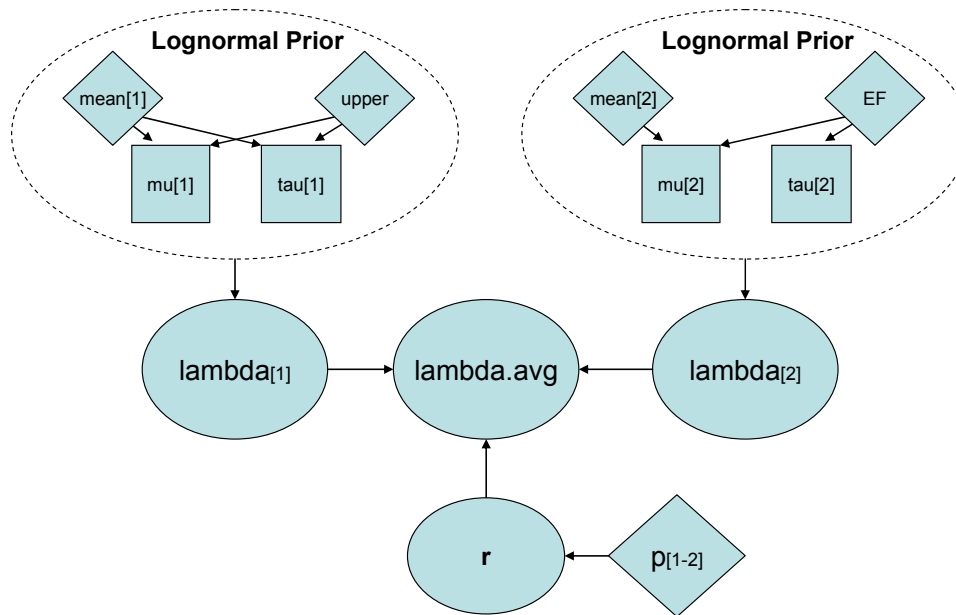
• Figure 4-105. DAG representing Script 55.

4.12.1.2 Control Subsystem—The control subsystem unreliability is directly related to the components of the Avionics system. The avionics package of the ATCS is yet to be determined, but there are years of data based upon the performance of similar avionics systems used under similar conditions to use as a baseline analysis. We will again estimate applicable subsystem performance.

The failure taxonomy for the Control Subsystem of the ATCS is:

Mission Directorate: **Exploration Systems Mission Directorate**
└ Program: **IntraSolar Exploration**
 └ Project: **New Manned Vehicle (NMV)**
  └ Subsystem: **Crew Module**
   └ Subsystem: **Active Thermal Control**
    └ Subsystem: **Control**
     └ Failure Mode: **Fails to Operate**
      └ Failure Mechanism: **Communication Break, Control Failure**
       └ Failure Cause: **Hardware Connection Loss, Software, Hardware Damage**

Referring to Figure 4-99 the avionics control system has a direct impact on operation of the pump and mixing valve. The control system continually monitors and adjusts the performance of the ATCS; therefore it is continually operating. It also has a single point of failure and the failure mode is again "Fails to Operate". This fits the inference flow diagram presented in Figure 4-101.

Once again, since there are no data, the posterior is equal to the prior.

Step 1: The priors

We gather information on the Avionics Control systems relating to the ATCS control. A NASA non-proprietary database provides us with specific data for several subsystems used in prior spacecraft, any of which could cause a failure of the ATCS control:

- Crew Critical Computer with a mean failure rate of 2.36×10^{-5} per hour.
- Wiring failures noted with a mean failure rate of 2.07×10^{-7} per hour.
- Electronic Control Unit with a mean failure rate of 1.26×10^{-5} per hour.

GSFC information provides the following:

- Data Processing Units related to thermal control never failed in 64,163 hours
- Wiring failures occurred twice in 64,163 hours

With no uncertainty provided for the failure rate information, a lognormal distribution with a conservative error factor of ten will be used to represent the uncertainty for each source. Also, the data processing unit can be used for both the crew critical computer and the electronic control unit.

Step 2: The data

Operational data on the Control System is not available. Consequently, the posterior result for the failure-to-operate inference process will be equivalent to the prior information.

Step 3: Bayesian calculation to estimate Control System failure rate

- ☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is Poisson.
- ☑ The prior information is heterogeneous.

All components analyzed for the ATCS Control System have the same information input format: one is in terms of number of failures in an operating time and another is expressed as a failure rate (with an assumed lognormal error factor). Script 56 shows the analysis of the crew critical computer, but the same script can be used for the other two components by substituting their information into the proper lines. The weighting of the information is chosen as equally important although the script lines are there to change the importance if the analyst should see fit to do so. The weights must sum to a value of one.   Table 14 presents the three component's results.

- Table 14. ATCS Control System component analysis values.

Component	mean, per hr	90% Credible Interval, per hr
Crew Critical Computer	1.6×10^{-5}	1.3×10^{-7} to 5.6×10^{-5}
Electronic Control Unit	1.0×10^{-5}	1.2×10^{-7} to 3.7×10^{-5}
Wiring	2.0×10^{-5}	1.3×10^{-8} to 7.2×10^{-5}

```
Component     : ATCS Control System Crew Critical Computer
Model         : Poisson
Prior         : Uninformative
        subscript 1 = Data Processing System GSFC, 0 failures in 64,163 hr
        subscript 2 = Crew Critical Computer in NASA non-proprietary database
              mean= 2.36E-5 per hr, EF = 10
model  {
x ~ dpois(mean.pois)
mean.pois <- lambda[1] * time           # Poisson parameter for prior
lambda[1] ~ dgamma(0.5, 0.0001)         # Jeffreys Priors
lambda[2] ~ dlnorm(mu, tau)     # Lognormal prior with mean and error factor given
mu <- log(mean) - pow(log(EF)/1.645, 2)/2
tau <- pow(log(EF)/1.645, -2)
lambda.avg <- lambda[r]                 # Weighted averaging
r ~ dcat(p[])
}

data
list(x=0, time= 64163, mean= 2.36E-5, EF=10.0, p= c(0.50, 0.50))
```

- Script 56. WinBUGS script for use with failures in time period with uninformed prior and a lognormal prior.
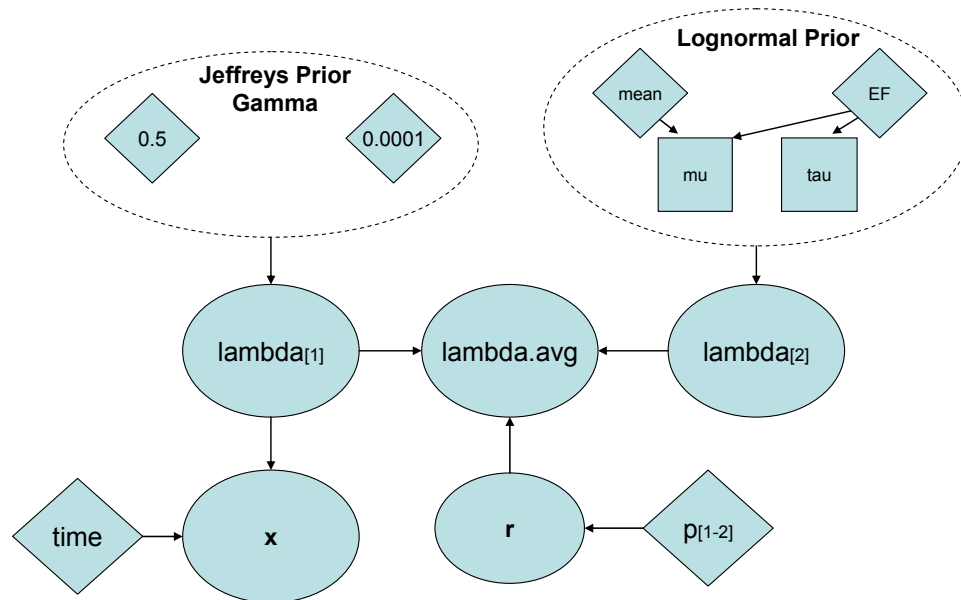
- Figure 4-106. DAG representing Script 56.

4.12.1.3 Radiator—The unreliability of the radiator of the ATCS as a part of the system requires two analyses. One analysis must determine the unreliability of the radiator in an operating state and the other must determine the unreliability of the radiator in a standby mode.

The failure taxonomy for the Radiators of the ATCS is:

Mission Directorate: **Exploration Systems Mission Directorate**
└ Program: **IntraSolar Exploration**
 └ Project: **New Manned Vehicle (NMV)**
  └ Subsystem: **Crew Module**
   └ Subsystem: **Active Thermal Control**
    └ Subsystem: **Radiator**
     └ Failure Mode: **Fails to Operate (Loop A), Fails on Demand (Loop B)**
      └ Failure Mechanism: **Leak or Plug**
       └ Failure Cause: **MMOD, vibration, corrosion, internal debris**

Referring to Figure 4-99 for guidance, we note that there are two loops of parallel radiators within the system. The radiator is assumed to be made of aluminum (as opposed to the carbon models under consideration) and the coolant fluid used is propylene glycol. In Loop A the component type is "operating" and the failure type is "fails to operate." In case this loop fails then the backup Loop B is chosen by the crew. Loop B is kept in stagnant standby; therefore its failure mode is "fails on demand". The "fails on demand" flow chart is shown in Figure 4-107. The "fails to operate" path through the flow chart is identical to that shown in Figure 4-101.
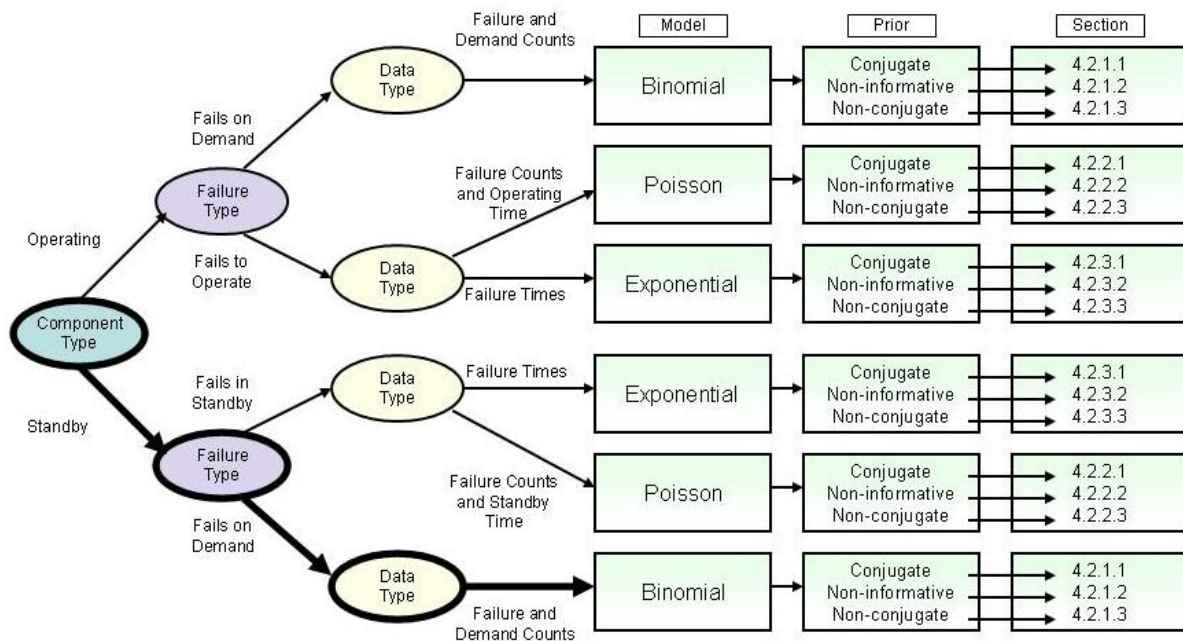
• Figure 4-107. Inference flow diagram path for the Loop B radiator

Step 1: The prior
We gather information related to aluminum radiator systems running propylene glycol and applicable information about radiator failures in space.

Assume data is found that a fleet of glycol-based radiators operated for a combined 438,000 hours over a 2 year period with 8 radiator failures.

For analysis purposes, we will treat the four-radiator system of the space shuttle as one operational radiator. Assume that a leak determined to be caused by Micro Meteoroid Orbital Debris (MMOD) in turn caused one radiator failure in 119 flights. The applicability of the information as compared to other

information is handled through the Bayesian update by assigning weights which must sum to one. Use an average length of a shuttle mission of 11 days (264 hours).

Assume that the Space Shuttle radiator prototype testing resulted in 1,000 total demands with zero failures.

Assume the International Space Station has accumulated 1,500 thermal redundant system checks without failure.

Step 2: The data
No information can be considered operational data for the radiator at this point in time.

Step 3: Bayesian calculation to estimate the individual operational Radiator failure rate
   ☑   The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is Poisson.
   ☑   The prior information is heterogeneous and is only partially applicable.

The Bayesian estimation of the failure rate for the radiator in operation can be performed with Script 57. Running this script in the usual manner produces a value for lambda.avg of 4.2×10^{-5} per hour with a 90% interval of (6.5×10^{-6} to 1.2×10^{-4}) per hour.

```
Component       : ATCS Radiator Loop A (fails during operation)
Model           : Poisson
Prior           : Noninformative
    subscript 1 = Space Shuttle radiator MMOD (1 failure in 119 flights, 11 hr avg flight)
    subscript 2 = Terrestrial aluminum radiators (8 failures in 483,000 hr)

model  {
    for(i in 1:2)  {
        lambda[i] ~ dgamma(0.5, 0.0001)        # Jeffreys priors for lambda
        x[i] ~ dpois(mean.pois[i])             # Poisson likelihood
        mean.pois[i] <- lambda[i] * time[i]    # Poisson mean
    }
lambda.avg <- lambda[r]                        # Weighted average prior
r ~ dcat(p[])                                  # Assignment of weights
}

data
# Weights and observed failures c(Shuttle, Terrestrial)

list(p=c(0.8, 0.2), x=c(1,8), time=c(31416, 438000))
```

• Script 57. WinBUGS script for estimating the failure rate of an operating component (ATCS Radiator) based upon multiple priors.
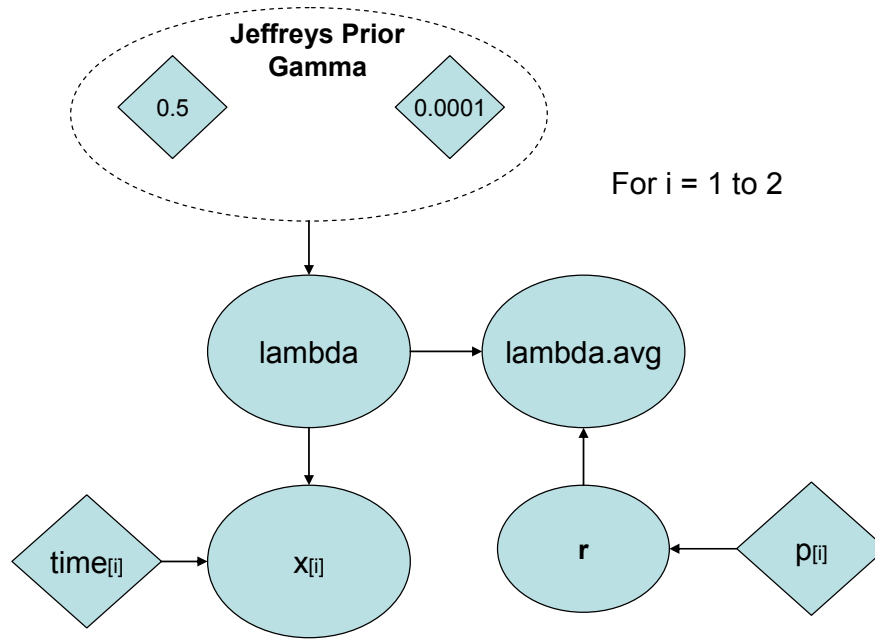
• Figure 4-108. DAG representing Script 57.

Step 4: Bayesian calculation to estimate the on-demand Radiator failure probability

The Bayesian estimation of the failure probability upon demand of the standby ATCS Radiator can be performed with Script 58. The weighting of the Shuttle and ISS information is considered equal, so the weights are assigned as 0.50 each. Running this script in the usual manner produces a mean demand failure probability of failure of 4.1×10^{-4} with a 90% interval of 1.6×10^{-6} to 1.6×10^{-3}. Note the wide range of the interval due to the sparse testing data.

```
Component      : ATCS Radiatior Loop B (fails on demand)
Model          : Binomial
Prior          : Noninformative
    subscript 1: Prototype demand testing (1000 weekly demands 0 failures)
    subscript 2: ISS thermal backup radiator checks (1500 demands, 0 fails)

model  {
    for (i in 1:2) {
        x[i] ~ dbin(p[i],n[i])        # Binomial dist. for number of failures in n demands
        p[i] ~ dbeta(0.5, 0.5)        # Jeffreys Priors, Conjugate beta prior distribution for p
    }
p.avg <- p[r]                         # Posterior average
r ~ dcat(q[])                         # Assignment of weights
}


data
list(x=c(0, 0), n= c(1000, 1500), q=c(0.50, 0.50))   # Priors information and weights
```

• Script 58. WinBUGS script for estimating the probability of failure of a standby component (ATCS Radiator) based on priors.
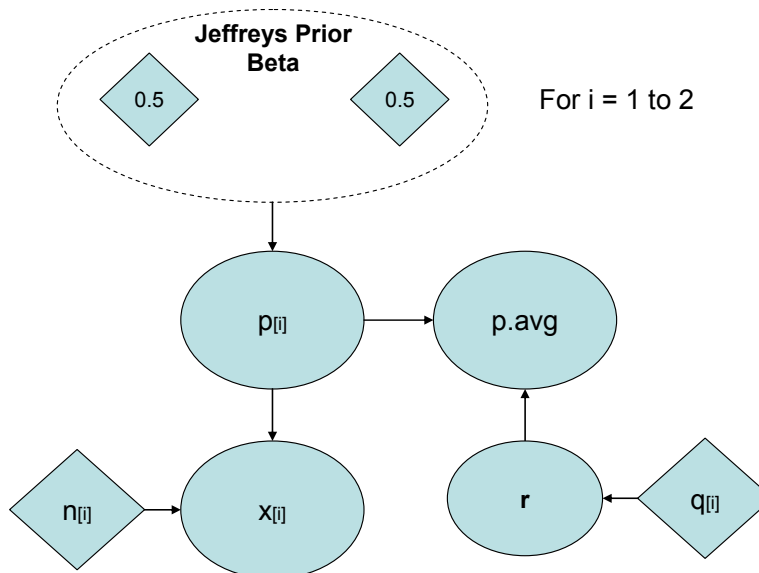
• Figure 4-109. DAG representing Script 58.

4.12.1.4 Control Valve—The unreliability of the Control Valve or three-way mixing valve (TWMV) of the ATCS affects the unreliability of the ATCS.

The failure taxonomy for the Control Valve of the ATCS is:

Mission Directorate: **Exploration Systems Mission Directorate**
└ Program: **IntraSolar Exploration**
 └ Project: **New Manned Vehicle (NMV)**
  └ Subsystem: **Crew Module**
   └ Subsystem: **Active Thermal Control**
    └ Subsystem: **Control Valve**
     └ Failure Mode: **Fails to Operate**
      └ Failure Mechanism: **Sticky Operation or Plug**
       └ Failure Cause: **Vibration, corrosion, internal debris**

Referring to Figure 4-99 for guidance, we note that the control valve is central to the operation of the system and that there is not a redundant component. It is always in operation and receives its power from the DC Power Source and its control signals from the Avionics. One prior source is expressed in terms of mean time between failures (MTBF) and the other sources are expressed as failure counts in hours of operation. MTBF data can be expressed through a normal distribution[1] using the MTBF as the mean and manipulating the confidence level value algebraically to determine the variance. The MTBF can then be translated into a failure rate for use in the Bayesian updating process.

Step 1: The prior
We gather information related to three-way mixing valves from three different sources:
- Assume information is found on three-way mixing valves used in commercial heating and cooling applications in terms of a MTBF of 85,000 hours with a 98% lower confidence level of 79,000 hours.
- Further assume that there have been no three-way mixing valve failures in 1309 hours of Space Shuttle operation including 4 mixing valves (5,236 valve-hours).

---

[1] A normal distribution simplifies the algebra significantly. As long as the epistemic uncertainty in the MTBF is small, as it is here, the normal distribution will not cause problems associated with MTBF < 0.

- From the NUCLARR database, 2 motor-operated globe valves (similar in design to the three-way mixing valve) failed in 1,270,000 hours of operation.

Step 2: The data
Operational data on the Control Valve is not available. Consequently, the posterior result for the failure-to-operate inference process will be equivalent to the prior information.

Step 3: Bayesian calculation to estimate the individual Control Valve failure rate
☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is Poisson.
☑ The prior information is heterogeneous and is known with certainty.

In the case of the ATCS control valve, we have prior information that is not in a homogeneous form. Script 59 constructs a prior that is a weighted average of the three information sources, using a weighting factor of 0.60 for the Space Shuttle data versus 0.20 for both the NUCLARR motor-operated globe valves and the terrestrial heating and cooling valves. The epistemic uncertainty for the terrestrial three-way mixing valve is described by a normal distribution with the stated mean and 98% lower limit. The results for **lambda.avg** are a mean of 6.0×10^{-5} hr^{-1} with 90% interval of (5.3×10^{-7} to 2.9×10^{-4}) hr^{-1}.

```
Component : ATCS Control Valve
Model      : Mixed data (Poisson and MTBF as a normal distribution)
Prior      : Noninformative
          subscript 1 = Space Shuttle use (0 failures in 5,236 hr)
          subscript 2 = NUCLARR motor-operated globe valves (2 failures in 1,270,000 hr)
          subscript 3 = Terrestrial heating and cooling 3-way mixing valves (MTBF=85,000
                    hr, 79,000 hr 98% lower confidence limit)

model  {
    for(i in 1:2)  {
        lambda[i] ~ dgamma(0.5, 0.0001)        # Jeffreys priors
        x[i] ~ dpois(mean.pois[i])
        mean.pois[i] <- lambda[i]*time[i]      # Poisson mean
    }
MTBF ~ dnorm(mu, tau)                          # MTBF to normal data
mu <- 85000                                    # MTBF mean from manufacturer
sigma <- (mu - 79000)/2.05                     # 98% probabiliy level variance
tau <- pow(sigma, -2)
lambda[3] <- 1/MTBF                            # MTBF to lambda
lambda.avg <- lambda[r]                        # Weighted lambda average
r ~ dcat(p[])
}

data
#Weights and observed failures c(Shuttle, NUCLARR, Terrestrial)
list(x=c(0,2), time=c(5236, 1.27E+6), p=c(0.6, 0.2, 0.2))
```

- Script 59. WinBUGS script for estimation of failure rate from Poisson and MTBF mixed priors.
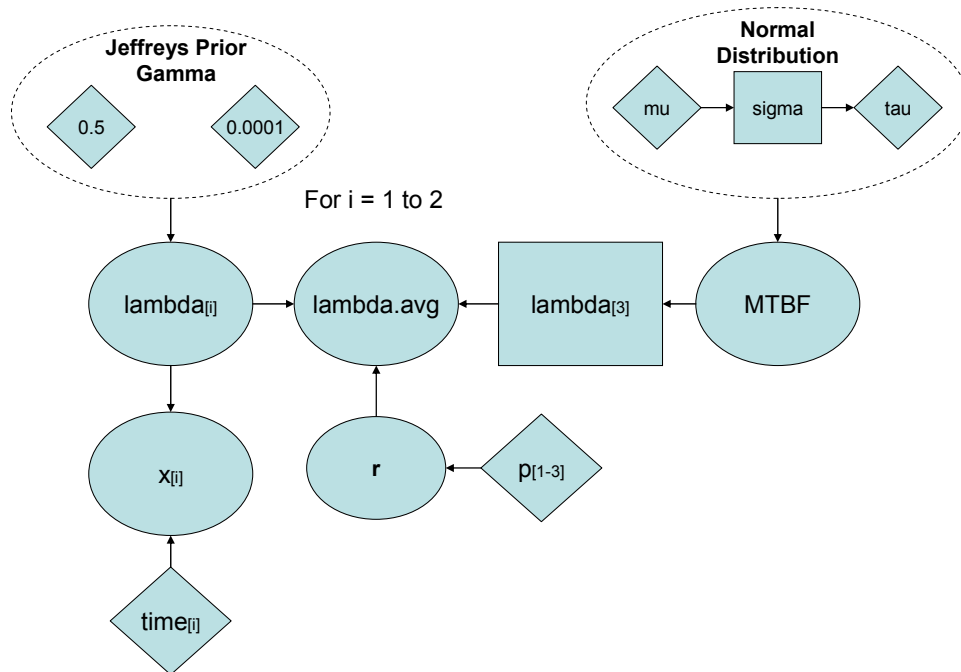
- Figure 4-110. DAG representing Script 59

4.12.1.5 Pump—The unreliability of the Pump is integral to the reliability of the ATCS. The failure taxonomy for the Pump of the ATCS is:

Mission Directorate: **Exploration Systems Mission Directorate**
 └ Program: **IntraSolar Exploration**
   └ Project: **New Manned Vehicle (NMV)**
     └ Subsystem: **Crew Module**
       └ Subsystem: **Active Thermal Control**
         └ Subsystem: **Pump**
           └ Failure Mode: **Fails to Operate**
             └ Failure Mechanism: **Reduced or no flow**
               └ Failure Cause: **Leak, cavitation, impeller damage**

Referring to Figure 4-99 for guidance, we note that the pump is a centrifugal design. There is not a redundant component; it is always in operation, receives its power from the DC Power Source and its control signals from the Avionics. The information we will use for the priors in this case is similar to the control valve, one prior source is expressed in terms of MTBF and the other sources are in terms of failure counts in hours of operation.

Step 1: The prior
We gather information related to electric centrifugal pumps of the size used in the ATCS.
Assume the manufacturer's data for the proposed pump lists a Mean Time Between Failure (MTBF) of 50,000 hours with a 95% lower confidence level of 42,000 hours.

Further assume that there have been no ATCS pump failures in 31416 hours of Space Shuttle operation (4 pumps per mission = 125,664 pump-hours).

Step 2: The data

Operational data on the pump is not available. Consequently, the posterior result will be equivalent to the prior information.

Step 3: Bayesian calculation to estimate the individual Pump failure rate
☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is Poisson.
☑ The prior information is not homogeneous and is uncertain.
☑ The observed data are homogeneous and are known with certainty.

The shuttle data is considered more relevant than the manufacturer's estimate, so Script 60 constructs a weighted-average prior, again assuming a normal distribution around the MTBF estimate from the manufacturer. Running this script in the usual manner gives a posterior mean for **lambda.avg** of 1.4×10^{-5} per hour, with a 90% interval of 1.6×10^{-6} to 2.9×10^{-5} per hour.

```
Component      : ATCS Pump
Model          : Poisson / MTBF data (normal distribution assumption)
Prior          : Noninformative
        subscript 1 = Space Shuttle data (1 failures in 125,664 total hr)
        subscript 2 = Manufacturer's estimate (50,000 hr MTBF mean, 42,000 hr MTBF,
                95% CL)

model  {
x.sh ~ dpois(mean.sh)                   # Poisson likelihood of Space Shuttle data
mean.sh <- lambda[1] * time.sh
lambda[1] ~ dgamma(0.5, 0.0001)         # Jeffreys priors for lambda
MTBF ~ dnorm(mu,tau)                    # Manufacturer's estimate
mu <- 50000                             # MTBF mean
sigma <- (mu - 42000)/1.64              # 95% lower probability value
tau <- pow(sigma, -2)
lambda[2] <- 1/MTBF                     # MTBF to lambda
lambda.avg <- lambda[r]                 # Weighted average prior
r ~ dcat(p[])
}

data
list(p=c(0.80, 0.20), x.sh=1, time.sh=125664)
```

• Script 60. WinBUGS script for ATCS Pump using multiple data with times to failure and MTBF.
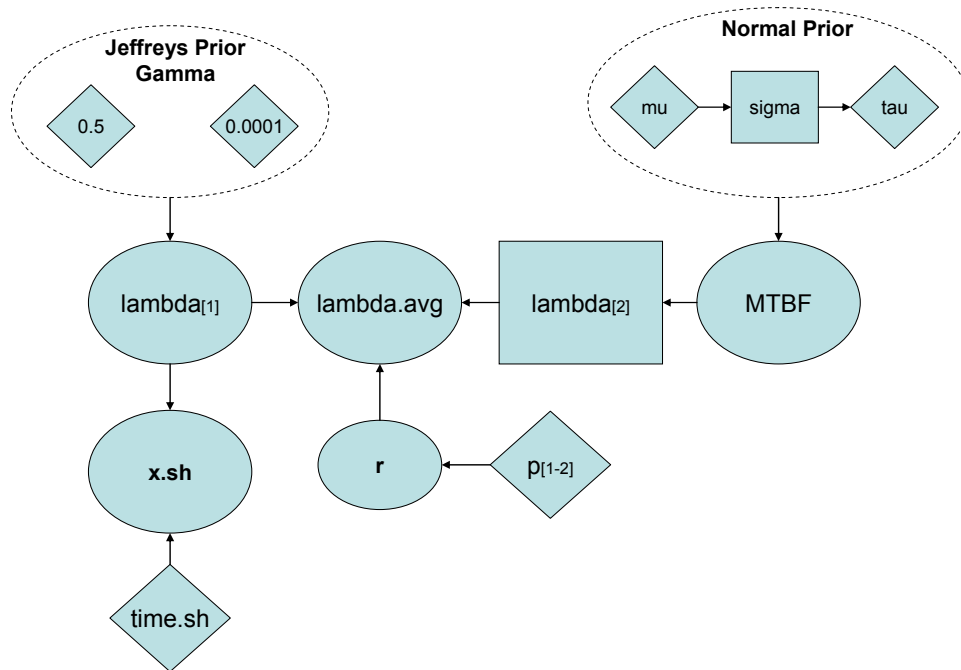
• Figure 4-111. DAG representing Script 60.

4.12.1.6 Refrigerant Tank—The unreliability of the refrigerant tank of the ATCS is integral to the unreliability of the system. The refrigerant tank is a pressure vessel that must meet ASME standards and is thus expected to be very reliable. The failure taxonomy for the Refrigerant Tank of the ATCS is:

Mission Directorate: **Exploration Systems Mission Directorate**
  └ Program: **IntraSolar Exploration**
    └ Project: **New Manned Vehicle (NMV)**
      └ Subsystem: **Crew Module**
        └ Subsystem: **Active Thermal Control**
          └ Subsystem: **Refrigerant Tank**
            └ Failure Mode: **Fails to Operate Maintain Pressure**
              └ Failure Mechanism: **Leak**
                └ Failure Cause: **MMOD, cracked weld, poor fitting connection**

Step 1: The prior
Information related to refrigerant tanks and pressure vessels similar to those used in the ATCS:
Savannah River Generic Database records for pressurized tanks indicate a 1.0×10^{-8} hour mean with a lognormal error factor of 10.

Tanks that are pressurized have the following data in NUCLARR:

Mean	Median	EF	Failures	Cumulative Hrs.
	9.3E-10	8.0	0	2.52E+8
1.6E-8		2.0	4	2.52E+8
2.6E-9		3.3	1	3.80E+8
1.1E-8		2.0	4	3.80E+8

Step 2: The data

In the Refrigerant Tank case, there is operational information from space applications that could be considered applicable enough to be treated as data in a Bayesian update, or this information could be factored into a weighted-average prior, along with the prior information listed above. Both types of analysis are illustrated to illustrate the similarity of results.

Operational records indicate 1,540,000 hours of operation without a refrigerant tank failure.

A manufacturer of a compressor-based cooling system lists 49 machine years (429,240 hours) with 0 failures in a life test and 38 machine years (332,880 hours) with 1 failure in operational orbit. The failure was not attributable to the refrigerant tank.

Step 3: Bayesian calculation to estimate the individual Refrigerant Tank failure rate
- ☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is Poisson.
- ☑ The prior information is heterogeneous and uncertain.
- ☑ The observed data are homogeneous and are known with certainty.

Something to consider when constructing the analysis is whether to pool information. The manufacturer's data includes both operational and life testing hours of operation all with zero failures. In general, applicable zero failure data over long time periods can be pooled and the total time of operation of 762,120 hours with zero failures is used.

The importance of the operational data is considered greater than the manufacturer's data. Also, the Savannah River Generic Database information might be considered of greater importance than the NUCLARR information. When considering the weighting of the satellite system information versus the terrestrial information, the satellite information carries a much greater importance. The actual weights assigned are up to the analyst. In this case, it was decided to assign a lower weight to the terrestrial information. Of course, depending on the placement of the component and other factors, a decision could be made in a real-life scenario to omit terrestrial information as completely not applicable to low ambient pressure operation of pressurized tanks.

The model can be developed in one of two ways. One is to consider the GSFC and the manufacturer's information as data and perform a Bayesian update using them as such, taking the posterior result and using it in a weighted average comparison to the lognormal priors. This is illustrated in Script 61. The alternative model is to consider all the information as priors and average the resulting lambdas as is illustrated in Script 62. Running these two models in the usual manner gives nearly identical results:

Script 61: lambda.avg = 3.3×10^{-7} per hour with a 90% interval of 7.6×10^{-10} to 1.4×10^{-6} per hour

Script 62: lambda.avg = 3.2×10^{-7} per hour with a 90% interval of 7.4×10^{-10} to 1.4×10^{-6} per hour

```
Component:     : ATCS Refrigerant Tank
Model          : Poisson
Prior          : Informative Mixture Prior
        subscript 1 = Satellite records (1.54E+6 hr, 0 refrigerant tank failures)
        subscript 2 = Satellite Cooling System manufacturer (762,120 hr, 0 failures)
        subscript 3 = Savannah River Generic Database (1.0E-8 per hr mean, EF = 10.0)
        subscript 4 = NUCLARR tank data (9.3E-10 median, 8.0 EF, 1.6E-8 mean, 2.0 EF,
                      2.6E-9 mean, 3.3 EF, 1.1E-8 mean, 2.0 EF)


model  {
    for (i in 1:n) {
        x[i] ~ dpois(mean.pois[i])       # Poisson distribution for number of events
        mean.pois[i] <- lambda.p[i] * time[i]     # Poisson parameter
        lambda.p[i] ~ dgamma(0.5, 0.0001)         # Jeffreys prior distribution for lambda
    }
lambda[6] <- lambda.p[r]                  # Weighted Poisson lambda posterior average
r ~ dcat(p[])

# Lognormal prior with mean and error factor given
for (j in 1:4)       {
        mu[j] <- log(prior.mean[j]) - pow(log(prior.mean.EF[j])/1.645,2)/2
        tau[j] <- log(prior.mean.EF[j])/1.645, -2)
    }

# Lognormal prior with median and error factor given
mu[5] <- log(prior.median)
tau[5] <- pow(log(prior.med.EF)/1.645, -2)

for (k in 1:5)       {
        lambda[k] ~ dlnorm(mu[k], tau[k])
    }
lambda.avg <- lambda[s]
s ~ dcat(q[])

# Define Weights
p[1] <- (0.70/0.85)                 # Weight for operational records
p[2] <- (1-(0.70/0.85))             # Weight for Sat Cooling System
q[1] <- 0.15 * 0.60                 # Weight for Savannah River Database info
for (l in 2:5)       {              # Weights for NUCLARR information
    q[l] <- (0.15 * 0.40)/4
    }
q[6] <- 0.85                        # Weight for GSFC and SCS combined
}

data
list(x=c(0,0,0), time=c(1.54E+6, 762120), n=2, prior.median=9.3E-10,
    prior.med.EF=8.0, prior.mean=c(1.0E-8, 1.6E-8, 2.6E-9, 1.1E-8),
    prior.mean.EF=c(10.0, 2.0, 3.3, 2.0))
```

• Script 61. WinBUGS script for ATCS Refrigerant Tank using posterior of data sources mixed with lognormal priors for posterior lambda.out.

```
Component      : ATCS Refrigerant Tank
Model          : Poisson
Prior          : Informative Mixture Prior
    subscript 1 = Satellite records (1.54E+6 hr, 0 refrigerant tank failures)
    subscript 2 = Satellite Cooling System manufacturer (762,120 hr, 0 failures)
    subscript 3 = Savannah River Generic Database (1.0E-8 per hr mean, EF = 10.0)
    subscript 4 = NUCLARR tank data (9.3E-10 median, 8.0 EF, 1.6E-8 mean, 2.0 EF,
                  2.6E-9 mean, 3.3 EF, 1.1E-8 mean, 2.0 EF)

model  {
# Update Jeffreys prior with past data from GSFC and satellite cooling system records
    for(i in 1:2)  {
        x[i] ~ dpois(mean.pois[i])
        mean.pois[i] <- lambda[i]*time[i]
        lambda[i] ~ dgamma(0.5, 0.0001) #Jeffreys prior
    }
# Incorporate SRL and NUCLARR information
    for(j in 3:7)  {
        lambda[j] ~ dlnorm(mu[j], tau[j])
    }
# SRL information
mu[3] <- log(SRL) - pow(log(EF.SRL)/1.645, 2)/2
tau[3] <- pow(log(EF.SRL)/1.645, -2)

# NUCLARR information
mu[4] <- log(NUCLARR[1])
tau[4] <- pow(log(EF.NUCLARR[1])/1.645, -2)
mu[5] <- log(NUCLARR[2]) - pow(log(EF.NUCLARR[2])/1.645, 2)/2
tau[5] <- pow(log(EF.NUCLARR[2])/1.645, -2)
mu[6] <- log(NUCLARR[3]) - pow(log(EF.NUCLARR[3])/1.645, 2)/2
tau[6] <- pow(log(EF.NUCLARR[3])/1.645, -2)
mu[7] <- log(NUCLARR[4]) - pow(log(EF.NUCLARR[4])/1.645, 2)/2
tau[7] <- pow(log(EF.NUCLARR[4])/1.645, -2)

#Construct overall mixture prior for lambda
lambda.avg <- lambda[r]
r ~ dcat(p[])
# Define weights
p[1] <- 0.7                      # Weighting for operational records
p[2] <- 0.15                     # Weighting for satellite data
p[3] <- 0.15*0.6                 # Weighting for SRL information
    for(k in 4:7) {
        p[k] <- 0.15*0.40/4      # Weighting for each piece of NUCLARR information
    }
}

data
list(x=c(0, 0), time=c(1.54E+6, 762120), SRL=1.0E-8, EF.SRL=10, NUCLARR=c(9.3E-10,
1.6E-8, 2.6E-9, 1.1E-8), EF.NUCLARR=c(8, 2, 3.3, 2))
```

• Script 62. WinBUGS script for ATCS Refrigerant Tank using all information as priors.
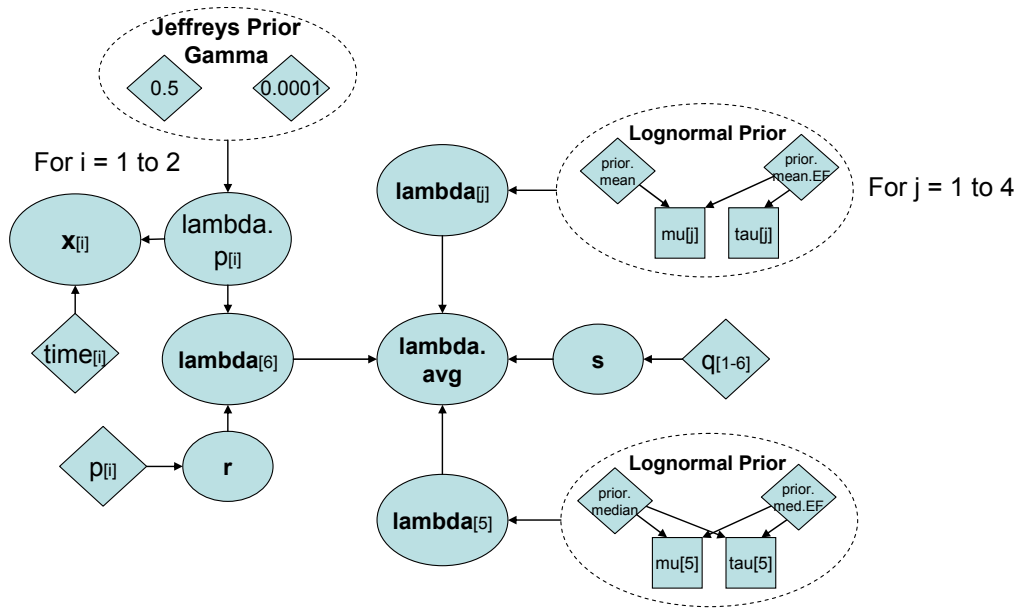
• Figure 4-112. DAG representing Script 61.
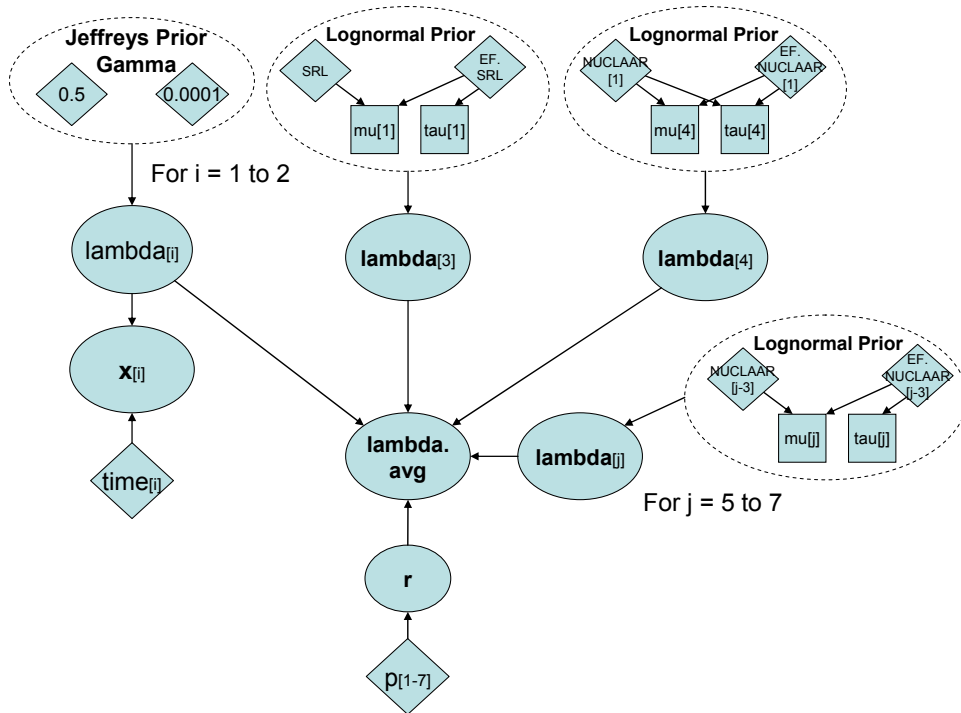
• Figure 4-113. DAG representing Script 62.

4.12.2 Case Study Implementation (Final Design and Modification) Phase

After the initial design acceptance and prior information has been evaluated, the final design takes shape and performance characteristics are analyzed. The Bayesian updating analysis can cover anything from load capacity of cabling to prototype testing of complete subsystems.

4.12.2.1 Radiator—The ATCS radiator has reached a final design and there are design differences from the ones used in the priors to consider and prototype testing data to include. Data from an additional four Space Shuttle missions are also available.

Step 1: The priors

Of the operational radiator priors, the terrestrial prior is identical to the Approval Phase analysis but the Space Shuttle prior is modified for two reasons. One is simply that there have been more flights between the Approval Phase and the Implementation Phase. The second modification is a result of design and operational differences between the Space Shuttle and the Constellation. The design differences have decreased the probability of MMOD damage to the Constellation ATCS radiator as compared to the Space Shuttle. The probability of MMOD penetration is calculated to be 77% that of the shuttle due to less surface area and modified protection. The value of 0.77 is a measure of the model applicability and is used prior to the Bayesian updating by multiplying the number of failures noted in the Space Shuttle design by this factor. As before, the applicability of the data as compared to other data is handled through the Bayesian update by assigning weights which must sum to one.

- Our current hypothetical data is one leak on a shuttle radiator in 123 flights (average of 11-day mission equates to 32,472 hours).
- Assume data is found that a fleet of 50 radiators operated for a combined 438,000 hours over a 2 year period with 8 radiator failures.

For the on-demand priors, we have the original Space Shuttle information and an update of the ISS information:
- Assume that the Space Shuttle radiator prototype demand produced demand testing of 1000 total demands with zero failures.
- Assume the International Space Station has accumulated 1,547 thermal redundant system checks without failure.

Step 2: The data

Prototype testing of the final design of the radiator has produced:
- Zero failures in 11,000 total component hours of operation
- Zero failures in 500 demands

Step 3: Bayesian calculation to estimate the individual operational Radiator failure rate

- ☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is Poisson.
- ☑ The prior information is heterogeneous and is only partially applicable.
- ☑ The data is known with certainty.
  The Bayesian estimation of the failure rate for the radiator in operation can be performed with Script 63. Running this script in the usual manner produces a mean value for lambda.avg of 2.7 $\times 10^{-5}$, with a 90% interval of (3.1×10^{-6} to 7.5×10^{-5}).

```
Component     : ATCS Radiator Loop A (fails during operation)
Model         : Poisson
Prior         : Noninformative
    subscript 1 = Space Shuttle radiator MMOD (1 failure in 123 flights, 264 hr avg flight)
    subscript 2 = Terrestrial aluminum radiators (8 failures in 483,000 hr)
    subscript 3 = Prototype Constellation radiator testing (1 failures in 11,000 hr)

model  {
    for(i in 1:2)  {
        lambda[i] ~ dgamma(0.5, 0.0001)        # Jeffreys priors for lambda
        x[i] ~ dpois(mean.pois[i])             # Poisson likelihood
        mean.pois[i] <- lambda[i] * time[i]    # Poisson mean
    }
lambda.avg <- lambda[r]                        # Weighted average prior
r ~ dcat(p[])                                  # Assignment of weights
x[3] ~ dpois(mean.pois.c)
mean.pois.c <- lambda.avg * time[3]            # Poisson mean
}

data
# Weights and observed failures c(Shuttle, Terrestrial, proto)
list(p=c(0.8, 0.2),          x=c(0.77, 8, 0), time=c(32472, 438000, 11000))
```

• Script 63. WinBUGS script to update the operational Radiator failure rate based on updated priors information and prototype testing.
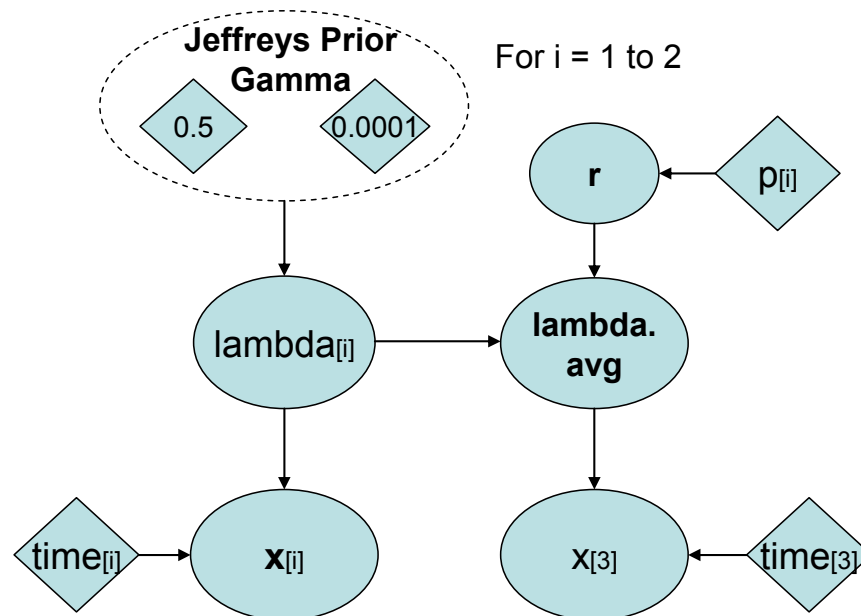
• Figure 4-114. DAG representing Script 63, Script 65, and Script 67.

Step 4: Bayesian calculation to estimate the on-demand Radiator failure probability
- ☑ The aleatory model of the world (akin to the likelihood function in Bayes' Theorem) is Binomial.
- ☑ The prior information is heterogeneous and is only partially applicable.
- ☑ The data is known with certainty.

The updated Bayesian estimation of the failure probability upon demand of the standby ATCS Radiator can be performed using Script 64. Running the script in the usual manner produces a mean probability of failure of 2.9×10^{-4}, with a 90% interval of 1.2×10^{-6} to 1.1×10^{-3}.

```
Component     : ATCS Radiatior Loop B (fails on demand)
Model         : Binomial
Prior         : Noninformative
    subscript 1: Prototype demand testing (1000 weekly demands 0 failures)
    subscript 2: ISS thermal backup loop radiator checks (1547 demands, 0 fails)
    subscript 3: Prototype testing (500 demands 0 failures)

model  {
    for (i in 1:2) {
        x[i] ~ dbin(p[i],n[i])        # Binomial dist. for number of failures in n demands
        p[i] ~ dbeta(0.5, 0.5)        # Jeffreys Priors, Conjugate beta prior distribution for p
    }
p.avg <- p[r]
r ~ dcat(q[])
x[3] ~ dbin(p.avg,n[3])
}

data
list(x=c(0, 0, 0), n= c(1000, 1547, 500), q=c(0.50, 0.50))     #Priors information and weights
```

• Script 64. WinBUGS script to update the standby Radiator probability of failure upon demand in the Implementation Phase incorporating new priors evidence and prototype testing.
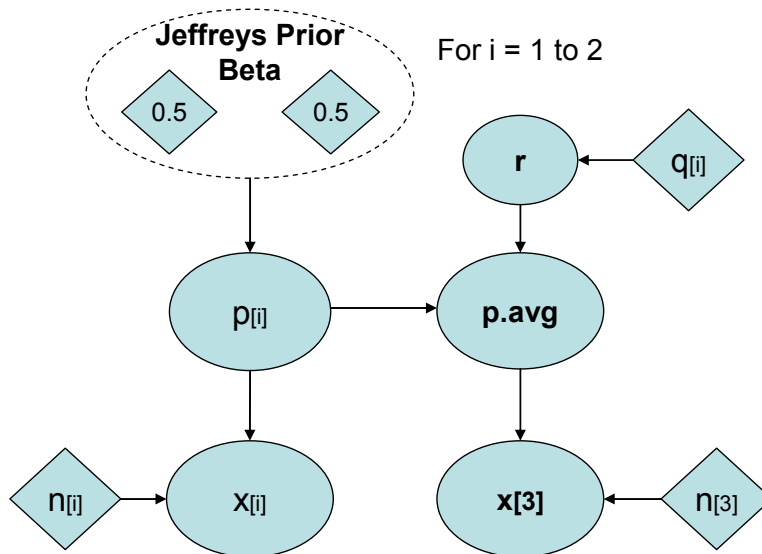
• Figure 4-115. DAG representing Script 64, Script 66, and Script 68.

4.12.3 Case Study Integration Phase
The final design is integrated into the system and final operational testing is performed. An update of the component reliability can be performed with this new information.

4.12.3.1 Radiator—The ATCS radiator is now integrated into the system. Original prototype testing of the unit alone continues and now there is a life test of the complete system in progress. The new data along with updates from the priors all are used to perform an updated estimate.

Step 1: The priors
There have been five more Space Shuttle flights and the ISS has continued to operate and periodically test its backup cooling loops. The design differences that decreased the probability of MMOD damage to the new ATCS radiator as compared to the Space Shuttle have remained the same, keeping the Space Shuttle information factor at 0.77. As before, the applicability of the data as compared to other data is handled through the Bayesian update by assigning weights which must sum to one.
- Our current hypothetical information is that MMOD caused a leak on one shuttle radiator in 128 flights (average of 11-day mission equates to 33,792 hours).
- Assume data is found that a fleet of 50 radiators operated for a combined 438,000 hours over a 2 year period with 8 radiator failures.

For the on-demand priors, we have the original Space Shuttle information and an update of the ISS information:
- Assume that the Space Shuttle radiator prototype demand produced demand testing of 1000 total demands with zero failures.
- Assume the International Space Station has accumulated 1,600 thermal redundant system checks without failure.

Step 2: The data
Prototype testing of the final design of the radiator has produced:
- Zero failures in 24,000 total component hours of operation
- Zero failures in 1,100 demands

ATCS System life testing has produced:
- Zero failures in 3,000 accumulated system hours
- Zero failures in 330 demands

Step 3: Bayesian calculation to estimate the individual operational Radiator failure rate
- ☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is Poisson.
- ☑ The prior information is heterogeneous and is only partially applicable.
- ☑ The data is known with certainty.

The Bayesian estimation of the failure rate for the radiator in operation during the Integration Phase can be performed with Script 65. As was noted in the analysis of the refrigerant tank during the Approval Phase, equally applicable zero-failure data can be pooled. This allows us to pool the ATCS System life testing data with the radiator prototype testing data. Running this script in the usual manner produces a mean value for lambda.avg of 2.0×10^{-5} per hour with a 90% interval of (2.3×10^{-6} to 5.3×10^{-5}).

```
Component      : ATCS Radiator Loop A (fails during operation)
Model          : Poisson
Prior          : Noninformative
    subscript 1 = Space Shuttle radiator MMOD (1 failure in 128 flights, 264 hr avg flight)
    subscript 2 = Terrestrial aluminum radiators (8 failures in 483,000 hr)
    subscript 3 = Prototype radiator testing and ATCS System life testing pooled data
                  (0 failures in 27,000 hr)


model  {
    for(i in 1:2)  {
        lambda[i] ~ dgamma(0.5, 0.0001)      #Jeffreys priors for lambda
        x[i] ~ dpois(mean.pois[i])           #Poisson likelihood
        mean.pois[i] <- lambda[i] * time[i]  #Poisson mean
    }
lambda.avg <- lambda[r]                      #Weighted average prior
r ~ dcat(p[])                                #Assignment of weights
x[3] ~ dpois(mean.pois.c)
mean.pois.c <- lambda.avg * time[3]          #Poisson mean
}

data
#Weights and observed failures c(Shuttle, Terrestrial, proto)
list(p=c(0.8, 0.2),        x=c(0.77, 8, 0), time=c(33792, 438000, 27000))
```

• Script 65. WinBUGS script to update the operational Radiator failure rate based on updated priors
information and pooled prototype and system testing.

Figure 4-114 shows the DAG representing Script 65.

Step 4: Bayesian calculation to estimate the on-demand Radiator failure probability
☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is
  Binomial.
☑ The prior information is heterogeneous and is only partially applicable.
☑ The data is known with certainty.

The updated Bayesian estimation of the failure probability upon demand of the standby ATCS Radiator
can be performed using Script 66. Running the script in the usual manner produces a mean demand
failure probability of 1.8×10^{-4} with a 90% interval of 7.3×10^{-7} to 7.1×10^{-4}.

```
Component      : ATCS Radiatior Loop B (fails on demand)
Model          : Binomial
Prior          : Noninformative
     subscript 1: Prototype demand testing (1000 weekly demands 0 failures)
     subscript 2: ISS thermal backup loop radiator checks (1600 demands, 0 fails)
     subscript 3: Pooled radiator prototype and system life testing
                  (1430 demands 0 failures)


model  {
    for (i in 1:2) {
         x[i] ~ dbin(p[i],n[i])      #Binomial dist. for number of failures in n demands
         p[i] ~ dbeta(0.5, 0.5)      #Jeffreys Priors, Conjugate beta prior distribution for p
    }
p.avg <- p[r]
r ~ dcat(q[])
x[3] ~ dbin(p.avg,n[3])             #Posterior distribution
}

data
list(x=c(0, 0, 0), n= c(1000, 1600, 1430), q=c(0.50, 0.50))   #Priors information and weights
```

• Script 66. WinBUGS script to update the standby Radiator probability of failure upon demand in the Integration Phase incorporating new priors evidence and pooled prototype and ATCS system life testing.

Figure 4-115 shows the DAG representing Script 66.

4.12.4 Case Study Operation Phase

The components are now operational and have flown as part of the ATCS system. The analyst has decisions to make as the components and system mature. At what point should the prototype data and system life test data move into the priors? Once these data are moved to priors, what weight should be assigned to them? Now that there is more known about the system, are the initial design priors from the approval stage still applicable enough to include, even with a heavy discounting in the analysis?

4.12.4.1 Radiator—The radiator is now operational and has flown four missions as part of the ATCS system. The prototype testing and ATCS life testing programs are complete. The Space Shuttle program has ended after 10 more flights and experienced one more MMOD caused failure on an ATCS radiator. The new information needs to be incorporated into the current failure probability estimates.

Step 1: The priors

There have been ten more Space Shuttle flights and the ISS has continued to operate and periodically test its backup cooling loops. The design differences that decreased the probability of MMOD damage to the new ATCS radiator as compared to the Space Shuttle have remained the same, keeping the Space Shuttle information factor at 0.77. As before, the applicability of the data as compared to other data is handled through the Bayesian update by assigning weights which must sum to one.

Operational priors include:
   • Our current hypothetical information is that operation and MMOD caused a leak on one shuttle radiator in 138 flights (average of 11 day mission equates to 36,432 hours).
   • Assume data is found that a fleet of 50 radiators operated for a combined 438,000 hours over a 2 year period with 8 radiator failures.
   • Prototype testing of the final design of the radiator concluded with 30,000 component hours of operation without failure.
   • ATCS System life testing concluded with 15,000 accumulated system hours without failure.

On-demand priors include:
- Assume that the Space Shuttle radiator prototype demand produced 1000 total demands with zero failures.
- Assume the International Space Station has accumulated 2,100 thermal redundant system checks without failure.
- Radiator prototype testing concluded at 2,000 demands without failure.
- ATCS System life testing concluded at 1,000 demands on the standby radiator without failure.

Step 2: The data
The new ATCS has flown four missions averaging 15 days (1,440 component operational hours). The standby loop has been tested 60 times. No operational or standby failures have been noted.

Step 3: Bayesian calculation to estimate the individual operational Radiator failure rate
☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is Poisson.
☑ The prior information is heterogeneous and of varying applicability.
☑ The data is known with certainty.

The Bayesian estimation of the failure rate for the radiator in operation during the Integration Phase can be performed with Script 67. When considering the weighting factors on the priors, the Space Shuttle operational information was still given a high weight (0.48) due to its ability to include the risk of MMOD strikes on the system. The terrestrial data was discounted heavily (0.02) and the majority of the weight went to the prototype and ATCS life test pooled data (0.50). Running this script in the usual manner produces a mean value for lambda.avg of 2.2×10^{-5} per hour with a 90% interval of (1.7×10^{-7} to 7.4×10^{-5}) per hour. This slight increase in posterior mean and interval over the Integration Phase is expected due to the relatively sparse operational data used to develop the posterior.

```
Component     : ATCS Radiator Loop A (fails during operation)
Model         : Poisson
Prior         : Noninformative
    subscript 1 = Space Shuttle radiator MMOD (1 failure in 138 flights, 264 hr avg flight)
    subscript 2 = Terrestrial aluminum radiators (8 failures in 483,000 hr)
    subscript 3 = Prototype radiator testing and ATCS System life testing pooled data
                  (0 failures in 45,000 hr)
    subscript 4 = ATCS operational phase time in service (0 failures in 1440 hr)


model  {
    for(i in 1:3)  {
        lambda[i] ~ dgamma(0.5, 0.0001)        # Jeffreys priors for lambda
        x[i] ~ dpois(mean.pois[i])             # Poisson likelihood
        mean.pois[i] <- lambda[i] * time[i]    # Poisson mean
    }
lambda.avg <- lambda[r]                        # Weighted average prior
r ~ dcat(p[])                                  # Assignment of weights
x[4] ~ dpois(mean.pois.c)
mean.pois.c <- lambda.avg * time[4]            # Poisson mean
}
data
#Weights and observed failures c(Shuttle, Terrestrial, proto)
list(p=c(0.48, 0.02, 0.50),    x=c(0.77, 8, 0, 0), time=c(36432, 438000, 45000, 1440))
```

• Script 67. WinBUGS script for the operational phase of the ATCS radiator moving prototype and ATCS life testing data to the priors and using operational data to update posterior.

Figure 4-114 shows the DAG representing Script 67.

Step 4: Bayesian calculation to estimate the on-demand Radiator failure probability
- ☑ The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) is Binomial.
- ☑ The prior information is heterogeneous and is only partially applicable.
- ☑ The data is known with certainty.

The updated Bayesian estimation of the failure probability upon demand of the standby ATCS Radiator can be performed using Script 68. Running the script in the usual manner produces a mean probability of failure of 2.0×10^{-4} with a 90% interval of 7.1×10^{-7} to 7.8×10^{-4}. Again, the probability of failure has increased slightly despite more demands with failure free operation because of the sparse operational data and the movement of the testing data to the priors for operational phase updating.

```
Component      : ATCS Radiatior Loop B (fails on demand)
Model          : Binomial
Prior          : Noninformative
    subscript 1: Shuttle Prototype demand testing (1000 weekly demands 0 failures)
    subscript 2: ISS thermal backup loop radiator checks (2100 demands, 0 fails)
    subscript 3: Pooled radiator prototype & system life tests (3011 demands 0 failures)
    subscript 4: ATCS operational backup loop radiator checks (60 demands, 0 failures)

model   {
    for (i in 1:3) {
        x[i] ~ dbin(p[i],n[i])        #Binomial dist. for number of failures in n demands
        p[i] ~ dbeta(0.5, 0.5)        #Jeffreys Prior for p
    }
p.avg <- p[r]
r ~ dcat(q[])
x[4] ~ dbin(p.avg,n[4])               #Posterior distribution
}

data
#Priors information and weights
list(x=c(0, 0, 0, 0), n= c(1000, 2100, 3011, 60), q=c(0.10, 0.10, 0.80))
```

- Script 68. WinBUGS script for Operational Phase update of standby ATCS radiator with testing data moved to priors and operational data used to update posterior.

Figure 4-115 shows the DAG representing Script 68.

4.13 Extensions, Issues, and Pitfalls

This section provides coverage of extensions to methods already addressed, general analysis issues, and potential pitfalls.

4.13.1 MCMC Sampling Initial Values and Convergence

In the past, MCMC-based methods have been associated with long run times both for convergence and follow on iterations for estimates. Modern software and processors make the MCMC technology viable. For the majority of the inference models presented in this guidebook, the MCMC used in OpenBUGS runs 100,000 iterations in less than 30 seconds on a desktop processor (circa 2005). Even the most complex models take around two minutes to run 100,000 iterations. It is not out of the question to run a million or more iterations in a reasonable amount of time.

In OpenBUGS, the initial sampling values are loaded manually through the script or generated automatically by the program and give the MCMC engine a place to start. Regardless of where the start is, the destination is the same, convergence to the inferred answer. This guidebook cautions to check for convergence and presented the tools used to check for convergence. The standard run of generating the initial values through the program, running 1,000 iterations to allow for convergence and 100,000 iterations for sampling is acceptable for most applications. Some models however, require further checks for convergence. Many of the scripts throughout the guide set the initial values within the script. Many of these scripts also run two or more chains of initial values to check for convergence. The use of initial values is twofold: one is that the program can not always generate a set of initial values that will allow it to start the MCMC; the other is to provide data to confirm convergence.

To illustrate, recall Script 21. The results running the script with the given initial values are a = -3.4, b = 0.16, p.value = 0.47, and p[10] = 0.15. If the script is run by compiling just one chain and clicking on generate initial values ("gen inits"), the software will error because the flat distributions on parameters a and b require a starting point. Given initial values that are reasonable but different, the MCMC will converge to the same results.

What are reasonable initial values for Script 21? Looking at the logit(x) function used, the OpenBUGS help under scalar functions shows the logit function as $\ln\left(\dfrac{x}{1-x}\right)$, so logit(p(i)) in the script is

$\ln\left(\dfrac{p(i)}{1-p(i)}\right)$ and p(i) should be between a value of 0 and 1. This binds the logit(p(i)) value between about -6.9 to 6.9. Therefore, values of "a" and "b" entered as initial values such that $-6.9 < a + bi < 6.9$ would be considered reasonable starting points.

What about using unreasonable starting values? The MCMC will attempt to start at wherever you tell it to, and many times it can be successful even if started outside of a reasonable range. To illustrate this property, load initial values for Script 21 as chain 1: a = 97, b = 0, and chain 2: a = -1, b = 5. The convergence graphs below show that the model will take longer to converge and that the values of the chain are originally far from the end values. However, the overall results are identical to that found in the original Script 21.
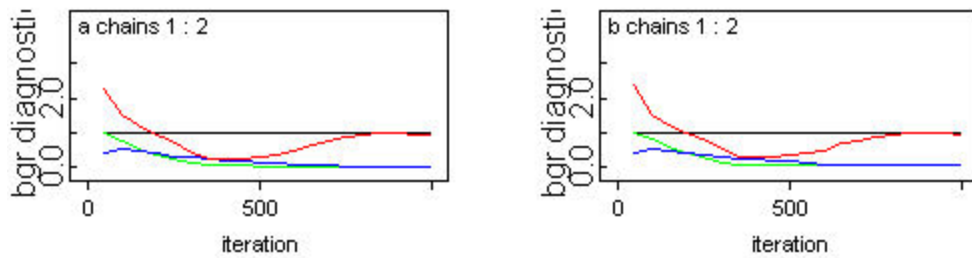
● Figure 4-116. BGR Diagnostics results for Script 21 with initial values a = (97, -1) and b = (0, 5).
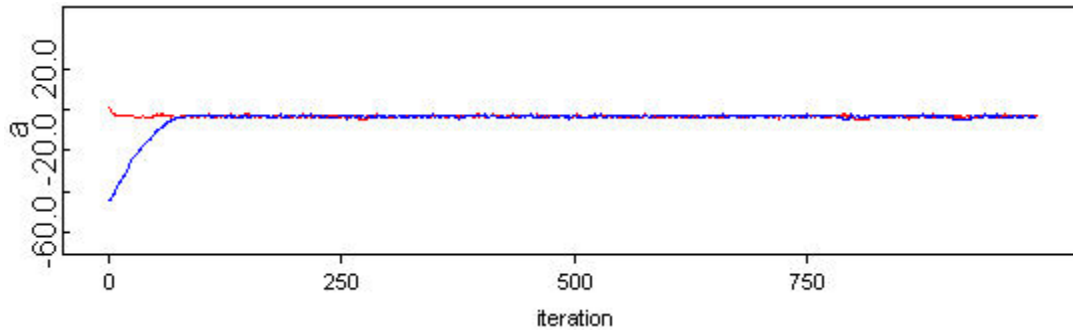
● Figure 4-117. History results for Script 21 parameter "a" with initial values a = (97, -1) and b = (0, 5).
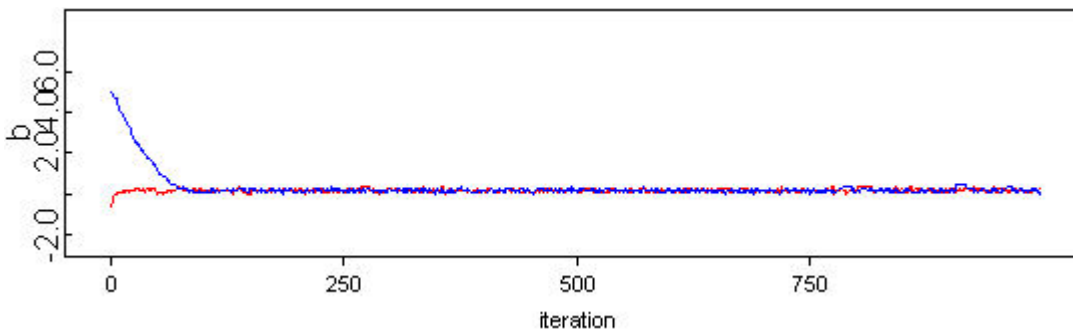
● Figure 4-118. History results for Script 21 parameter "b" with initial values a = (97, -1) and b = (0, 5).

Although the MCMC is forgiving in that it will take a large range of starting values and eventually converge to a consistent answer, it is best practice to choose the initial values within a reasonable range based upon the equations that are used. It is also good practice to set the different strings far apart from each other within the reasonable range.

There is, of course, a limit to how far away from the reasonable values the initial values can be set. For instance, in Script 21, the program will terminate and not provide a final answer if one tries to run the script with a value for "a" greater than 97.5.

4.13.2 Extensions of Existing Methods
Assume a component similar in application and environment to prior components will be used on a proposed platform and an estimation of the probability of failure upon demand is required. The analysis can be split into three steps.

1. Gather all the information available on the component and its similar predecessors.
2. Select an analysis model and evaluate the applicability of the predecessor components information.

3. Create an inference diagram to help choose an applicable OpenBUGS script (or solve the problem via numerical integration if possible) to handle the inference.

In the first step, the analyst considers the operation of the component and determines that it is used to perform a mechanical operation on demand after resting in standby. There are processed data from telemetry of two missions where a similar component was used with demands and successes. The first mission produced 570 demands with no failures. The component operated in a degraded state on the second mission after 475 demands, where it would stick and require a re-issued command to get it to operate. Fifteen random instances occurred in this range where the command was re-issued and the component operated. A decision to not include the 15 commands is made based on analysis of the telemetry which was inconclusive as to signal reception by the component. Not counting the re-issued commands as a demand, a hard failure occurred at 677 demands where it would not operate any more. The two components which flew on the two separate missions were of the same manufacturer, although they are not of the same manufacturer as the proposed component. The proposed component comes with a reliability claim from its manufacturer in terms of mean cycles to failure and a confidence claim. There has been no prototype testing to date other than that performed by the manufacturer. Analysis of the information by the team determines that the degraded operation was an annoyance and only the hard fail at 677 demands would be counted. The information available by the numbers:

Mission 1, component A:	570 demands, 0 failures
Mission 2, component A:	677 demands, 1 failure
Component B Manufacturer's test data:	1500 demands, 1350 demands at 95% lower confidence level

The analyst determines that the best model for operation of the component is the binomial based on the operation profile of the component and the demand and success data. The two prior missions are considered to be equivalent in application and environment. At a design meeting, the new component is touted as less expensive, yet meets the criteria of the old component and the applicability of the prior mission data is determined to be 90%. The provided testing documentation from the new component's manufacturer shows due diligence and is accepted as a 100% valid estimation.

An inference diagram is drawn based on the data provided. These diagrams have to be drawn with the thought in mind that Bayesian inference is from a distribution to a distribution, which is why one can't just place a number such as the lower bound estimate of the test data into the inference, rather the whole normal distribution about the data developed with the 95% confidence lower limit is used. Since there are no parameters known for the prior operation of the component to this information, a Jeffreys prior is used for the binomial model used for the two existing missions. The manufacture's information can be modeled as a normal distribution about the estimate in the same manner the MTBF was in Sections 4.12.1.4 and 4.12.1.5. The outputs of the two results would need the posteriors averaged, so the diagram will need the p.avg node and the r node used for a categorical distribution. The diagram ends up as shown in Figure 4-119.
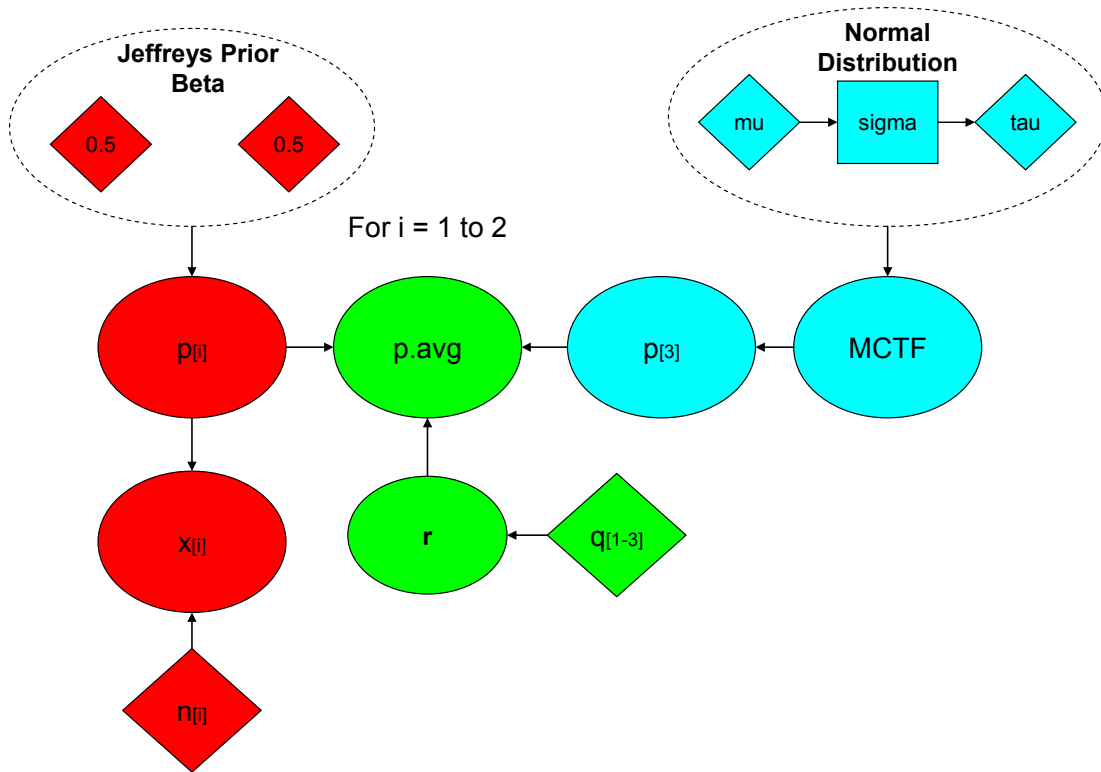
- Figure 4-119. Inference diagram (DAG) for a component with applicable flight data and manufacturer's estimate.

The model requires construction next. There are several pieces of the model identifiable in the DAG, which are color coded in the figure. The binomial piece is in red, the manufacturer information piece is in blue, and the posterior averaging is in green. WinBUGS scripts always begin with the "model {" keyword, and following that it doesn't matter what order the lines are in, the BUGS language runs the inference the same. Following the convention used through the majority of the case study, place the binomial piece in the script first. A good example of the piece required can be found in Script 55 which also has two binomial information sources. So far the model looks like:

```
model  {
    for (i in 1:2) {
        x[i] ~ dbin(p[i],n[i])        # Binomial dist. for number of failures in n demands
        p[i] ~ dbeta(0.5, 0.5)        # Jeffreys Priors, Conjugate beta prior distribution for p
    }
```

Next, the manufacturer information needs to be developed as a distribution. A normal distribution about an estimate avoids unnecessarily-complex arithmetic if the bound is tight enough to prevent the manufacturer information to be less than one. This is the case here with the manufacturer information being a normal distribution with mean of 1500 demands and a 95% lower limit of 1350 demands.

Utilizing the properties of the normal distribution: mu = mean = 1500, $tau = \dfrac{1}{sigma^2}$, sigma being

the standard deviation which can be calculated by $sigma = \dfrac{mu - LCValue}{Z}$, Z being found through

calculator or table as 1.645 for 95% lower confidence limit. Coded in the BUGS language it looks like:

```
mu <-  1500
tau <- pow(sigma, -2)
sigma <- (mu – 1350)/1.645
p ~ dnorm(mu, tau)
```

The next part of the model is the posterior averaging portion. Many scripts throughout Chapter 4 utilize this and it is written as:

```
p.avg <- p[r]
r ~ dcat(q[])
```

What remains is to determine the weights and add the data. The weights have to correspond to the order of the data. They can be listed within the script or in the data. Script 34 is an example of listing the formulas or distributions used to determine the weighting factor in the script. Nearly all of the case study scripts use the developed factors in the data. Here, we will place the formulas in the script. Putting the pieces together, the script is as follows:

```
model  {
#Flight data 1 and 2
    for (i in 1:2) {
        x[i] ~ dbin(p[i],n[i])        # Binomial dist. for number of failures in n demands
        p[i] ~ dbeta(0.5, 0.5)        # Jeffreys Priors, Conjugate beta prior distribution for p
    }

# Manufacturer's estimation
mu <-  1500                           # Mean estimation
tau <- pow(sigma, -2)
sigma <- (mu - 1350)/1.645            # Std. deviation using the 95% lower confidence value
MCTF ~ dnorm(mu, tau)                 # Resulting normal distribution about the mean
p[3] <- 1/MCTF                        # Convert to a probability
p.avg <- p[r]
r ~ dcat(q[])
qsum <- 2.8
q[1]  <- 0.9 / 2.8
q[2] <- 0.9 / 2.8
q[3] <- 1 / 2.8
}
data
list(x=c(0, 1), n=c(570, 677))
```

● Script 69. WinBUGS script for a component with applicable flight data and manufacturer's estimate.

The results for p.avg after 1000 burn-in iterations followed by 100000 iterations for estimation are a mean of 1.2×10^{-3} with a 90% credible interval of (3.4×10^{-5}, 4.2×10^{-3}).

Examining the densities of the results gives some insight into the Bayesian process. Figure 4-120 shows that p[1] and p[2] have a shape consistent with a Beta distribution, p.avg is an average mixture of the three, but one might question why there is a slight skew to the left in the p[3] density of the

normal distribution used for the manufacturer information. Bayesian inference flows in both forward and reverse directions throughout a network. At iteration number 1, the manufacturer information estimation is the normal distribution that was placed in the model. As the iterations increase, influence is felt both on the posterior by the normal distribution and also on the normal distribution by the developing posterior and the other distributions in the network. Very little direct data is available for the performance of the component, as is the case for many problems faced in estimating the performance of a relative unknown, but the forward and backflow of inference in Bayesian analysis refines the estimations and builds upon what is known to further hone in on the posterior value.
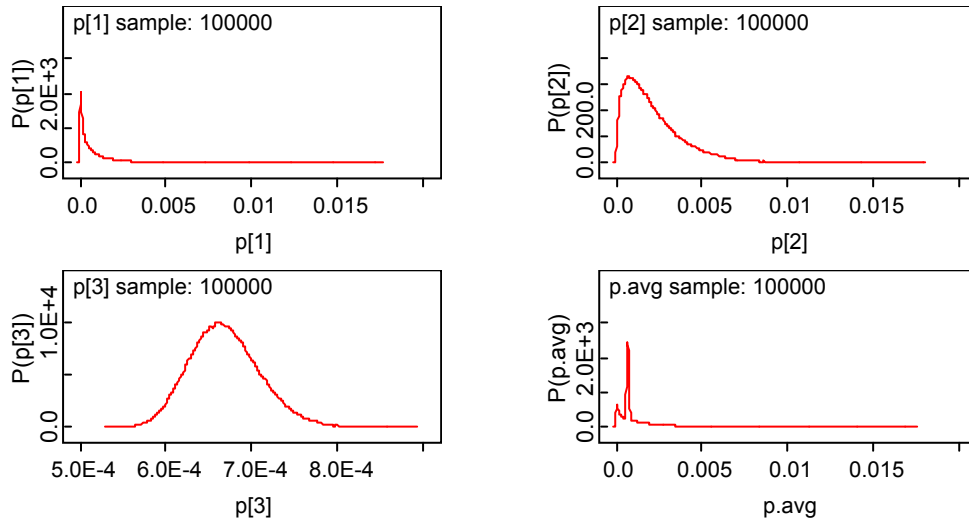
• Figure 4-120. Parameter densities for the example.

Next, let's take the same problem developed thus far and say that further analysis of the telemetry from the first flight indicates that the component was receiving the command signal during the "soft" failures starting at 475 demands. Looking back in the guide, Script 31 shows a way to handle uncertainty in binomial demands. Analyzing the telemetry shows that the soft failure started at 475 demands, and a total of 15 soft failures occurred with a corresponding command re-issued before the end failure at 677 demands. Since the soft failures are being considered now, the command re-issues should also be added into the mix. The result is that the component operated for 690 demands with a failure between 475 and 692. In the first script, the two flights were combined because the binomial model was the same for both sets of data. This situation will not allow the combination of the two sets of flight data in the script because of the uncertainty in the second set of information requires a uniform distribution on the number of demands as per Script 31. Drawing this as a DAG in Figure 4-121, it splits into 4 parts, one for the binomial model with a known number of demands (red), one for the binomial model with an uncertain number of demands (yellow), one for the manufacturer information (blue) and the posterior averaging part (green).
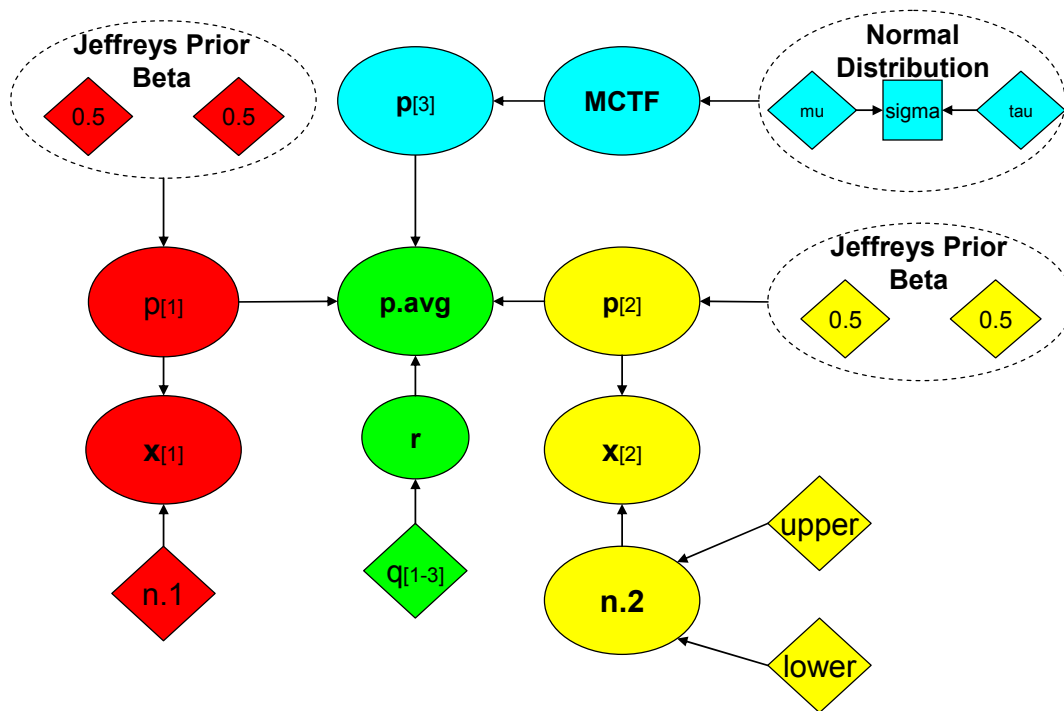
- Figure 4-121. DAG for a component with uncertain flight data and manufacturer's estimate.

The new script handling this uncertainty can be built by separating the three pieces of information. This script has a uniform distribution for the number of demands on one of those flights, so it can not use the convenient "for i in x" loop. Data points where the parameters are the same, such as the number of failures, x, can still use the list annotation in the data **c( )**. The resulting script is as follows:

```
model   {
# Flight 1 information
x[1] ~ dbin(p[1],n.1)                # Binomial dist. for number of failures in n demands
p[1] ~ dbeta(0.5, 0.5)              # Jeffreys Priors, Conjugate beta prior distribution for p

# Flight 2 information
x[2] ~ dbin(p[2], n.2)             # Binomial dist. for number of failures in n demands
n.2 ~ dunif(lower, upper)          # Uniform distribution for number of demands
p[2] ~ dbeta(0.5, 0.5)             # Conjugate beta prior distribution for p

# Manufacturer's estimation
mu <-  1500                        # Mean estimation
tau <- pow(sigma, -2)
sigma <- (mu - 1350)/1.645         # Std. deviation using the 95% lower confidence value
MCTF ~ dnorm(mu, tau)              # Resulting normal distribution about the mean
p[3] <- 1/MCTF                     # Convert to a probability
p.avg <- p[r]
r ~ dcat(q[])
qsum <- 2.8
q[1]  <- 0.9 / 2.8
q[2] <- 0.9 / 2.8
q[3] <- 1 / 2.8
}

data
list(x=c(0, 1), n.1= 570, lower = 450, upper = 692)
```

• Script 70. WinBUGS script for a component with uncertain flight data and manufacturer's estimate.

The results for p.avg after 1000 burn-in iterations followed by 100,000 iterations for estimation are a mean of 1.4 x 10^{-3} with a 90% credible interval of (3.4 x 10^{-5}, 5.0 x 10^{-3}).

4.13.3 Ad hoc Methods versus Bayesian Inference

For a group of failure records, one common ad hoc method is to simply pool the data together. For example, each source might be used to generate a mean and variance as follows.

If the number of failures (x) is greater than zero, then each source produces a beta distribution with parameters α = x and β = n –x, where n is the number of demands (the standard conjugate updating approach). From the properties of a beta distribution, the mean is then x/n and the variance is approximately $x(n – x)/n^3$. If no failures were recorded for a particular data source, then α is taken to be 0.5 (assuming a Jeffreys prior is used).

To get the overall posterior distribution for the set of information, a beta distribution may be fit to the overall mean and variance using:

$$\mu_{tot} = \frac{1}{n} \sum_{i=1}^{n} \mu_i$$

$$\sigma_{tot}^2 = \frac{1}{n} \sum_{i=1}^{n} \sigma_i^2 + \frac{1}{n-1} \sum_{i=1}^{n} \left(\mu_i - \mu_{tot} \right)^2$$

For the hypothetical data in Table 15, the overall mean and variance can be found to be 9.6E-4 and 2.8E-6, respectively. From these moments, the parameters of the resulting beta distribution can be found using:

$$\alpha_{tot} \approx \frac{\mu_{tot}^2}{\sigma_{tot}^2}$$

$$\beta_{tot} = \frac{\alpha_{tot} \left(1 - \mu_{tot} \right)}{\mu_{tot}}$$

A lognormal distribution could also be fit using these overall moments. For the data in Table 15, the overall beta distribution has parameters 0.3 and 343.6. The fitted lognormal distribution has mean 9.6E-4 and error factor (EF) of 7.

- Table 15. Hypothetical failure data for fan check valves.

Record Number	Failure Mode	Failures	Demands
469	FTO	0	11,112
470	FTO	0	3,493
471	FTO	0	10,273
472	FTO	1	971
473	FTO	0	4,230
474	FTO	0	704
475	FTO	0	7,855
476	FTO	0	504
477	FTO	0	891
478	FTO	0	846
480	FTO	0	572
481	FTO	0	631
482	FTO	0	2,245
488	FTO	0	7,665
532	FTO	0	1,425
534	FTO	0	700
538	FTO	0	716
549	FTO	8	1,236
550	FTO	0	926
551	FTO	1	588
552	FTO	0	856
554	FTO	1	708
569	FTO	0	724
570	FTO	12	8,716
592	FTO	2	632
593	FTO	0	564

The overall posterior developed by this ad hoc method using the average-moment approach depends on the fitted distribution and summarized below for two different distributions.

- Table 16. Summary of overall fan check valve prior, average-moment approach.

Fitted Distribution	Mean	5th Percentile	95th Percentile
Beta	9.6E-4	9.4E-8	4.0E-3
Lognormal	9.6E-4	6.8E-5	3.3E-3

4.13.3.1 The Hierarchical Bayes Approach—Because of the large source-to-source variability exhibited by Table 15, it may be inappropriate to pool the data. The standard Bayesian approach to such a problem is to specify a hierarchical prior for the demand failure probability, p. We will compare the results from this approach with the *ad hoc* average-moment approach. We will analyze two different first-stage priors, beta and logistic-normal, with independent diffuse hyperpriors in both cases. WinBUGS Script 71 was used to carry out the analysis.

```
model  {
    for(i in 1:N) {
        x[i] ~ dbin(p[i], n[i])        # Binomial model for number of events in each source
        p[i] ~ dbeta(alpha, beta) # First-stage beta prior
        #p[i] <- exp(p.norm[i])/(1 + exp(p.norm[i])) # Logistic-normal first-stage prior
        #p.norm[i] ~ dnorm(mu, tau)
        x.rep[i] ~ dbin(p[i], n[i])  # Replicate value from posterior predictive distribution
        #Generate inputs for Bayesian p-value calculation
        diff.obs[i] <- pow(x[i] - n[i]*p[i], 2)/(n[i]*p[i]*(1-p[i]))
        diff.rep[i] <- pow(x.rep[i] - n[i]*p[i], 2)/(n[i]*p[i]*(1-p[i]))
    }
p.avg ~ dbeta(alpha, beta)          #Average beta population variability curve
#p.avg ~ dlnorm(mu, tau)
#p.norm.avg ~ dnorm(mu, tau)
#p.avg <- exp(p.norm.avg)/(1 + exp(p.norm.avg))
#Compare observed failure total with replicated total
x.tot.obs <- sum(x[])
x.tot.rep <- sum(x.rep[])
percentile <- step(x.tot.obs - x.tot.rep)    #Looking for values near 0.5
# Calculate Bayesian p-value
chisq.obs <- sum(diff.obs[])
chisq.rep <- sum(diff.rep[])
p.value <- step(chisq.rep - chisq.obs)       #Mean of this node should be near 0.5
# Hyperpriors for beta first-stage prior
alpha ~ dgamma(0.0001, 0.0001)
beta ~ dgamma(0.0001, 0.0001)
#mu ~ dflat()
#tau <- pow(sigma, -2)
#sigma ~ dunif(0, 20)
}

inits
list(alpha=1, beta=100) #Chain 1
list(alpha=0.1, beta=200) #Chain 2
list(mu=-11, sigma=1)
list(mu=-12, sigma=5)
```

Script 71 continued on next page.

```
data
x[]  n[]
0    11112
0    3493
0    10273
1    971
0    4230
0    704
0    7855
0    504
0    891
0    846
0    572
0    631
0    2245
0    7665
0    1425
0    700
0    716
8    1236
0    926
1    588
0    856
1    708
0    724
12   8716
2    632
0    564
END

list(N=26)
```

• Script 71. Hierarchical Bayesian script used for the comparison to ad hoc methods.

Using a beta first-stage prior
The overall average distribution representing source-to-source variability in p has a mean of 1.2E-3, variance of 1.4E-4, and a 90% credible interval of (3.5E-20, 4.1E-3). The very small 5th percentile is an artifact of choosing a beta distribution as a first-stage prior. The posterior mean of α is 0.12, and the average variability distribution has a sharp vertical asymptote at p = 0.

Using a logistic-normal first-stage prior
The logistic-normal distribution is constrained to lie between 0 and 1, and because the density function goes to 0 at both 0 and 1, it avoids the vertical asymptote at p = 0 from which the beta distribution suffers. For small values of p, the logistic-normal and lognormal distributions are very close; we chose to use the logistic-normal distribution because, with such large variability, the Monte Carlo sampling in WinBUGS can generate values of p > 1, and these have the potential to skew the results, particularly the mean.

With a logistic-normal first-stage prior, we found the overall average distribution representing source-to-source variability to have a mean of 0.01, variance of 8.7E-3, and 90% credible interval of (7.6E-11, 1.2E-2).

Update with New Data
Assume we would like to update the overall check valve prior with new data, which we take to be 0 failures in 2,000 demands. The results of the four different update possibilities are shown in Table 17. As a reference point, we include an update of a Jeffreys prior, which is a beta(0.5, 0.5) distribution.

• Table 17. Posterior results for the ad hoc versus Bayesian method comparison.

Method	Mean	5th Percentile	95th Percentile
Ad hoc (beta)	1.3E-4	1.4E-8	5.9E-4
Ad hoc (lognormal)	3.0E-4	4.3E-5	8.3E-4
Hierarchical Bayes (beta)	4.9E-5	5.9E-23	2.9E-4
Hierarchical Bayes (logistic-normal)	4.7E-5	2.6E-11	2.5E-4
Jeffreys prior	2.5E-4	9.8E-7	9.6E-4

As shown in Figure 4-122, a perhaps surprising outcome is that updating the lognormal distribution fit by *ad hoc* methods using the average-moment approach gives about the same mean and 95th percentile for p as simply updating the Jeffreys prior (however the 5th percentile differs between the two posteriors). The beta prior would give about the same result if there were not a vertical asymptote at p = 0, causing excess shrinkage of the mean toward 0.

Both hierarchical Bayes analyses give similar means and 95th percentiles; the 5th percentiles differ because of the vertical asymptote in the beta first-stage prior. Hierarchical Bayes allows the large number of sources with zero failures to more strongly influence the result than the average-moment ad hoc approach. With no failures in 2,000 demands, the posterior mean is pulled more towards a value of zero in the hierarchical Bayes analysis, giving a less conservative result.
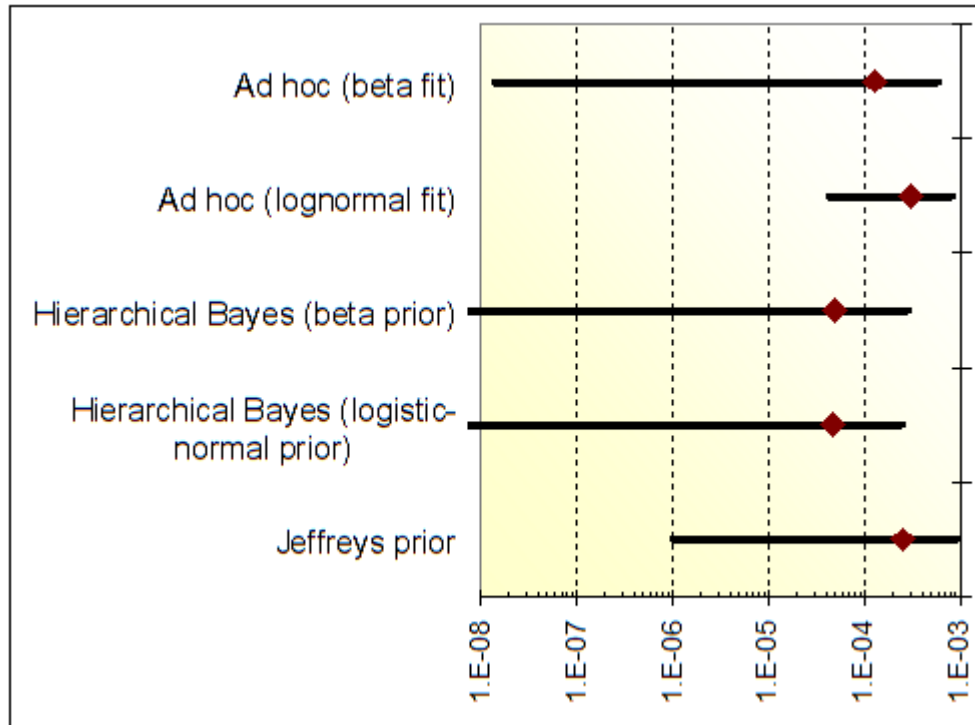
• Figure 4-122. Plot of the posterior results for the Bayesian versus ad hoc method comparison.

Model Validation

We can generate replicate failure counts for the data sources in Table 15, and then use the Bayesian p-value calculated from the chi-square summary statistic to compare models. Table 18 shows the results of this model validation calculation. The *ad hoc* distributions derived from the average-moment approach are poor at replicating the observed data: they over-predict the total number of failures (the observed total was 25) and they under-predict the variability in the failure count, leading to a very low Bayesian p-value. In contrast, the hierarchical Bayes models have much better predictive validity.

• Table 18. Model validation results for the ad hoc versus Bayesian method comparison.

Method	Total Replicated Failures (mean)	Bayesian p-value
Ad hoc (beta)	61.0	0.001
Ad hoc (lognormal)	67.3	0.001
Hierarchical Bayes (beta)	25.1	0.44
Hierarchical Bayes (logistic-normal)	25.0	0.38

Conclusions

The averaging approach used by the *ad hoc* method has the potential to give erroneous results, either overly conservative, as in the example here, or nonconservative. Without model validation, one cannot judge the validity of *ad hoc* approximations. The hierarchical Bayes approach is the preferred method for incorporating source-to-source variability. With tools like WinBUGS, model validation can be performed in tandem with parameter estimation. In the example here, hierarchical Bayes was superior at replicating the observed data, as measured by the total replicated failure count and Bayesian p-value.

References

Andrews, J. D. and T. R. Moss, 2002, *Reliability and Risk Assessment*, the American Society of Mechanical Engineers, New York.

Apostolakis, G. E., 1994, "A Commentary on Model Uncertainty," Proceedings of Workshop I in Advanced Topics in Risk and Reliability Analysis, Model Uncertainty: Its Characterization and Quantification, A. Mosleh, N. Siu, C. Smidts, and C. Lui, Eds., NUREG/CP-0138, U.S. Nuclear Regulatory Commission, Washington, D.C.

American Society of Mechanical Engineers, 1977, *Failure Data and Failure Analysis in Power and Processing Industries*, "Failure Data and Risk Analysis," W. E. Vesely, pp. 61- 75, The Energy Technology Conference.

Bayes, T., 1763, "An essay towards solving a problem in the doctrine of chances." *Philosophical Transactions of the Royal Society*, **53**: 370 – 418.

Birolini, A., 2004, *Reliability Engineering Theory and Practice*, 4th Edition, Springer.

D'Agostini, G., 1995, *Probability and Measurement Uncertainty in Physics - a Bayesian Primer,* DESY-95-242, http://xxx.lanl.gov/PS_cache/hep-ph/pdf/9512/9512295v2.pdf.

Dalal, S. R., E. B. Fowlkes, and B. Hoadley, 1989, "Risk Analysis of the Space Shuttle: Pre-Challenger Prediction of Failure," *Journal of the American Statistical Association*, **84**, No. 408 (December), pp. 945-957.

Dodge, R., 1924, "Problems of Human Variability," *Science*, Vol. LIX, No. 1525.

Eerola, M., 1994, Probabilistic Causality in Longitudinal Studies, Springer_Verlag, New York.

Guikema, S., 2005, "A comparison of reliability estimation methods for binary systems," *Reliability Engineering and System Safety*, **87**, Issue 3, pp. 365-376.

Henley, E. J. and H. Kumamoto, 1985, *Designing for Reliability and Safety Control*, Prentice-Hall.

Hill, A. B., 1965, "The Environment and Disease: Association or Causation?," *Proceedings for the Royal Society of Medicine*, **58**, 295-300.

Jaynes, E., 2003, *Probability Theory – The Logic of Science*, Cambridge University Press.

Kelly, D. L., 1998, "Bayesian Hypothesis Testing and the Maintenance Rule," 4th International Conference in Probabilistic Safety Assessment and Management (PSAM 4), Springer.

Laplace, P. S., 1814, *A Philosophical Essay on Probabilities*, Dover Publications (reprint 1996).

MIL-STD-1629A, 1984, Procedures for Performing a Failure Mode, Effects and Criticality Analysis, Change Note 2.

Mosleh, A., 1991 "Common Cause Failures: An Analysis Methodology and Examples," *Reliability Engineering and System Safety*, **34**, pp. 249-292.

NASA, 2002, NASA PRA Procedures Guide, v1.1

NASA, 2006, NASA Engineering and Safety Center Working Group Report – Taxonomy Working Group Final Report, RP-06-11, Version 2.0.

NASA, 2007, *NASA Systems Engineering Handbook*, NASA/SP-2007-6105, Rev. 1.

Pearl, J., 2000, *Causality -- Models, Reasoning, and Inference*, Cambridge University Press.

Rome Air Development Center, 1995, *Nonelectronic Parts Reliability Data*, NPRD-95.

Smith, C.L., V.N. Shah, T. Kao, and G. Apostolakis, 2000, Incorporating Aging Effects into Probabilistic Risk Assessment - A Feasibility Study Utilizing Reliability Physics Modeling, NUREG/CR-5632, November.

Sträter, O., 2004, "Considerations on the elements of quantifying human reliability," *Reliability Engineering and System Safety*, **85**, pp. 255-264.

Tumer, I., Stone, R., and Bell, D., 2003, "Requirements for a Failure Mode Taxonomy for Use in Conceptual Design," *Proceedings of the International Conference on Engineering Design, ICED 03*, **Paper 1612, Stockholm, Sweden.**

Winkler, R. L., 1972, *An Introduction to Bayesian Inference and Decision*, **Holt, Rinehart, and Winston, New York.**

Index

A. Definitions

Aleatory Pertaining to stochastic (non-deterministic) events, the outcome of which is described by a probability. From the Latin *alea* (game of chance, die).

Context The set of environmental conditions and circumstances, both helpful and adverse, that surround a reliability or risk-relevant event. In the analysis of human performance, context includes both system-manifested impacts (e.g., available time, ergonomics, and task complexity) and human-manifested impacts (e.g., stress level, degree of training, fatigue). In the analysis of hardware or software performance, context includes the operational environments and interactions with other hardware, software, and human elements.

Bayesian Inference A process of inference using Bayes' Theorem in which knowledge is used to newly infer the plausibility of a hypothesis. This process (Bayesian inference) produces information that adds to organizational knowledge.

Component A group of parts designed to work together to perform a specific function. Components are constituents of systems and subsystems.

Credible Interval Bayesian inference produces a probability distribution. The "credible interval" consists of the values at a set (one low, one high) of specified percentiles from the resultant distribution. For example, a 90% credible interval ranges from the value of the 5th percentile to the value of the 95th percentile.

Data Distinct observed (e.g., measured) values of a physical process. Data may be factual or not. Data may be subject to uncertainties, such as imprecision in measurement, truncation, and interpretation errors.

Distribution A cumulative distribution function represents the probability that the entity associated with the probability is less than or equal to a certain value [e.g., $F(x) = P(X < x)$, where F is the cumulative distribution function and X is the entity of interest]. Differentiating the cumulative distribution function yields the probability density function, or $f(x) = dF(x)/dx$. The total area under a normalized probability density function is equal to, by definition, 1.0.

Deterministic Pertaining to exactly predictable (or precise) events, the outcome of which is known with certainty if the inputs are known with certainty. As the antitheses of aleatory, this is the type of model most familiar to scientists and engineers and include relationships such as $E=mc^2$, $F=ma$, $F=G\, m_1\, m_2\, /r^2$, etc.

Epistemic Pertaining to the degree of knowledge of models and their parameters. From the Greek *episteme* (knowledge).

Expected Value The expected value, or mean, is the arithmetic sum of the values of a random variable divided by the total number of values. Common nomenclature to indicate the expected value of a variable X is E[X]. The expected value represents a "central" point of a distribution.

Failure A loss of function for a system, subsystem, component, or part.

Failure Cause A causal impact leading, either indirectly or directly, to failure.

Failure Effect An outcome resulting from a failure. A failure may have multiple effects.

Failure Mode	A specific type of failure, describing the manner of failure.
Generic Data	Surrogate or non-specific information related to a class of parts, components, subsystems, or systems.
Homogeneous	A set of information made up of similar constituents. A homogeneous population is one in which each item is of the same type.
Inference	The process of obtaining a conclusion based on what one knows.
Information	The result of evaluating, manipulating, or organizing data/information in a way that adds to knowledge.
Knowledge	What is known from gathered information.
Model	A mathematical construct that converts information (including data as a subset of information) into knowledge. Two types of models are used for safety analysis purposes, aleatory and deterministic.
Odds	The odds of an event is determined by taking the ratio of the event probability and one minus the event probability, or: Odds = Event Probability / (1 – Event Probability).
Part	The smallest piece of hardware contained in a component. It refers to one piece of hardware, or a group of pieces, that cannot be disassembled without destruction. Examples of a part are a nut, bolt, resister, or electronic chip.
Percentile	A percentile, p, is a specific value x such that approximately p% of the uncertainty is lower than x and (100-p)% of the uncertainty is larger than x. Common percentiles used in PRA include the lower-bound (5th percentile), the median (50th percentile), and upper-bound (95th percentile).
Reliability	The degree of plausibility that an item (e.g., component or system) would not fail during a specified time (or mission).
Subsystem	A group of components comprised as a portion of a system.
System	A group of subsystems and components organized as a whole. A system may have one or more functions.

B. Probability Fundamentals

B.1 Events

Any process, hypothetical or not, for which the result is uncertain can be considered an experiment, such as counting failures over time or measuring failure times. The result of one experiment is an **outcome**. Experiment trials would not be expected to produce the same outcomes due to process variations or other causal factors. The set of all possible outcomes is the sample space, which can be discrete (e.g., pass, fail) or a continuum (e.g., time to failure). An **event** E is a specified set of outcomes in a sample space S (denoted $E \subset S$, where $\subset$ denotes a subset).

Many events of interest are compound events, formed by some composition of two or more events. Composition of events can occur through the union, intersection, or complement of events, or through combinations of these.

For two events, E_1 and E_2, in a sample space S, the **union** of E_1 and E_2 is the event containing all sample points in E_1 or E_2 or both, and is denoted by the symbol $(E_1 \cup E_2)$. A union is the event that either E_1 or E_2 or both E_1 and E_2 occur.

The **intersection** of E_1 and E_2 is the event containing all sample points that are in both E_1 and E_2, denoted by the symbol $(E_1 \cap E_2)$. The intersection is the event that both E_1 and E_2 occur.

Figure B.1 Venn diagram with showing 3 events and 10 outcomes.

Figure B.1 shows a picture, called a Venn diagram, of some outcomes and events. In this example, the event E_1 contains three outcomes, event E_2 contains five outcomes, the union contains seven outcomes, and the intersection contains one outcome.

The **complement** of an event E is the collection of all sample points in S and not in E. The complement of E is denoted by the symbol $\overline{E}$ (alternatively /E) and is the event that all the outcomes in S that are not in E occur.

It is sometimes useful to speak of the empty or null set, a set containing no outcomes. In Figure B.1, the event E_3 is empty. It does not occur.

Two events, E_1 and E_2, are said to be mutually exclusive if the event $(E_1 \cap E_2)$ contains no outcomes in the sample space S. That is, the intersection of the two events is the null set. Mutually exclusive events are also referred to as disjoint events. Three or more events are called mutually exclusive, or disjoint, if each pair of events is mutually exclusive. In other words, no two events can happen together.

B.2 Probability Concepts

Each of the sample space outcomes has a probability associated with it. **Probabilities of outcomes** are seldom known; they are usually estimated via Bayesian methods. Once determined, the probabilities must satisfy two requirements:

- The probability of each outcome must be a number $x : 0 \leq x \leq 1$.

- The probabilities of all outcomes in a given sample space must sum to one.

Associated with any event *E* of a sample space *S* is the **probability of the event**, Pr(*E*). Probabilities are associated with each outcome in the sample space through a **probability model**. Probability models may be developed based on information derived from outcomes obtained from an experiment (where an "experiment" could be any process such as operating a component or system).

B.3 Basic Rules and Principles of Probability

The definition of probability satisfies the following axioms:

- If Pr(*E*) is defined for a type of subset of the sample space *S*, and if Pr(*E*) ≥ 0, for every event *E*, and

- $Pr(E_1 \cup E_2 \cup ...) = Pr(E_1) + Pr(E_2) + ...$, where the events E_1, E_2, . . . , are such that no two have a point in common, and

- Pr(*S*) = 1,

then Pr(*E*) is called a **probability function**.

This probability function defines how the probability is distributed over various subsets *E* of a sample space *S*. From this definition, several rules of probability follow that provide additional properties of a probability function.

The probability of an impossible event (the empty or null set) is zero, written as:

Pr(∅) = 0, where ∅ is the null set. The probability of the **complement** of *E* is given by:

$$Pr(E) = 1 - Pr(E) = \overline{E} .$$

In general, the probability of the union of any two events is given by:

$$Pr(E_1 \cup E_2) = Pr(E_1) + Pr(E_2) - Pr(E_1 \cap E_2) .$$

If E_1 and E_2 are mutually exclusive, then $Pr(E_1 \cap E_2) = $ Pr(∅) = 0, and

$$Pr(E_1 \cup E_2) = Pr(E_1) + Pr(E_2) ,$$

which is a special case of the second axiom of probability stated above and is sometimes referred to as the addition rule for probabilities.

For three events,

$$\Pr(E_1 \cup E_2 \cup E_3) = \Pr(E_1) + \Pr(E_2) + \Pr(E_3) - \Pr(E_1 \cap E_2) - \Pr(E_1 \cap E_3) - \Pr(E_2 \cap E_3) + \Pr(E_1 \cap E_2 \cap E_3)$$
.

This rule is also referred to as the **inclusion-exclusion principle** and can be generalized to *n* events.

The simultaneous occurrence of two or more events (the intersection of events) is called a joint event, and its probability is called a **joint probability**. Thus, the joint probability of both events E_1 and E_2 occurring simultaneously is denoted by $\Pr(E_1 \cap E_2)$.

The probability associated with one event, irrespective of the outcomes for the other events, can be obtained by summing all the joint probabilities associated with all the outcomes for the other events, and is referred to as a **marginal probability**. A marginal probability is therefore the *unconditional* probability of an event, unconditioned on the occurrence of any other event.

Two events E_1 and E_2 are often related in such a way that the probability of occurrence of one depends on whether the other has or has not occurred. The **conditional probability** of one event, given that the other has occurred, is equal to the joint probability of the two events divided by the marginal probability of the given event. Thus, the conditional probability of event E_2, given event E_1 has occurred, denoted $\Pr(E_2|E_1)$, is:

$$\Pr(E_2 \mid E_1) = \frac{\Pr(E_1 \cap E_2)}{\Pr(E_1)},$$

for $\Pr(E_1) > 0$. If $\Pr(E_1) = 0$, $\Pr(E_2|E_1)$ is undefined.

Rearranging this equation yields:

$$\Pr(E_1 \cap E_2) = \Pr(E_1)\Pr(E_2 \mid E_1) = \Pr(E_2)\Pr(E_1 \mid E_2).$$

Calculation of joint probability requires the concept of **statistical independence**. Two events E_1 and E_2 are statistically independent if the probability of one event does not change whenever the other event occurs or does not occur. Thus, E_2 is **independent** of E_1 if

$$\Pr(E_2 \mid E_1) = \Pr(E_2).$$

If E_2 is independent of E_1, then E_1 is independent of E_2. It follows that events E_1 and E_2 are independent if their joint probability is equal to the product of the *unconditional*, or *marginal*, probabilities of the events:

$$\Pr(E_1 \cap E_2) = \Pr(E_1)\Pr(E_2),$$

which is sometimes referred to as the multiplication rule for probabilities. If $\Pr(E_1)$ varies depending on whether or not event E_2 has occurred, then events E_1 and E_2 are said to be **statistically dependent**.

If E_1, E_2, ... are mutually exclusive, and if the union of E_1, E_2, ... equals the entire sample space, then the events E_1, E_2, ... are said to form a **partition** of the sample space. Exactly one of the events must occur, not more than one but exactly one. In this case, the **law of total probability** says

$$\Pr(A) = \sum \Pr(A \mid E_i)\Pr(E_i) \ .$$

A special case can be written when there are only two sets. In this case, write E_1 as E and E_2 as $/E$.

Then the law of total probability simplifies to

$$\Pr(A) = \Pr(A \mid E)\Pr(E) + \Pr(A \mid /E)\Pr(/E)$$

for event A.

The concepts of mutually exclusive events and statistically independent events are often confused. If E_1 and E_2 are mutually exclusive events and $\Pr(E_1)$ and $\Pr(E_2)$ are nonzero, $\Pr(E_1 \cap E_2) = \Pr(\varnothing) = 0$. From the conditional probability equation: $\Pr(E_2 \mid E_1) = \dfrac{\Pr(E_1 \cap E_2)}{\Pr(E_1)}$ = 0, which does not equal $\Pr(E_2)$. Thus, the two events are not statistically independent. Mutually exclusive events cannot be statistically independent and vice versa.

B.4 Random Variables and Probability Distributions

B.4.1 Random Variables

A **random variable** is any mathematical construct that associates real numbers with observable outcomes of an experiment. If the numbers associated with the outcomes of an experiment are all distinct and countable, the corresponding random variable is called a **discrete** random variable. An example of a discrete random variable would be the variable tracking the outcomes of a coin toss, either heads or tails.

If the sample space contains an infinite number of outcomes (like those contained in any interval), the random variable is **continuous**. Time T is a common continuous random variable, for example time to failure where the random variable T can assume any value over the range 0 to ∞.

B.4.2 Probability Distributions

A **probability function** associates a probability with each possible value of a random variable and, thus, describes the distribution of probability for the random variable. For a discrete random variable, this function is referred to as a **discrete probability distribution function** (PDF). A discrete PDF, commonly denoted by *f*, is also referred to as a **discrete distribution**, or **discrete probability mass function**, for example:

- The discrete random variable is X

- This random variable can take on a value x

- The probability distribution function is denoted f(x) [alternatively, as Pr(x)]

The notation used here is that a random variable itself is given in *upper case* while an observed value (data) of the random variable is denoted in *lower case*.

Note that the sum of the probabilities over all the possible values of x must be 1.

B-4

Some discrete random variables have wide application and have therefore been defined and given specific names. The two most commonly used discrete random variables in PRA applications are the **binomial** and **Poisson** random variables. These are also called aleatory models.

A continuously distributed random variable has a **density function**, a nonnegative integral function, with the area between the graph of the function and the horizontal axis equal to 1. This density function is also referred to as the **continuous** PDF. If x denotes a value that the continuous random variable X can assume, the PDF is often denoted as $f(x)$. The probability that X takes a value in a region A is the integral of $f(x)$ over A. In particular,

$$\Pr(a \le X \le b) = \int_a^b f(x)dx$$

and

$$\Pr(x \le X \le x + \Delta x) \approx f(x)\Delta x$$
$$\text{(B.1)}$$

for small Δx.

Some common continuous distributions in PRA are the **lognormal**, **exponential**, **gamma**, and **beta** distributions.

B.4.3 Cumulative Distribution Functions

Discrete probability distributions provide point probabilities for discrete random variables and continuous PDFs provide point densities for continuous random variables. A related function is the **cumulative distribution function** (CDF). This function is defined as the probability that the random variable assumes values less than or equal to the specific value x, and is denoted $F(x)$.

For a discrete random variable X, with outcomes x_i, and the corresponding probabilities $\Pr(x_i)$, $F(x)$ is the sum of the probabilities of all $x_i \le x$. That is,

$$F(x) = \Pr(X \le x) = \sum_{x_i \le x} \Pr(x_i).$$

For a continuous random variable X, $F(x)$ is the area beneath the PDF $f(x)$ up to x. $F(x)$ is the integral of $f(x)$:

$$F(x) = \Pr(X \le x) = \int_{-\infty}^x f(y)dy.$$

Thus, $f(x)$ is the derivative of $F(x)$. If X takes on only positive values, the limits of integration are zero to x. Note that, because $F(x)$ is a probability, $0 \le F(x) \le 1$. If X ranges from $-\infty$ to $+\infty$, then $F(-\infty) = 0$ and $F(+\infty) = 1$.

If X has a restricted range, with a and b being the lower and upper limits of X respectively, $a < X < b$, then

$F(a) = 0$ and $F(b) = 1$.

B-5

Also, *F(x)* is a non-decreasing function of *x*, that is, if $x_2 > x_1, F(x_2) \geq F(x_1)$. Another important property of *F(x)* is that

$$\Pr(x_1 < X \leq x_2) = F(x_2) - F(x_1)$$

for discrete random variables and

$$\Pr(x_1 < X \leq x_2) = F(x_2) - F(x_1)$$

for continuous random variables.

An example of a PDF and the associated CDF for a continuous distribution is shown in Figure B.2.
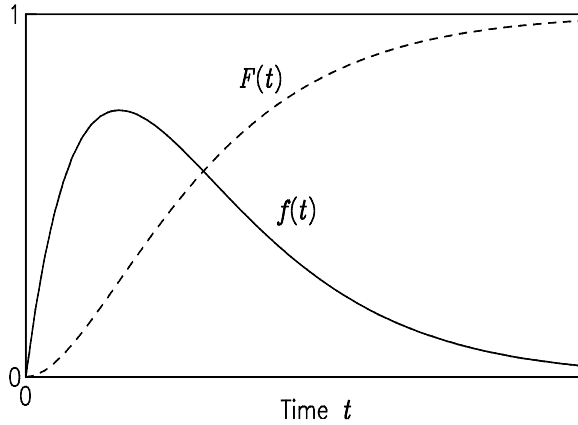
Figure B.2: Probability density function (PDF) and cumulative distribution function (CDF)

B.4.4 Reliability and Hazard Functions

Definitions

There are also characterizations that have special interpretations for time-to-failure distributions. Let *T* denote the random time to failure of a system. The **reliability function** of a system is defined as

$$R(t) = \Pr(T > t) \ .$$

Hence, *R(t)*, called the **reliability** at time *t*, is the probability that a system does not fail in the time interval [0, *t*] or equivalently, the probability that the system is still operating at time *t*. (This discussion uses the notation (*a*, *b*) to mean the set of times > *a* and ≤ *b*, but the distinction between < and ≤ is a mathematical fine point, not important in practice.) The reliability function is also sometimes called the **survival function**. It is equal to $1 - F(t)$.

When used as a reliability criterion, it is common to state a time, say t_0, called the **mission time**, and require that the reliability at mission time t_0 be at least some prescribed level, say R_0. For example, a power unit might be required to operate successfully for at least 12 hours with probability at least 0.95. The requirement in this case is $R_0 = 0.95$ and $t_0 = 12$. In terms of the reliability function, this would

B-6

mean $R(12h) \geq 0.95$. One interpretation would be that such a power unit would perform for the required mission time for 95% of the situations when it is called on to do so. Another interpretation is that 95% of all such power units would perform as required.

Consider a system that operates for a particular mission time, unless it fails. If it fails, no immediate repairs are attempted, so some authors call the system **nonrepairable**. A common way to characterize this system's reliability is in terms of the **hazard function**. Suppose that the system is still operating at time t, and consider the probability that it will fail in a small interval of time $(t, t + \Delta t)$. This is the conditional probability $\Pr(t < T \leq t + \Delta t \mid T > t)$. The hazard function, h, is defined so that when Δt is small:

$$h(t)\Delta t \approx \Pr(t < T \leq t + \Delta t \mid T > t)$$
    (B.2)

This function is also encountered, under the name of λ, in some treatments of Poisson processes. The hazard function gives, approximately,

$$h(t)\Delta t \approx \frac{\Pr(t < T \leq t + \Delta t)}{\Pr(T > t)} \approx \frac{f(t)\Delta t}{R(t)}$$

This is the basis for the formal definition of h:

$$h(t) = \frac{f(t)}{R(t)}$$

For details, see (Bain and Engelhardt 1992, p. 541). Equation B.2 is analogous to Equation B.1, except that the probability in Equation B.2 is conditional on the system having survived until t, whereas Equation B.1 refers to all systems in the original population, either still surviving or not. Suppose a large number, say N, of identical systems are put into operation at time $t = 0$, and n is the number which fail in the interval $(t, t + \Delta t)$. It follows that $f(t)\Delta t \approx \dfrac{n}{N}$, the observed relative frequency of systems failed in the interval $(t, t + \Delta t)$. On the other hand, if N_t denotes the number of the original N systems which are still in operation at time t, then $h(t)\Delta t \approx \dfrac{n}{N_t}$, the observed relative frequency of *surviving* systems which fail in this same interval. Thus, $f(t)$ is a measure of the risk of failing at time t for any system in the original set, whereas $h(t)$ is a measure of the risk of failing at time t, but only for systems that have survived this long.

The hazard function is used as a measure of "aging" for systems in the population. If $h(t)$ is an increasing function, then systems are aging or wearing out with time. Of course, in general the hazard function can exhibit many types of behavior other than increasing with time, and other possible behaviors are discussed later in this handbook. In actuarial science the hazard function is called the **force of mortality**, and it is used as a measure of aging for individuals in a population. More generally, the hazard function gives an indication of "proneness to failure" of a system after time t has elapsed. Other terms which are also used instead of hazard function are **hazard rate** and **failure rate**. The term *failure rate* is often used in other ways in the literature of reliability (Ascher and Feingold 1984, p. 19).

Relations among PDF, Reliability, and Hazard

Any one of the functions F, f, R, and h completely characterizes the distribution, and uniquely determines the other three functions. The definition

$$h(t) = \frac{f(t)}{R(t)}$$

was given above. The right hand side can be written as the derivative of $-\ln[R(t)]$, leading to

$$R(t) = \exp\left(-\int_0^t h(u)\,du\right) = \exp(-H(t))$$

where the function $H(t)$ is called the **cumulative hazard function**. The reliability function, $R(t)$, and the CDF, $F(t) = 1 - R(t)$, are therefore uniquely determined by the hazard function, $h(t)$, and the PDF can be expressed as

$$f(t) = h(t)\exp\left(-\int_0^t h(u)\,du\right).$$

Figure B.3 shows the reliability, hazard and the cumulative hazard function for the example of Figure B.2.
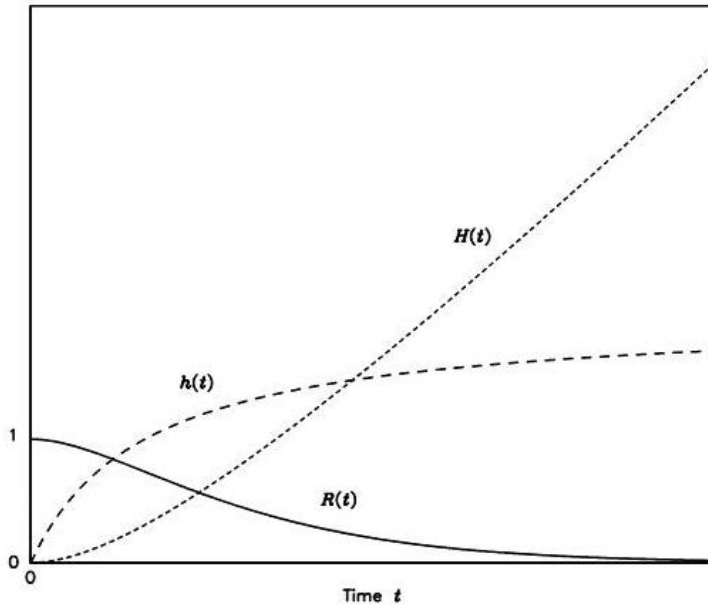
Figure B.3 The reliability function, hazard function and cumulative hazard function.

The hazard function in Figure B.3 is an increasing function of time. Therefore, it would be consistent with systems with a dominant wear-out effect for the entire life of the system. The lifetime of a system may be divided into three typical intervals: the **burn-in** or **infant** period, the **random** or **chance** failure period and the **wear-out** period. During the useful period, the dominant cause of failures is "random" failures. For example, systems might fail due to external causes such as power surges or other environmental factors rather than problems attributable to the defects or wear-out in the systems. This example is somewhat idealized because for many types of systems the hazard function will tend to increase slowly during the later stages of the chance failure period. This is particularly true of mechanical systems. On the other hand, for many electrical components such as transistors and other solid-state devices, the hazard function remains fairly flat once the burn-in failure period is over.

B.4.5 Joint, Marginal, and Conditional Distributions

Many statistical methods are based on selecting a sample of size n from a probability distribution $f(x)$. Such a sample is denoted by

$(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = (x_1, x_2, ..., x_n),$

where $x_1, x_2, ..., x_n$ are the actual values of the random variable X which has the distribution $f(x)$.

The concepts of simultaneous events and joint, marginal, and conditional probability, discussed in Section B.3, also pertain to random variables and probability distributions. Two random variables X_1 and X_2 (both continuous, both discrete, or one of each) can have a **joint distribution**, or joint PDF, denoted $f(x_1, x_2)$. The point (x_1, x_2) can be thought of as a point in two-dimensional Euclidean space. Similarly, n random variables have joint distribution $f(x_1, x_2, ..., x_n)$. Also, the n random variables have joint cumulative distribution $F(x_1, x_2, ..., x_n)$.

The **marginal distribution** of X_i is defined as the joint PDF integrated (for continuous random variables) or summed (for discrete random variables) over the $n - 1$ other corresponding dimensions, resulting in a function of x_i alone. Thus, the marginal distribution of X_i is the unconditional PDF of X_i, $f_i(x_i)$.

The **conditional distribution** of X_1 given X_2, denoted $g(x_1 \mid x_2)$, is defined by

$$g(x_1 \mid x_2) = \frac{f(x_1, x_2)}{f_2(x_2)}$$

where $f_2(x_2) \neq 0$, and can be shown to satisfy the requirements of a probability function. Sampling from a conditional PDF would produce only those values of X_1 that could occur for a given value of $X_2 = x_2$. The concept of a conditional distribution also extends to n random variables.

Two random variables X_1 and X_2 are independent if their joint PDF is equal to the product of the two individual PDFs. That is,

$f(x_1, x_2) = f(x_1) f(x_2).$

In general, $X_1, X_2, ..., X_n$ are **independent random variables** if

$f(x_1, x_2, ..., x_n) = f(x_1) \cdot f(x_2) \cdot .\ .\ .\ \cdot f(x_n).$

B.4.6 Characterizing Random Variables and their Distributions

Distribution Characteristics

Probability distributions have many characteristics of interest, some of which are described by **distribution parameters**. The term parameter is used to refer to a fixed characteristic. In contrast to a statistic, which changes from sample to sample, a parameter for a particular distribution is a constant and does not change. However, when a parameter's value is not known, sample statistics can be used to estimate the parameter value. Parameter estimation is discussed in Appendix B.10.

A very useful distribution characteristic is the parameter that serves as a measure of central tendency, which can be viewed as a measure of the middle of a distribution. When a change in the parameter slides the distribution sideways, as with the mean of a normal distribution, the parameter is referred to as the **location parameter**. It serves to locate the distribution along the horizontal axis. Sometimes, however, a change in the parameter squeezes or stretches the distribution toward or away from zero, as with the mean of the exponential distribution. In that case, the parameter is a **scale parameter**.

In any case, the most common measure of central tendency is the **mean**, μ, of the distribution, which is a weighted average of the outcomes, with the weights being probabilities of outcomes. For a discrete random variable X,

$$\mu_X = \sum_i x_i \Pr(x_i).$$

For a continuous random variable X,

$$\mu_X = \int_{-\infty}^{\infty} x f(x) dx.$$

Another distribution characteristic commonly used as a measure of central tendency, or location, is the **median**, which is the point along the horizontal axis for which 50% of the area under the PDF lies to its left and the other 50% to its right. The median of a random variable, X, is commonly designated med(X) or $x_{.50}$ and, for a continuous distribution, is the value for which $\Pr(X \leq x_{.50}) = .50$ and $\Pr(X \geq x_{.50}) = .50$. In terms of the cumulative distribution, $F(x_{.50}) = .50$. The median is a specific case of the general 100αth **percentile**, x_α, for which $F(x_\alpha) = \alpha$. When the factor of 100 is dropped, x_α is called the α **quantile**. Along with the median as the 50th percentile (or equivalently, the 0.5 quantile), the 25th and 75th percentiles are referred to as **quartiles** of a distribution.

Figure B.4 shows the quartiles, $x_{0.25}$ and $x_{0.75}$, the median, $x_{0.50}$, and the mean. The quartiles and the median divide the area under the density curve into four pieces, each with the same area. Note that the mean is greater than the median in this example, which is the usual relation when the density has a long right tail, as this one does.

Figure B.5 shows the same quantities plotted with the CDF. By definition, the q quantile, x_q, satisfies $F(x_q) = q$.
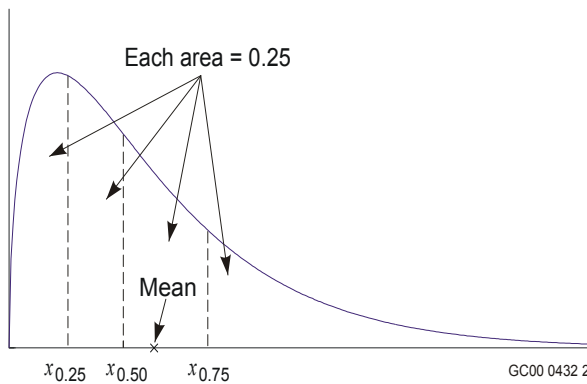
Figure B.4 Density, showing quartiles, median, and mean.
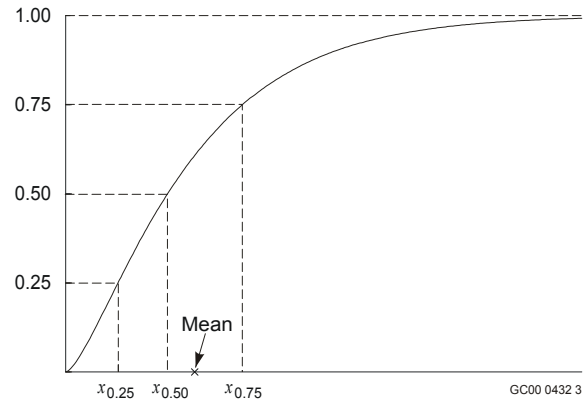
Figure B.5 Cumulative distribution function (CDF) showing quartiles, median, and mean.

B-10

The mean and the median are used to measure the center or location of a distribution. Since the median is less affected by tail-area probabilities, it can be viewed as a better measure of location than the mean for highly-skewed distributions. For symmetric distributions, the mean and median are equivalent.

A different measure of center or location of a distribution is the **mode**, which indicates the most probable outcome of a distribution. The mode is the point along the horizontal axis where the "peak" or maximum of the PDF is located. Note that the mode does not necessarily have to be near the middle of the distribution. It simply indicates the most likely value of a distribution. Note also that a peak does not have to exist and, in some cases, more than one peak can exist.

Another important characteristic of a distribution is its **variance**, denoted by σ^2. The variance is the average of the squared deviations from the mean. The **standard deviation**, σ, of a distribution is the square root of its variance. Both the variance and standard deviation are measures of a distribution's spread or dispersion. For a discrete random variable X,

$$\sigma_X{}^2 = \sum_i (x_i - \mu)^2 \Pr(x_i).$$

For a continuous random variable X,

$$\sigma_X{}^2 = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx \ .$$

Though less used than the mean and variance, the **skewness** is defined as

$$\frac{E(X - \mu)^3}{\sigma^3} \ .$$

It measures asymmetry. It is usually positive if the density has a longer right tail than left tail, and negative if the density has a longer left tail than right tail. For example, the density in Figure B.4 has positive skewness.

Mathematical Expectation

The definitions of distribution means and variances arise from **mathematical expectation** and **moments of a distribution**, which form an important method for calculating the parameters of a known PDF In general, the **expectation** (**expected value** or mathematical expectation) of a function $g(X)$, denoted $E[g(X)]$, is

$$E[g(X)] = \sum_i g(x_i) \Pr(x_i) \ ,$$

when X is discrete, and

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f(x) dx \ ,$$

when X is continuous.

Because of their wide use, several expectations have special names. For $g(X) = X$, the expectation $E(X)$ becomes the mean of X. Thus, the mean is also commonly referred to as the expected value (or expectation) of the random variable X. In addition, for $g(X) = X$, the expectation $E(X)$ is known as the **first moment about the origin**.

The variance, σ_X^2, also denoted by Var(X), of a distribution is defined by mathematical expectation with $g(X) = (X - \mu_X)^2$. Thus,

$$Var(X) = \sigma_X^2 = E\left[(X - \mu_X)^2\right] = E(X^2) - [E(X)]^2 ,$$

which is known as the **second moment about the mean**.

Ordinary moments (moments about the origin) of a random variable X are defined as

$$M_r = E\left(X^r\right),$$

for r = 1, 2, ... Thus,

$$Var(X) = \sigma_X^2 = E\left(X^2\right) - [E(X)]^2 = M_2 - M_1^2 .$$

Central moments (moments about the mean) of a random variable X are defined as being equal to $E\left[(X - \mu)^r\right]$ for r = 2, 3, ... The ordinary and central moments can be seen to define characteristics of distributions of random variables.

An important rule of expectation commonly used in PRA is that the expected value of a product of independent random variables is the product of their respective expected values. That is, $E(X_1 \cdot X_2 \cdot ... \cdot X_n) = E(X_1) \cdot E(X_2) \cdot ... \cdot E(X_n)$ when all X_i are independent. This rule also applies to conditionally independent random variables. If the random variables $X_2, X_3, ..., X_n$ are all conditionally independent given $X_1 = x_1$, then

$$f\left(x_2, x_3, ..., x_n \mid x_1\right) = f(x_2 \mid x_1) A \cdot f(x_3 \mid x_1) A \cdot ... \cdot A f(x_n \mid x_1)$$

It follows that

$$E(X_2 A X_3 A ... A X_n \mid x_1) = E(X_2 \mid x_1) A \cdot E(X_3 \mid x_1) A \cdot ... \cdot A E(X_n \mid x_1).$$

Thus,

$$E(X_1 A \cdot X_2 A \cdot ... \cdot A X_n) = E\left[X_1 A \cdot E(X_2 \mid x_1) A \cdot E(X_3 \mid x_1) A \cdot ... \cdot A E(X_n \mid x_1)\right].$$

Covariance and Correlation

For two random variables, X and Y, with means μ_x and μ_y, the expected value $E[(X - \mu_x)(Y - \mu_y)]$ is called the **covariance** of X and Y, denoted $Cov(X,Y)$. The covariance of X and Y divided by the product of the standard deviations of X and Y is called the **correlation coefficient** (or correlation) between X and Y, denoted $Cor(X,Y)$. That is,

$$Cor(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}} = \frac{E(X - \mu_X)E(Y - \mu_Y)}{\sqrt{E[(X - \mu_X)^2]E[(Y - \mu_Y)^2]}} .$$

The correlation coefficient measures the degree of association between X and Y, that is, the strength of a linear relationship between X and Y.

B.4.7 Distribution of a Transformed Random Variable

This section considers the distribution of $Y = h(X)$, when X has a known distribution and h is a known function. The problem is straightforward when X has a discrete distribution. When X is continuous and h is monotone, either increasing or decreasing, the CDFs are also related in the natural way, as follows. Let F be the CDF of X and let G be the CDF of Y. Then we have

$$G(y) = \Pr(Y \le y) = \Pr\left[h(X) \le y\right].$$

If h is monotone increasing, this equals

$$\Pr\left[X \le h^{-1}(y)\right] = F(x),$$

where x and y are related by $y = h(x), x = h^{-1}(y)$. In summary, $G(y) = F(x)$.

If, instead, h is monotone decreasing, then a similar argument gives

$G(y) = 1 - F(x)$.

The surprise comes with the densities. Differentiate both sides of either of the above equations with respect to y, to obtain the density of y. This involves using the chain rule for differentiation. The result is

$$g(y) = f(x)\left|\frac{dx}{dy}\right|.$$

That is, the density of Y is not simply equal to the density of X with a different argument. There is also a multiplier, the absolute value of the derivative. If $Y = \exp(X)$, then

$$g(y) = f[\ln(y)](1/y)$$

If $Y = 1/X$, then

$$g(y) = f(1/y)(1/y^2).$$

The formulas here are the basis for the densities of the lognormal distribution and the inverted gamma distribution.

B.5 Bayes' Theorem

It is frequently desired to calculate the probability of an event A given that another event B has occurred at some prior point in time. It can also be of interest to calculate the probability that a state exists given that a certain sample is observed, for example, belonging to a certain population based on a sample measurement or observation. Conditional probability leads directly to **Bayes' Theorem**, which, along with subjective probability, forms the basis for Bayesian inference commonly used in PRA.

B-13

Bayes' Theorem states that: if A_1, A_2, ..., A_n are a sequence of disjoint events and if B is any other event such that $\Pr(B) > 0$, then

$$\Pr(A_i \mid B) = \frac{\Pr(B \mid A_i)\Pr(A_i)}{\Pr(B)}$$

(B.3)

Where

$$\Pr(B) = \sum_{j=1}^{n} \Pr(B \mid A_j)\Pr(A_j) \ .$$

Equation B.3 follows from the definition of conditional probability:

$$\Pr(A_i \mid B) = \frac{\Pr(B \cap A_i)}{\Pr(B)} = \frac{\Pr(B \mid A_i)}{\Pr(B)}$$

The $\Pr(A_i \mid B)$ is the **posterior** (or a posteriori) probability for the event A_i, meaning the probability of A_i once B is known. The $\Pr(A_i)$ is the **prior** (or a priori) probability of the event A_i before experimentation or observation. The event B is the observation. The $\Pr(B \mid A_i)$ is the probability of the observation given that A_i is true. The denominator serves as a normalizing constant.

Calculating the posterior probabilities $\Pr(A_i|B)$ requires knowledge of the probabilities $\Pr(A_i)$ and $\Pr(B|A_i)$, i = 1, 2, ..., n. The probability of an event can often be determined if the population is known, thus, the $\Pr(B|A_i)$ can be determined. However, the $\Pr(A_i)$, i = 1, 2, ..., n, are the probabilities that certain states exist and are either unknown or difficult to ascertain. These probabilities, $\Pr(A_i)$, are called prior probabilities for the events A_i because they specify the distribution of the states prior to conducting the experiment.

Application of Bayes' Theorem utilizes the fact that $\Pr(B|A_i)$ is easier to calculate than $\Pr(A_i|B)$. If probability is viewed as degree of belief, then the prior belief is changed, by the test evidence, to a posterior degree of belief. In many situations, some knowledge of the prior probabilities for the events A_1, A_2, ..., A_n exists. Using this prior information, inferring which of the set A_1, A_2, ..., A_n, is the true population can be achieved by calculating the $\Pr(A_i|B)$ and selecting the population that produces the highest probability.

Equation B.3 pertains to disjoint discrete events and discrete probability distributions. Bayes' Theorem has analogous results for continuous PDF's. Suppose X is a continuous random variable, with PDF depending on parameter θ, and with conditional PDF of X, given θ, specified by $f(x|\theta)$. Consider θ to be a possible value of the random variable Θ (using the convention of denoting random variables with uppercase letters). If the prior PDF of Θ is denoted $g(\theta)$, then for every x such that $f(x) > 0$ exists, the posterior PDF of Θ, given $X = x$, is

$$g(\theta \mid x) = \frac{f(x \mid \theta)g(\theta)}{f(x)} \ ,$$

Where

$$f(x) = \int f(x \mid \theta) g(\theta) d\theta$$

is the marginal PDF of *X*. Again, the prior and posterior PDFs can be used to represent the knowledge and beliefs about the likelihood of various values of a random variable Θ prior to and posterior to observing a value of another random variable *X*.

B.6 Discrete Random Variables

B.6.1 Binomial Distribution

The **binomial** distribution (Bernoulli model) describes the number of failures *X* in *n* independent trials. The random variable *X* has a binomial distribution if:

1. The number of random trials is one or more and is known in advance.

2. Each trial results in one of two outcomes, usually called success and failure (or could be pass-fail, hit-miss, defective-nondefective, etc.).

3. The outcomes for each trial are statistically independent.

4. The probability of failure, *p*, is constant across trials.

Equal to the number of failures in the *n* trials, a binomial random variable *X* can take on any integer value from 0 to *n*. The probability associated with each of these possible outcomes, *x*, is defined by the binomial(*n*, *p*) PDF as

$$\Pr(X = x) = \binom{n}{x} p^x (1-p)^{n-x} \quad , \quad x = 0,...,n$$

Here

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

is the **binomial coefficient** and

n! = *n*(*n* −1)(*n* −2) ... (2)(1)

denotes *n* factorial, with 0! defined to be equal to 1. This binomial coefficient provides the number of ways that exactly *x* failures can occur in *n* trials (number of combinations of *n* trials selected *x* at a time).

The binomial distribution has two parameters, *n* and *p*, of which *n* is known. (Although *n* may not always be known exactly, it is treated as known in this handbook.)

The mean and variance of a binomial(*n*, *p*) random variable *X* are

E(*X*) = *np*

and

$$\text{Var}(X) = np(1 - p).$$
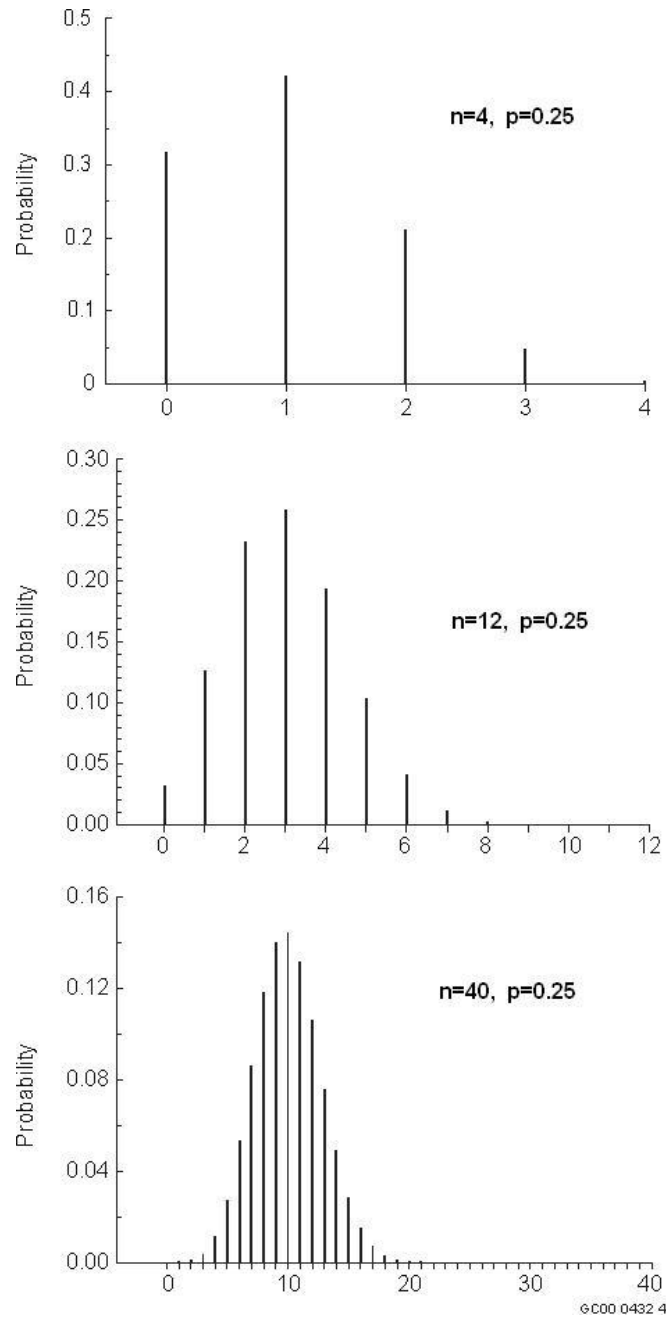
Figure B.6 Three binomial probability distribution functions.

Figure B.6 shows three binomial probability distribution functions, with parameter p = 0.25, and n = 4, 12, and 40. In each case, the mean is np. The means have been aligned in the three plots.

B.6.2 Poisson Distribution

The **Poisson** distribution provides a discrete probability model that is appropriate for many random phenomena that involve counts. Examples are counts per fixed time interval of the number of items that fail, the number of customers arriving for service, and the number of telephone calls occurring. A common use of the Poisson distribution is to describe the behavior of many rare event occurrences. The Poisson distribution is also frequently used in applications to describe the occurrence of system or component failures under steady-state conditions.

The count phenomena that occur as Poisson random variables are not necessarily restricted to occurring over a time interval. They could also be counts of things occurring in some region, such as defects on a surface or within a certain material. A process that leads to a Poisson random variable is said to be a **Poisson process**.

The Poisson distribution describes the total number of events occurring in some interval of time t (or space). The PDF of a Poisson random variable X, with parameter $\mu = \lambda t$, is

$$\Pr(X = x) = \frac{e^{-\mu}\mu^x}{x!} = \frac{e^{-\lambda t}(\lambda t)^x}{x!}$$

(B.4)

for $x = 0, 1, 2, ...$, and $x! = x(x - 1)(x - 2) ... (2)(1)$, as defined previously.

The Poisson distribution has a single parameter μ, denoted Poisson(μ). If X denotes the number of events that occur during some time period of length t, then X is often assumed to have a Poisson distribution with parameter $\mu = \lambda t$. In this case, X is considered to be a Poisson process with **intensity** $\lambda > 0$ (Martz and Waller 1991). The variable λ is also referred to as the **event rate** (or **failure rate** when the events are failures). Note that λ has units 1/time; thus, $\lambda t = \mu$ is dimensionless.

If only the total number of occurrences for a single time period t is of interest, the form of the PDF in Equation B.4 using μ is simpler. If the event rate, λ, or various time periods, t, are of interest, the form of the PDF in Equation B.4 using λt is more useful.

The expected number of events occurring in the interval 0 to t is $\mu = \lambda t$. Thus, the mean of the Poisson distribution is equal to the parameter of the distribution, which is why μ is often used to represent the parameter. The variance of the Poisson distribution is also equal to the parameter of the distribution. Therefore, for a Poisson(μ) random variable X,

$E(X) = \text{Var}(X) = \mu = \lambda t.$

Figure B.7 shows three Poisson probability distribution functions, with means $\mu = 1.0, 3.0,$ and 10.0, respectively. The three means have been aligned in the graphs. Note the similarity between the Poisson distribution and the binomial distribution when $\mu = np$ and n is not too small.

Several conditions are assumed to hold for a Poisson process that produces a Poisson random variable:

For small intervals, the probability of exactly one occurrence is approximately proportional to the length of the interval (where $ë$, the event rate or intensity, is the constant of proportionality).

For small intervals, the probability of more than one occurrence is essentially equal to zero (see below).

The numbers of occurrences in two non-overlapping intervals are statistically independent.

B-18
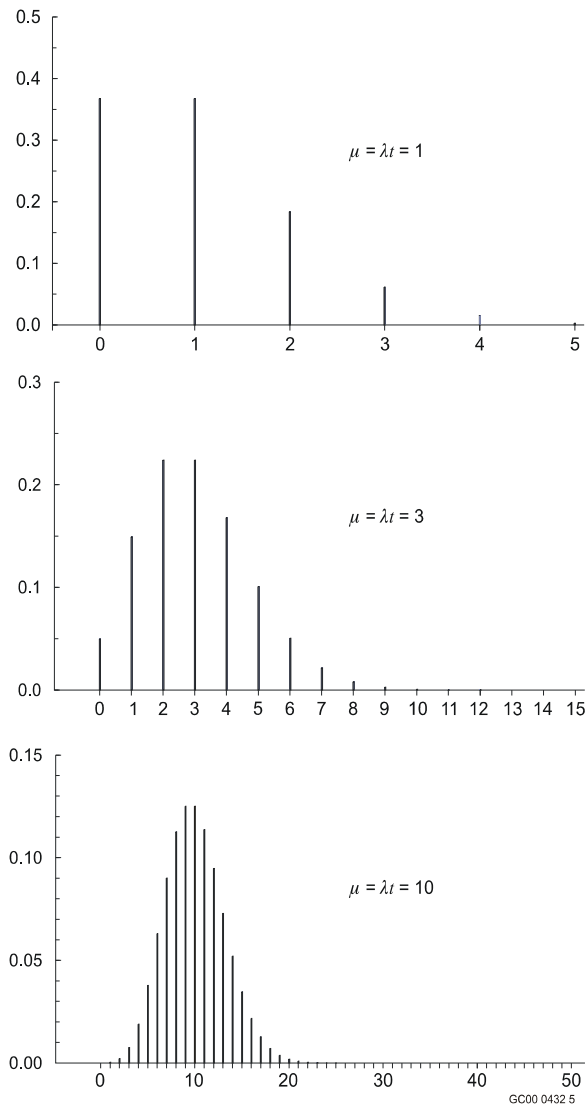
Figure B.7 Three Poisson probability distribution functions.

More precise versions of condition 2 are: (1) the probability of more than one event occurring in a very short time interval is negligible in comparison to the probability that only one event occurs (Meyer 1970), (2) the probability of more than one event occurring in a very short time interval goes to zero faster than the length of the interval (Pfeiffer and Schum 1973), and (3) simultaneous events occur only with probability zero (Çinlar 1975).

The Poisson distribution also can serve as an **approximation to the binomial distribution**. Poisson random variables can be viewed as resulting from an experiment involving a large number of trials, *n*, that each have a small probability of occurrence, *p*, of an event. However, the rare occurrence is offset by the large number of trials. As stated above, the binomial distribution gives the probability that an occurrence will take place exactly *x* times in *n* trials. If $p = \mu/n$ (so that *p* is small for large *n*), and *n* is large, the binomial probability that the rare occurrence will take place exactly *x* times is closely approximated by the Poisson distribution with $\mu = np$. In general, the approximation is good for large *n*, small *p*, and moderate *μ* (say $\mu \leq 20$) (Derman et al. 1973).

The Poisson distribution is important because it describes the behavior of many rare event occurrences, regardless of their underlying physical process. It also has many applications to describing the occurrences of system and component failures under steady-state conditions. These applications utilize the relationship between the Poisson and **exponential** (continuous random variable, see Section B.7.4) distributions: the times between successive events follow an exponential distribution.

B.7 Continuous Random Variables

B.7.1 Uniform Distribution

A **uniform** distribution, also referred to as a rectangular distribution, represents the situation where any value in a specified interval, say [*a*, *b*], is equally likely. For a uniform random variable, *X*, because the outcomes are equally likely, *f(x)* is equal to a constant. The PDF of a uniform distribution with parameters *a* and *b*, denoted uniform(*a*, *b*) is

$$f(x) = \frac{1}{b-a}$$

for $a \le x \le b$.

Figure B.8 shows the density of the uniform(a, b) distribution.

The mean and variance of a uniform(a, b) distribution are

$$E(X) = \frac{b+a}{2}$$

And

$$Var(X) = \frac{(b-a)^2}{12}$$

Area = 1

1/(b-a)

0

a b GC00 0432 6
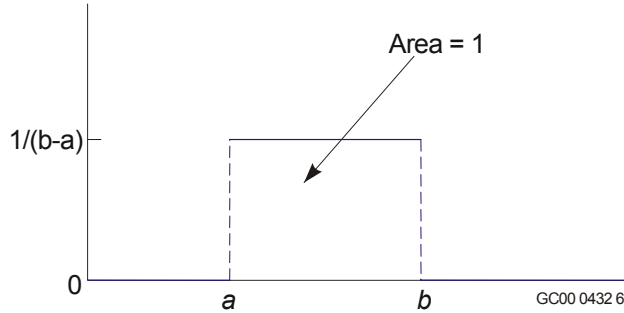
Figure B.8 Density of uniform(a, b) distribution.

B.7.2 Normal Distribution

One of the most widely encountered continuous probability distributions is the **normal** distribution, which has the familiar bell shape and is symmetrical about its mean value. The importance of the normal distribution is due to: (1) its applicability in describing a very large number of random variables that occur in nature and (2) the fact that certain useful functions of nonnormal random variables are approximately normal. Details on the derivation of the normal distribution can be found in many basic mathematical statistics textbooks (e.g., Hogg and Craig 1995).

The normal distribution is characterized by two parameters, μ and σ. For a random variable, X, that is normally distributed with parameters μ and σ, the PDF of X is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

(B.5)

for $-\infty < x < \infty$, $-\infty < \mu < \infty$, and $\sigma > 0$. Increasing μ moves the density curve to the right and increasing σ spreads the density curve out to the right and left while lowering the peak of the curve. The units of μ and σ are the same as for X.

The mean and variance of a normal distribution with parameters μ and σ are

$E(X) = \mu$

and

$$Var(X) = \sigma^2.$$

The normal distribution is denoted normal(μ, σ^2).

B-20

Figure B.9 shows two normal(μ, σ^2) densities. The distribution is largest at μ and is more concentrated around μ when σ is small than when σ is large.

Note the similarity of the normal density to a binomial PDF with large np or a Poisson PDF with large μ. This illustrates the fact that a normal distribution can sometimes be used to approximate those distributions.
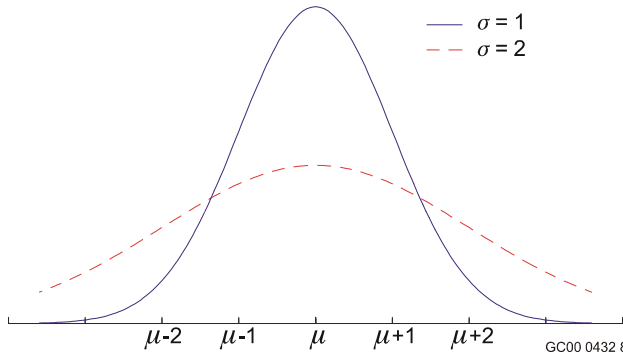
The normal(0, 1) distribution is called the **standard normal** distribution, which, from Equation B.5, has PDF

$$\phi(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$$

(B.6)

for $-\infty < x < \infty$.

Figure B.9 Two normal densities.

The cumulative distribution of the standard normal distribution is denoted by Φ. Tables for the standard normal distribution are presented in almost all books on statistics.

It can be shown that the transformed random variable $Z = (X - \mu)/\sigma$ is normal(0, 1). Thus, to calculate probabilities for a normal(μ, σ^2) random variable, X, when $\mu \neq 0$ and/or $\sigma^2 \neq 1$, the tables for the standard normal can be used. Specifically, for any number a,

$$\Pr[X \leq a] = \Pr\left[\frac{(X-\mu)}{\sigma} \leq \frac{(a-\mu)}{\sigma}\right] = \Pr\left[Z \leq \frac{(a-\mu)}{\sigma}\right] = \Phi\left[\frac{(a-\mu)}{\sigma}\right]$$

Part of the importance of the normal distribution is that it is the distribution that sample sums and sample means tend to possess as n becomes sufficiently large. This result is known as the **central limit theorem**, which states that, if X_1, X_2, ..., X_n, are independent random variables, each with mean μ

and variance σ^2, the sum of these n random variables, $\Sigma_i X_i$, tends toward a normal($n\mu$, $n\sigma^2$) distribution for large enough n. Since the sample mean is a linear combination of this sum, the central limit

theorem also applies. Thus, $\overline{X} = \Sigma_i X_i/n$ tends to a normal(μ, σ^2/n) distribution. The importance of the central limit theorem is it can be used to provide approximate probability information for the sample sums and sample means of random variables whose distributions are unknown. Further, because many natural phenomena consist of a sum of several random contributors, the normal distribution is used in many broad applications.

Because a binomial random variable is a sum, it tends to the normal distribution as n gets large. Thus, the normal distribution can be used as an **approximation to the binomial distribution**. One rule of thumb is that the approximation is adequate for $np \geq 5$.

A Poisson random variable also represents a sum and, as presented previously, can also be used as an approximation to the binomial distribution. It follows that the normal distribution can serve as an **approximation to the Poisson distribution** when $\mu = \lambda t$ is large. One rule of thumb is that the approximation is adequate for $\mu \geq 5$.

B-21

B.7.3 Lognormal Distribution

Use of the **lognormal distribution** has become increasingly widespread. It is commonly used as a distribution for failure time and in maintainability analysis (Martz and Waller 1991). It has also been widely used as a prior distribution for unknown positive parameters.

The lognormal distribution arises from the *product* of many independent random variables.   If $Y = Y_1 \cdot Y_2 \cdot ... \cdot Y_n = \prod_i Y_i$ is the product of *n* independent positive random variables that are (nearly) identically distributed, then $\ln(Y) = \ln\left(\prod_i Y_i\right) = \sum_i \ln(Y_i)$ is a sum that tends toward a normal distribution.

The distribution of *Y* is defined to be lognormal when the distribution of ln(*Y*) is normal. That is, when *Y* is lognormal, ln(*Y*) is normal(μ, σ^2). The parameters of the lognormal distribution are μ and σ, the parameters from the underlying normal distribution. For a random variable, *Y*, that is lognormally distributed with parameters μ and σ, denoted lognormal(μ, σ^2), the PDF of *Y* is

$$f(y) = \frac{1}{\sigma y \sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2}(\ln(y) - \mu)^2 \right]$$

for $0 < y < \infty$, $-\infty < \mu < \infty$, and $\sigma > 0$. Note the *y* in the denominator, for reasons explained in Section B.4.7. The mean and variance of a lognormal(μ, σ^2) distribution are

$$E(Y) = \exp\left[\frac{(\mu + \sigma^2)}{2} \right]$$

and

$$Var(Y) = \exp(2\mu + \sigma^2)\left[\exp(\sigma^2) - 1\right].$$

In addition, the median of a lognormal distribution is exp(μ) and the mode is exp($\mu - \sigma^2$). See Martz and Waller (1991) for more information on the lognormal distribution.

Sometimes the median of *Y* = exp(μ) is used as a parameter. In addition, a parameter commonly used in PRA is the **error factor** (EF), where

$$EF = \exp(1.645\sigma),$$

and is defined as

$$\Pr\left[\frac{med(Y)}{EF} \leq Y \leq med(Y) * EF \right] = 0.90 .$$

Figure B.10 shows three lognormal densities. The value μ = −7 corresponds to a median of about 1.E−3. [More exactly, it corresponds to exp(−7) = 9.E−4.] The value μ = −6.5 corresponds to a median of about 1.5E−3. The value σ = 0.67 corresponds to an error factor EF = 3, and σ = 1.4 corresponds to an error factor EF = 10.

$\mu = -7, \; \sigma = 1.4$
$\mu = -7, \; \sigma = 0.67$
$\mu = -6.5, \; \sigma = 0.67$

0E0   1E-3   2E-3   3E-3
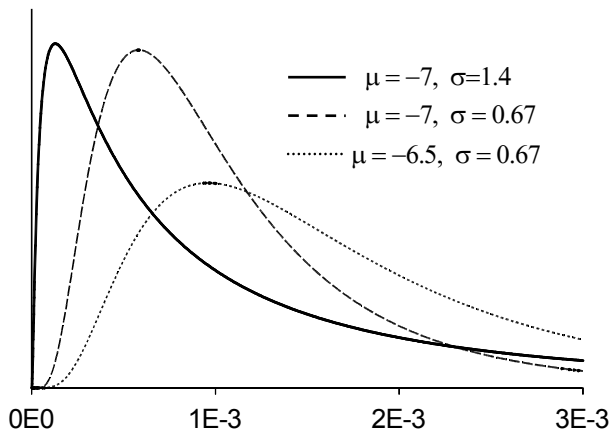
Figure B.10 Three lognormal densities.

The two distributions with $\sigma = 0.67$ and different values of μ have essentially the same shape, but with different scales. The larger μ corresponds to spreading the distribution out more from zero. The distribution with $\sigma = 1.4$, and therefore EF = 10, has a very skewed distribution.

To calculate probabilities for a lognormal(μ, σ^2) random variable, Y, the tables for the standard normal can be used. Specifically, for any number b,

$$\Pr[Y \le b] = \Pr[\ln(Y) \le \ln(b)] = \Pr[X \le \ln(b)]$$

$$= \Phi\left[\frac{(\ln(b) - \mu)}{\sigma}\right]$$

where $X = \ln(Y)$ is normal(μ, σ^2).

B.7.4 Exponential Distribution

The **exponential distribution** is widely used for modeling time to failure and is inherently associated with the Poisson process (Martz and Waller 1991). For a Poisson random variable X defining the number of failures in a time interval t and for a random variable T defining the time to failure, it can be shown that T has the exponential PDF

$$f(t) = \lambda e^{-\lambda t} \; ,$$

0   GC00 0433 1
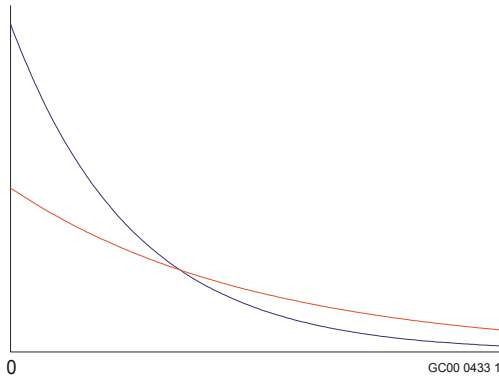
Figure B.11 Two exponential densities.

for $t > 0$. Thus, the time to first failure and the times between successive failures follow an exponential distribution and the number of failures in a fixed time interval follows a Poisson distribution.

Figure B.11 shows two exponential densities, for two values of λ. The intercept (height of the curve when $t = 0$) equals λ. Thus, the figure shows that the distribution is more concentrated near zero if λ is large. This agrees with the interpretation of λ as a frequency of failures and t as time to first failure.

The exponential distribution parameter, λ, corresponds to the λt parameterization of the Poisson PDF in Equation B.4. and is referred to as the **failure rate** if the component or system is repaired and restarted immediately after each failure. It is called the **hazard rate** if the component or system can only fail once and cannot be repaired. Section 4.6.2 discusses modeling duration times with different distributions and defines the hazard rate as $h(t) = \dfrac{f(t)}{[1 - F(t)]}$. For the exponential distribution, the hazard rate is constant, λ. The CDF of the exponential distribution is

$$F(t) = 1 - e^{-\lambda t} \; .$$

The exponential distribution with parameter λ is denoted exponential(λ). The mean and variance of an exponential(λ) distribution are

B-23

$$E(T) = \frac{1}{\lambda}$$

and

$$Var(T) = \frac{1}{\lambda^2} \; .$$

The relationship of the exponential distribution to the Poisson process can be seen by observing that the probability of no failures before time t can be viewed in two ways. First, the number of failures, X, can be counted. The probability that the count is equal to 0 is given by Equation B.4 as

$$\Pr(X = 0) = e^{-\lambda t} \frac{(\lambda t)^0}{0!} = e^{-\lambda t} \; .$$

Alternatively, the probability that first failure time, T, is greater than t is

$$\Pr(T > t) = 1 - \Pr(T \leq t) = 1 - F(t) = 1 - \left[1 - e^{-\lambda t} \right] = e^{-\lambda t} \; .$$

Thus, the two approaches give the same expression for the probability of no failures before time t.

The assumptions of a Poisson process require a constant failure rate, λ, which can be interpreted to mean that the failure process has no memory (Martz and Waller 1991). Thus, if a device is still functioning at time t, it remains as good as new and its remaining life has the same exponential(λ) distribution. This constant failure rate corresponds to the flat part of the common "bathtub" failure curve (number of failures plotted against time) and does not pertain to initial (burn-in) failures and wear-out failures.

A different, sometimes useful, parameterization uses $\mu = 1/\lambda = E(T)$. For example, if T represents a time to failure, μ is called the mean time to failure. If T is the time to repair, or to fire suppression, or to some other event, the name for μ is the mean time to repair, or other appropriate name. The exponential(μ) distribution for T has density

$$f(t) = \left(\frac{1}{\mu} \right) \exp\left(-\frac{t}{\mu} \right) \; , \text{ for } t \geq 0$$

and CDF

$$F(t) = 1 - \exp\left(-\frac{t}{\mu} \right) \; , \text{ for } t \geq 0 \; .$$

The units of μ are the same as the units of t, minutes or hours or whatever the data have. The mean and variance are

$$E(T) = \mu \quad \text{and}$$

$$Var(T) = \mu^2 \; .$$

B.7.5 Weibull Distribution

The **Weibull distribution** is widely used in reliability and PRA and generalizes the exponential distribution to include nonconstant failure or hazard rates (Martz and Waller 1991). Different Weibull distributions have been successfully used to describe initial failures and wear-out failures. The Weibull distribution is appropriate when a system is composed of a number of components, and system failure is due to any one of the components failing. It, therefore, is commonly referred to as a distribution corresponding to failure of the weakest link.

For a random variable, *T*, that has a Weibull distribution, the PDF is

$$f(t) = \frac{\beta}{\alpha}\left(\frac{t-\theta}{\alpha}\right)^{\beta-1} \exp\left[-\left(\frac{t-\theta}{\alpha}\right)^{\beta}\right],$$

for $t \geq \theta \geq 0$ and parameters $\alpha > 0$ and $\beta > 0$. The parameter θ is a location parameter and corresponds to a period of guaranteed life that is not present in many applications (Martz and Waller 1991). Thus, θ is usually set to zero. The CDF for *T* is

$$F(t) = 1 - \exp\left[\left(\frac{t-\theta}{\alpha}\right)^{\beta}\right],$$

for $t \geq \theta$ and $\alpha > 0$ and $\beta > 0$.

The α parameter is a scale parameter that expands or contracts the density along the horizontal axis. The β parameter is a shape parameter that allows for a wide variety of distribution shapes (Martz and Waller 1991) for further discussion and examples]. When $\beta = 1$, the distribution reduces to the exponential distribution. Therefore, the Weibull family of distributions includes the exponential family of distributions as a special case.

A Weibull distribution with parameters α, β, and θ is referred to as Weibull(α, β, θ) and, when $\theta = 0$, Weibull(α,β). The mean and variance of the Weibull distribution are given by Martz and Waller (1991) as

$$\theta + \alpha\Gamma\left(1 + \frac{1}{\beta}\right)$$

and

$$\alpha^2\left[\Gamma\left(1 + \frac{2}{\beta}\right) - \Gamma^2\left(1 + \frac{1}{\beta}\right)\right].$$

Here, Γ is the gamma function, defined in Sec. B.7.6.

Figure B.12 shows four Weibull densities, all with the same scale parameter, α, and all with location parameter $\theta = 0$. The shape parameter, β, varies. When $\beta < 1$, the density becomes infinite at the origin. When $\beta = 1$, the distribution is identical to the exponential distribution. Surprisingly, the distribution is *not* asymptotically normal as β becomes large, although it is approximately normal when β is near 3.

Figure B.12 Four Weibull densities, all having $\theta = 0$ and all having the same α.

GC00 0433 2

$\beta = 0.5$
$\beta = 1$
$\beta = 2$
$\beta = 3$

B.7.6 Gamma and Chi-Squared Distributions

The **gamma distribution** is an extension of the exponential distribution and is sometimes used as a failure time model (Martz and Waller, 1991). It is also often used as a prior distribution in Bayesian estimation (Sections B.6.2 and B.7.4) of the failure rate parameter λ from Poisson(λt) or exponential(λ) data. The **chi-squared distribution** is a re-expression of a special case of the gamma distribution.

The gamma distribution arises in many ways. The distribution of the sum of independent exponential(λ) random variables is gamma, which forms the basis for a confidence interval for λ from exponential(λ) data. Because the sum of n independent exponentially distributed random variables has a gamma distribution, the gamma distribution is often used as the distribution of the time, or waiting time, to the nth event in a Poisson process. In addition, the chi-squared distribution is the distribution for a sum of squares of independent, identically distributed normal random variables, which forms the basis for a confidence interval for the variance of a normal distribution. The gamma distribution is also often used as a distribution for a positive random variable, similar to the lognormal and Weibull distributions. In PRA work, it is often used as a Bayesian distribution for an uncertain positive parameter.

Two parameterizations of the gamma distribution are common, with various letters used for the parameters. The parameterization given here is most useful for Bayesian updates, the primary use of the gamma distribution in this handbook. For a random variable, T, that has a gamma distribution, the PDF is

$$f(t) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} t^{\alpha-1} \exp(-t\beta) \ ,$$

for t, α, and $\beta > 0$. Here

$$\Gamma(\alpha) = \int_{0}^{\infty} x^{\alpha-1} e^{-x} dx$$

is the **gamma function** evaluated at α. If α is a positive integer, $\Gamma(\alpha) = (\alpha - 1)!$.

A gamma distribution with parameters α and β is referred to as gamma(α, β). The mean and variance of the gamma(α, β) random variable, T, are:

$$E(T) = \frac{\alpha}{\beta}$$

and

$$Var(T) = \frac{\alpha}{\beta^2} \ .$$

The parameters α and β are referred to as the shape and scale parameters. The shape parameter α allows the density to have many forms. If α is near zero, the distribution is highly skewed. For $\alpha = 1$, the gamma distribution reduces to an exponential(β^{-1}) distribution. Also, the gamma($\alpha = n/2$, $\beta = \frac{1}{2}$) distribution is known as the **chi-squared distribution** with n degrees of freedom, denoted $\chi^2(n)$. The PDF for the $\chi^2(n)$ distribution is found by substituting these values into the above formula for the gamma PDF It also can be found in many statistics texts (e.g., Hogg and Craig 1995, Chapter 4).

In addition, if T has a gamma(α, β) distribution, then $2\beta T$ has a $\chi^2(2\alpha)$ distribution, which forms the defining relationship between the two distributions. The gamma and chi-squared distributions can, therefore, be viewed as two ways of expressing one distribution. Since the chi-squared distribution usually is only allowed to have integer degrees of freedom, the gamma distribution can be thought of as an interpolation of the chi-squared distribution.

Percentiles of the chi-squared distribution are tabulated in many statistic books. These tables can be used as follows to find the percentiles of any gamma distribution. The 100×p percentile of a gamma(α, β) distribution is $\chi^2_p(2\alpha)/(2\beta)$, where $\chi^2_p(2\alpha)$ denotes the 100×p percentile of the chi-squared distribution with 2α degrees of freedom.
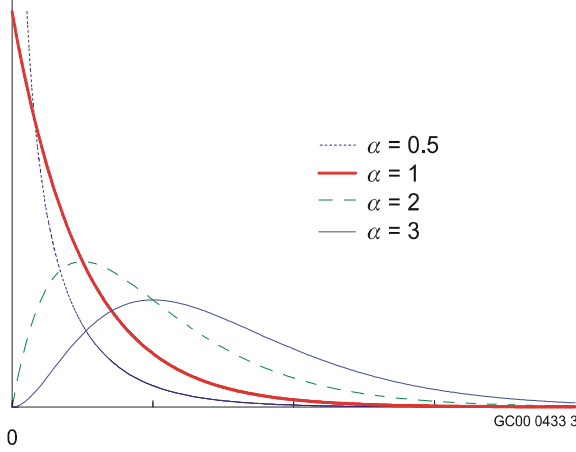
Figure B.13 shows gamma densities with four shape parameters, α. When $\alpha < 1$, the density becomes infinite at 0. When $\alpha = 1$, the density is identical to an exponential density. When α is large, the distribution is approximately a normal distribution. Note that when $\alpha = 0.5$ and $\beta = 0$ (which is not a proper distribution), the distribution is the Jeffreys noninformative prior for a Poisson likelihood function.

As stated previously, the sum of exponential lifetimes or waiting times has a gamma distribution, with the shape parameter α equal to the number of exponential lifetimes. Thus, when α is large, the gamma distribution is approximately normal.

Figure B.13 Gamma densities with four shape parameters.

An alternative parameterization of the gamma distribution uses the scale parameter, say $\tau = \beta^{-1}$. If T has a gamma(α, τ) distribution, its PDF is

$$f(t) = \frac{1}{\tau^\alpha \Gamma(\alpha)} t^{\alpha-1} \exp\left(-\frac{t}{\tau}\right)$$

for t, α, and $\tau > 0$. The mean and variance of the gamma(α, τ) random variable, T, are:

$$E(T) = \alpha\tau \quad \text{and} \quad Var(T) = \alpha\tau^2 .$$

B.7.7 Inverted Gamma and Inverted Chi-Squared Distributions

The **inverted gamma distribution** is often used as a prior distribution for Bayesian estimation of the time to failure of an exponential(λ) distribution (Martz and Waller 1991). It is also used as a prior and posterior distribution for σ^2 when the data have a normal distribution with variance σ^2 (Box and Tiao 1973, Lee 1997).

For a gamma(α, β) random variable, T, $W = 1/T$ has an inverted gamma distribution with PDF

$$f(w) = \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{w}\right)^{\alpha+1} \exp\left(-\frac{\beta}{w}\right) ,$$

for w, α, and $\beta > 0$. The parameters here are the same as for the gamma distribution. For example, if T has units of time then w and β both have units 1/time. A comparison of this density with the gamma

B-27

density shows that this density has an extra w^2 in the denominator, for reasons explained in Section B.4.7.

The parameters of the inverted gamma distribution are α and β and this distribution is denoted inverted gamma(α, β). Similar to the gamma(α, β) distribution, α is the shape parameter and β is the scale parameter. The distribution can also be parameterized in terms of $\tau = \beta^{-1}$.

The mean and variance of an inverted gamma(α, β) random variable, W, are

$$E(W) = \frac{\beta}{\alpha - 1} \quad , \quad \alpha > 1,$$

and

$$Var(W) = \frac{\beta^2}{(\alpha - 1)^2 (\alpha - 2)} \quad , \quad \alpha > 2.$$

Note that, for $\alpha \leq 1$, the mean and higher moments do not exist and, for $1 < \alpha \leq 2$, the mean exists but the variance does not exist (Martz and Waller, 1991).

GC00 0433 6
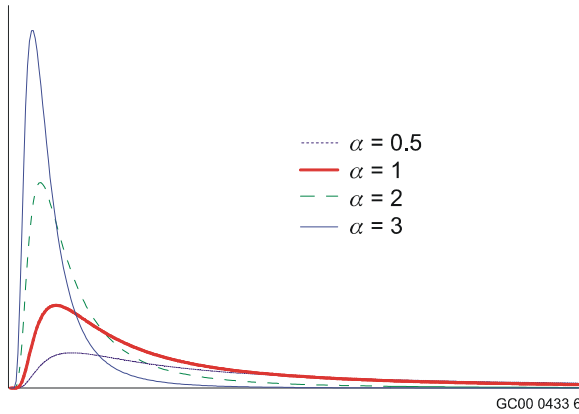
Figure B.14 shows four inverted gamma distributions, all having the same scale parameter, β, and having various shape parameters, α.

In the special case with $\alpha = n/2$ and $\beta = \frac{1}{2}$, the distribution is called the **inverted chi-squared** distribution with n degrees of freedom. Values from this distribution are sometimes denoted $\chi^{-2}(n)$. This form of the distribution is often used in connection with a prior for σ^2 when the data are normally distributed.

Figure B.14 Four inverted gamma densities, having the same scale parameter, β, and various

B.7.8 Beta Distribution

Many continuous quantitative phenomena take on values that are bounded by known numbers a and b. Examples are percentages, proportions, ratios, and distance to failure points on items under stress. The **beta distribution** is a versatile family of distributions that is useful for modeling phenomena that can range from 0 to 1 and, through a transformation, from a to b.

The beta distribution family includes the uniform distribution and density shapes that range from decreasing to uni-modal right-skewed to symmetric to U-shaped to uni-modal left-skewed to increasing (Martz and Waller 1991). It can serve as a model for a reliability variable that represents the probability that a system or component lasts at least t units of time. The beta distribution is also widely used in Bayesian estimation and reliability analysis as a prior distribution for the binomial distribution parameter p that represents a reliability or failure probability.

The PDF of a beta random variable, Y, is

$$f(y) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} y^{\alpha-1}(1-y)^{\beta-1} \quad,$$

for $0 \le y \le 1$, with the parameters α, $\beta > 0$, and is denoted beta(α, β). The gamma functions at the front of the PDF form a normalizing constant so that the density integrates to 1.

The mean and variance of the beta(α, β) random variable, Y, are

$$E(Y) = \frac{\alpha}{\alpha + \beta}$$

and

$$Var(Y) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \quad.$$

Various beta distributions are shown in Figures B.15 and B.16. Figure B.15 shows beta densities with $\alpha = \beta$, and therefore with mean 0.5. When $\alpha < 1$, the density becomes infinite at 0.0, and when $\beta < 1$, the density becomes infinite at 1.0. When $\alpha = \beta = 1$, the density is uniform. When $\alpha = \beta = 0.5$, the density is U shaped and is the Jeffreys noninformative prior for a binomial likelihood function. When α and β are large, the density is approximately normal.

Figure B.16 shows densities with mean 0.1. Again, when $\alpha < 1$, the density becomes infinite at 0.0, and when $\alpha > 1$, the density is zero at 0.0. As the parameters α and β become large, the density approaches a normal distribution.
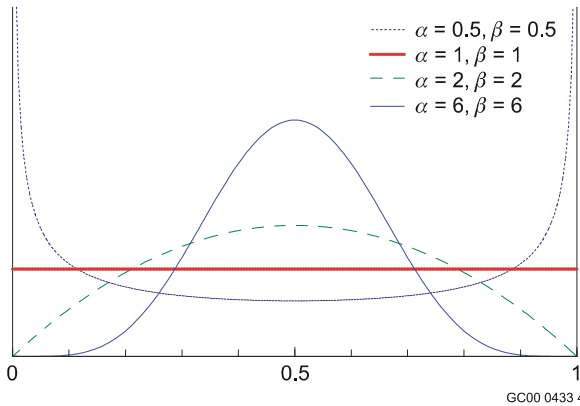
Figure B.15 Beta distributions with mean = 0.5.
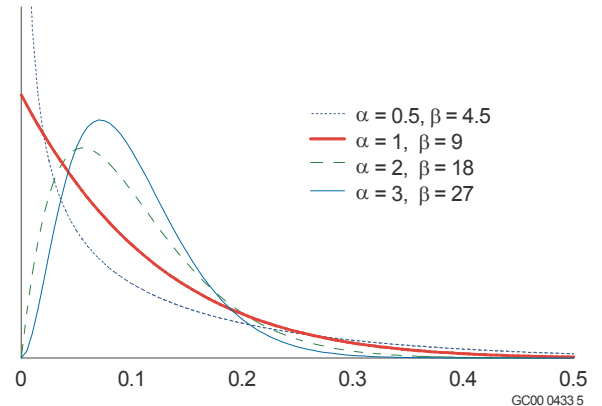
Figure B.16 Four beta distributions with mean 0.1.

Another parameterization of the beta distribution uses the parameters $x_0 = \alpha$ and $n_0 = \alpha + \beta$. This parameterization is used by Martz and Waller (1991) because it simplifies Bayes formulas and Bayesian estimation. The PDF of a beta(x_0, n_0) is

$$f(y) = \frac{\Gamma(n_0)}{\Gamma(x_0)\Gamma(n_0 - x_0)} y^{x_0-1}(1-y)^{n_0 - x_0 - 1} \quad,$$

for $0 \le y \le 1$, with the parameters x_0 and n_0 satisfying

$n_0 > x_0 > 0$.

The mean and variance of the beta(x_0, n_0) random variable, Y, are

$$E(Y) = \frac{x_0}{n_0}$$

and

$$Var(Y) = \frac{x_0(n_0 - x_0)}{n_0^2(n_0 + 1)} \ .$$

Percentiles of the beta distribution occur in the formula for a Bayes credible interval for p when a conjugate prior is used. Many software packages, including some commonly used spreadsheets, can calculate these percentiles.

B.7.9 Logistic-Normal Distribution

This distribution is used for Bayesian inference, especially as a prior for the binomial parameter p when p could plausibly be fairly large. X has a logistic-normal distribution if $\ln[X/(1-X)]$ is normally distributed with some mean μ and variance σ^2. The function $\ln[X/(1-X)]$ may appear strange, but it is common enough in some areas of application to have a name, the **logit** function. Therefore, the above statements could be rewritten to say that X has a logistic-normal distribution if logit(X) is normally distributed.

Properties of the logistic-normal distribution are summarized here.

- Let $y = \ln[x/(1-x)]$. Then $x = e^y/(1+e^y)$. This implies that x must be between 0 and 1.

- As x increases from 0 to 1, $y = \ln[x/(1-x)]$ increases monotonically from $-\infty$ to $+\infty$. Thus, y can be generated from a normal distribution with no problem of going outside the possible range.

- The monotonic relation between x and y means that the percentiles match. For example, the 95th percentile of Y is $\mu + 1.645\sigma$. Denote this by $y_{0.95}$. Therefore, the 95th percentile of X is

$$x_{0.95} = \exp\left[\frac{(x_{0.95})}{[1 + \exp(x_{0.95})]}\right] \ .$$

  Alternatively, this can be written as

$$y_{0.95} = \ln\left[\frac{x_{0.95}}{(1 - x_{0.95})}\right] \ .$$

- If X is close to 0 with high probability, so that $X/(1-X)$ is close to X with high probability, then the logistic-normal and lognormal distributions are nearly the same.

The third bullet shows how to find the percentiles of a logistic-normal distribution. Unfortunately there is no equally easy way to find the moments, such as the mean or variance. Moments must be found using numerical integration.

Figure B.17 shows several logistic normal distributions that all have median 0.5. These correspond to a normally distributed y with mean $\mu = 0$ and with various values of σ. Figure B.18 shows several logistic normal distributions that all have median 0.1. These correspond to a normally distributed y with mean $\mu = -2.2 = \ln[0.1/(1 - 0.1)]$.

Note the general similarities to the beta distributions in Figures B.15 and B.16. Note also the differences: Logistic-normal distributions are characterized most easily by percentiles, whereas beta distributions are characterized most easily by moments. Also, the beta densities can be J-shaped or U-shaped, but the logistic-normal densities always drop to zero at the ends of the range.
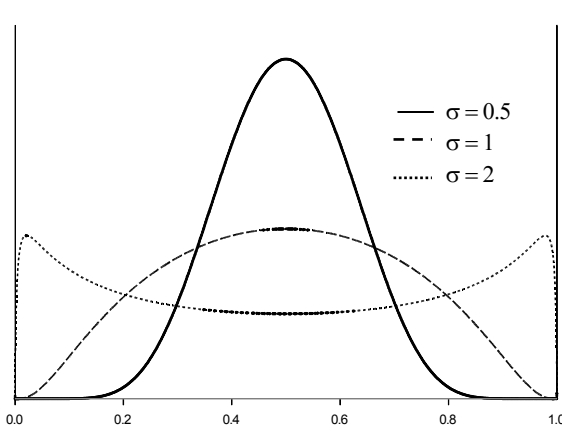
Figure B.17 Three logistic-normal densities with median = 0.5.
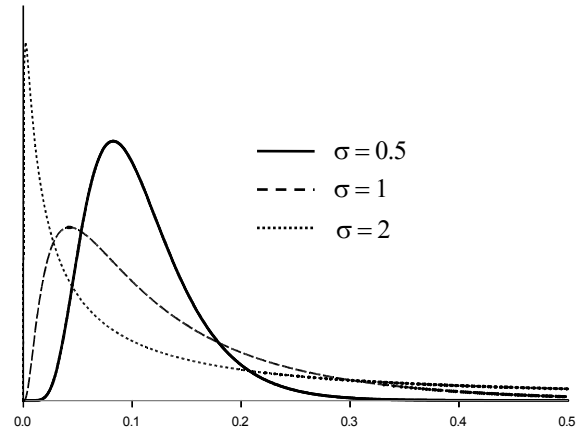
Figure B.18 Three logistic-normal densities with median = 0.1.

B.7.10 Student's t Distribution

If Z has a standard normal distribution, X has a chi-squared distribution with d degrees of freedom, and Z and X are statistically independent, then

$$T = \frac{Z}{\sqrt{\dfrac{X}{d}}}$$

has a Student's t distribution with d degrees of freedom. Therefore, T has a distribution that is symmetrical about 0, and it can take values in the entire real line. If d is large, the denominator is close to 1 with high probability, and T has approximately a standard normal distribution. If d is smaller, the denominator adds extra variability, and the extreme percentiles of T are farther out than are the corresponding normal percentiles.

The PDF and first two moments of T are given here. (many standard texts, such as DeGroot 1975.) The PDF is

B-31

$$f(t) = \frac{\Gamma\left[\dfrac{(d+1)}{2}\right]}{(d\pi)^{1/2}\Gamma\left(\dfrac{d}{2}\right)}\left[1+\left(\frac{t^2}{2}\right)\right]^{-\frac{(d+1)}{2}} .$$

If $d > 1$ the mean is 0. If $d > 2$ the variance is $d/(d-2)$. If $d \le 2$ the variance does not exist. If $d = 1$, even the mean does not exist; in this case the distribution is called a **Cauchy distribution**.

B.8 Random Samples

When sampling from a distribution (or population), it is usually assumed that the n observations are taken at random, in the following sense. It is assumed that the n random variables X_1, X_2, ..., X_n are independent. That is, the sample X_1, X_2, ..., X_n taken from a distribution $f(x)$ has the joint PDF h satisfying

$h(x_1, x_2, ..., x_n) = f(x_1) \cdot f(x_2) \cdot \ldots \cdot f(x_n)$.

This follows the definition of independent random variables given in Section B.4.5. A sample taken in this way is called a **random sample**. (As elsewhere in this handbook, upper case letters denote random variables and lower case letters denote particular values, number.)

The random variables X_1, X_2, ..., X_n forming such a random sample are referred to as being independent and identically distributed. If n is large enough, the sampled values will represent the distribution well enough to permit inference about the true distribution.

B.9 Sample Moments

Mathematical expectation and moments provide characteristics of distributions of random variables. These ideas can also be used with observations from a random sample from a distribution to provide **estimates** of the parameters that characterize that distribution.

A **statistic** is a function of one or more random variables that does not depend on any unknown parameters. A function of random variables that can be computed from the collected data sample is thus a statistic. Note that a function of random variables is also a random variable that has its own probability distribution and associated characteristics.

If X_1, X_2, ..., X_n denote a random sample of size n from a distribution $f(x)$, the statistic

$$\overline{X} = \sum_{i=1}^{n} \frac{X_i}{n}$$

is the **mean of the random sample**, or the **sample mean** and the statistic

$$S^2 = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}{n-1}$$

B-32

is the **variance of the random sample**. Note that $n - 1$ is used as the denominator in the S^2 statistic to make the statistic an *unbiased* estimator of the population variance, σ^2.

Similarly, the statistics defined by

$$m_r = \sum_{i=1}^{n} \frac{X_i^r}{n} \quad ,$$

for $r = 1, 2, ...,$ are called the **sample moments**.

One of the common uses of statistics is estimating the unknown parameters of the distribution from which the sample was generated. The sample mean, or average, $\overline{X}$, is used to estimate the distribution mean, or population mean, μ, the sample variance, S^2, is used to estimate the population variance, σ^2, and so forth.

B.10 Statistical Inference

Since values of the parameters of a distribution are rarely known, the distribution of a random variable is rarely completely known. However, with some assumptions and information based on a random sample of observations from the distribution or population, values of the unknown parameters can often be estimated. Probabilities can then be calculated from the corresponding distribution using these parameter estimates.

Statistical inference is the area of statistics concerned with using sample data to answer questions and make statements about the distribution of a random variable from which the sample data were obtained. Parameter **estimators** are functions of sample data that are used to estimate the distribution parameters. Statements about parameter values are inferred from the specific sample to the general distribution of the random variable or population. This inference cannot be perfect; all inference techniques involve uncertainty. Understanding the performance properties of various estimators has received much attention in the statistics field.

For the purposes of this handbook, statistical inference procedures can be classified as follows:

- parameter estimation

  o estimation by a point value

  o estimation by an interval

- hypothesis testing

  o tests concerning parameter values

  o goodness-of-fit tests and other model-validation tests.

Parametric statistical inference assumes that the sample data come from a particular, specified family of distributions, with only the parameter values unknown. However, not all statistical inference is based on parametric families. In many cases, in addition to not knowing the distribution parameter values, the form of the parametric family of distributions is unknown. **Distribution-free**, also called **nonparametric**, techniques are applicable no matter what form the distribution may have. **Goodness-of-fit tests** are an important type of nonparametric tests that can be used to test whether a data set follows a hypothesized distribution.

B.11 Distribution Summary

Probability Density Function	Mean	Variance[†]	Excel Formulation	SAPHIRE Formulation
Beta				
$\pi(p) = \dfrac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha,\beta)}$ $\text{where } B(\alpha,\beta) = \dfrac{\Gamma(\beta)\Gamma(\alpha)}{\Gamma(\alpha+\beta)}$	$\dfrac{\alpha}{\beta+\alpha}$	$\dfrac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$	Density Not available Cumulative BETADIST(p, α, β) Inverse cumulative BETAINV(%, α, β)	1st parameter Mean 2nd parameter β
Gamma				
$\pi(\lambda) = \dfrac{\beta^{\alpha}\lambda^{\alpha-1}e^{-\beta\lambda}}{\Gamma(\alpha)}$	$\dfrac{\alpha}{\beta}$	$\dfrac{\alpha}{\beta^2}$	Density Gammadist(x,α,1/β,false) Cumulative Gammadist (x, α, 1/β, true) Inverse cumulative Gammadist (%, α, 1/β)	1st parameter Mean 2nd parameter α (called "r")
Lognormal				
$\pi(\lambda) = \dfrac{1}{\sqrt{2\pi}\sigma\lambda}\exp\left[-\dfrac{(\ln\lambda-\mu)^2}{2\sigma^2}\right]$	$\exp(\mu+\dfrac{\sigma^2}{2})$	$(mean)^2\left[\exp(\sigma^2)-1\right]$		1st parameter Mean 2nd parameter Error factor
Uniform				
Same as the beta distribution with α=1 and β=1				

[†] Recall that the variance is equal to the standard deviation squared, or Var(x) = σ^2.

B.12 REFERENCES FOR APPENDIX B

Atwood, C. L., J. L.LaChance, H. F. Martz, D. J. Anderson, M. Englehardt, D. Whitehead, T. Wheeler, 2003, *NUREG/CR-6823, Handbook of Parameter Estimation for Probabilistic Risk Assessment.*

Ascher, H. and H. Feingold, 1984, *Repairable Systems Reliability – Modeling, Inference, Misconceptions and Their Causes*, Marcel Dekker, Inc., New York.

Bain, L. J., and M. Engelhardt, 1992, *Introduction to Probability and Mathematical Statistics, 2nd Ed.*, PWS, Boston.

Çinlar, E., 1975, *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, NJ.

DeGroot, Morris H., 1975, *Probability and Statistics*, Addison-Wesley, Reading, MA.

Derman, C., L. J. Gleser and I. Olkin, 1973, *A Guide to Probability Theory and Application*. Holt, Rinehart and Winston, NY.

Hogg, R. V. and A. T. Craig, 1995, *Introduction to Mathematical Statistics*. Macmillan, NY.

Lee, Peter M., 1997, *Bayesian Statistics: An Introduction, Second Edition*, Arnold, a member of the Hodder Headline Group, London.

Martz, H. F. and R. A. Waller, 1991, *Bayesian Reliability Analysis*. R. E. Krieger Publishing Co., Malabar, FL.

Meyer, P. L., 1970, *Introductory Probability and Statistical Applications*. Addison-Wesley Pub. Co., Reading, MA.

Pfeiffer, P. E. and D. A. Schum, 1973, *Introduction to Applied Probability*. Academic Press, NY.

C. Applicable Tools

C.1 WinBUGS and OpenBUGS

BUGS refers to **B**ayesian inference **U**sing **G**ibbs **S**ampling. WinBUGS is freely available software for implementing Markov chain Monte Carlo (MCMC) sampling. It is available in a Windows version from the BUGS Project website http://www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml which also has links to the open source code (OpenBUGS) directly available at http://mathstat.helsinki.fi/openbugs/Home.html . Both programs are commonly referred to as WinBUGS or just BUGS.

WinBUGS is commonly used independently but can be called from other programs through shell commands or the open-source statistical program R through the R2WinBUGS or BRUGS packages for R. For more information on R, see the R Project homepage, www.r-project.org.

The OpenBUGS user manual (Spiegelhalter, et al, 2007) comes with the program and is the basis for much of this appendix.

C.1.1 Distributions Supported in OpenBUGS

Over 20 distributions are supported in OpenBUGS. Discrete and continuous univariate and multivariate distributions are supported. Common distributions used in PRA that are supported include:

- Binomial: dbin(p,n)

- Poisson: dpois(mu)

    o Users will often have mu = λt

- Exponential: dexp(lambda)

- Weibull: dweib(v,lambda)

- Gamma: dgamma(r,mu)

- Uniform: dunif(a,b)

- Beta: dbeta(a,b)

- Lognormal: dlnorm(mu,tau)

    o $tau = \dfrac{1}{\sigma^2}$

$$\circ \quad \sigma = \frac{\ln(EF)}{1.645}$$

It is also possible to analyze user-defined distributions in WinBUGS. See the OpenBUGS User Manual (Spiegelhalter, et al, 2007) for information on how to do this.

C.1.2 WinBUGS Script

WinBUGS is a scripting language with a menu-driven interface (Thomas, 2006). There are three parts to a WinBUGS script: the model description, data section, and initial values. A sample script is presented in Figure C.1.

```
Script to update failure rate

model   {                    #Model defined between {} symbols
   events ~ dpois(mu)        #Poisson dist. for number of events
   mu <- lambda*time         #Parameter of Poisson distribution
   lambda ~ dgamma(1.6,44)        #Prior distribution for lambda
        }
Data                         #Observed data
list(events=1,time=44)
```

The model description includes the likelihood function, prior distribution, and any derived quantities (e.g., system reliability).

The data can be written within the WinBUGS script or input from a separate text file. The initial values can be written within the WinBUGS script from a separate text file.

Figure C.1: Sample WinBUGS Script

The use of the # character is for comments. It is highly encouraged to comment scripts.

C.1.3 Demonstration of WinBUGS via an Example Analysis

Conjugate Prior Example

Assume the frequency of an event (lambda) has a gamma(1.6, 44 yr) prior distribution. The likelihood function is a Poisson distribution with observed data consisting of 1 event in a 44-year period. The posterior distribution is gamma(1.6+1, 44+44 yr) and the posterior mean of lambda is 2.6/88 yr = 0.03 per year.

The script in Figure C.1 is used to update the prior distribution for lambda with the observed data. This can be written either in the WinBUGS new project screen or cut and pasted from a text editor.

The **first** step is to check the model's syntax by highlighting the word "model" and selecting "Model – Specification" from the toolbar as in Figure C.2.

The Specification Tool will appear on the screen, after which the "Check Model" button should be selected. A status message will be displayed at the bottom left of the WinBUGS palette. If all is well, the message is returned, "model is syntactically correct".

Figure C.2: Checking the WinBUGS model for syntax.

Leave the Specification Tool on the screen (Figure C.3) and double-click (highlight) the word "list" in the data portion of the script, then select "load data". A status message will be displayed at the bottom left of the WinBUGS palette that states "data loaded". Next, select "compile" and WinBUGS reports "model compiled". Note that models using multiple chains can be run. The number of chains in this example is one.

The last task to do with the Specification Tool is to load the initial values. For the conjugate prior example, the model is such a simple one that we can let WinBUGS generate initial values. Select "gen inits" to have WinBUGS generate the initial values and a message of "initial values generated, model initialized" should appear in the status bar.
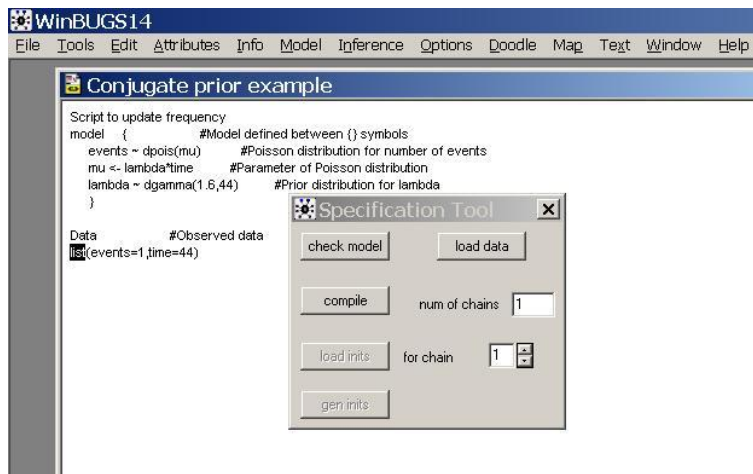
Figure C.3: WinBUGS Specification Tool.

The next step in setting up the analysis is to select the variables to monitor. For the conjugate prior example, we are interested in the frequency, or rate of occurrence (lambda). Close the Specification Tool and select "Inference" and then "Samples" from the toolbar. The Sample Monitor Tool (Figure C.4) will pop up. Type the variable name "lambda" in the "node" box. Enter 1001 in the "beg" box as the iteration at which to start monitoring lambda. This will allow this particular model enough iterations to converge. A discussion on convergence will follow. Select the default of 10,000,000 as the "end" value. Save

Figure C.4: WinBUGS Sample Monitor Tool.

the setting by clicking the "set" button. Multiple variables and chains can be monitored, although we will only monitor lambda for this example.

Close the Sample Monitor Tool and select "Model" and "Update" from the toolbar. The Update Tool appears on the screen (Figure C.5). Enter in the number of updates to perform, in this case 10,000 is the quantity desired. The refresh rate will control the display in the iteration box during an update. A lower number for the refresh rate will be informative, but will also slow the sampling time. Select "update". WinBUGS will report that the "model is updating" in the status bar and the iteration window will display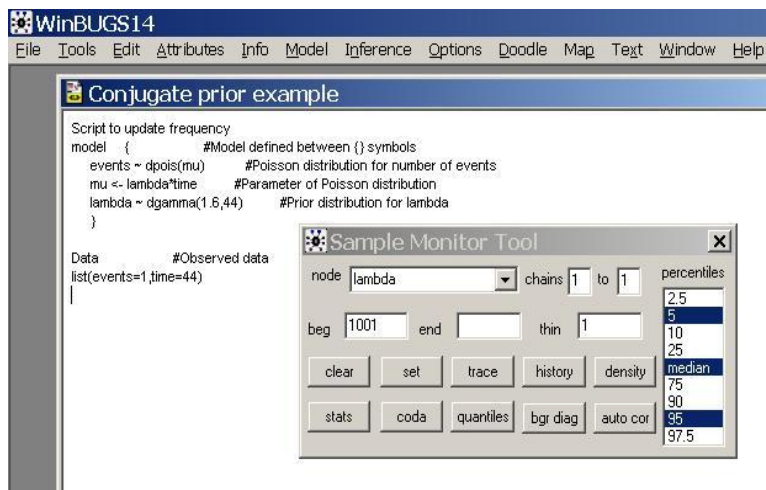 the iterations by increments specified in the refresh rate. When the model update has completed a message will appear in the status bar that indicates "updates took X s".
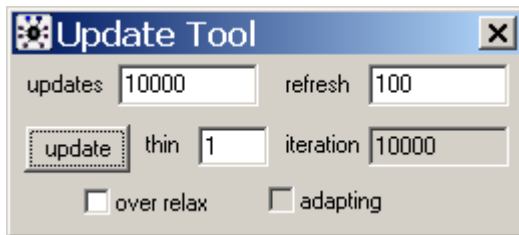
Figure C.5: WinBUGS Update Tool.

Close the Update Tool when the sampling is complete and re-open the Sample Monitor Tool by selecting "Inference → Samples" from the toolbar. Select "lambda" from the drop-down list of monitored nodes. Highlight the percentiles 5, median (50%), and 95 by holding down the Ctrl key while selecting with the left mouse button. To display a graph of the posterior density (Figure C.6) select "density".
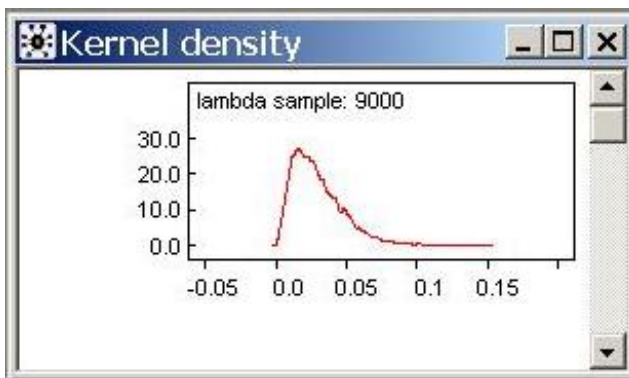
Figure C.6: Posterior distribution density.

Next select the "stats" button to get the posterior mean and percentiles selected. These results shown in Figure C.7 can be compared to hand/spreadsheet calculation results of:

- Posterior mean = 0.03/yr

- 5th percentile = 0.007/yr

- 95th percentile = 0.06/yr

Note that the results, both the density plots and the statistics, may be selected and copied (via the CTRL+C key combination in Windows) for pasting into other programs.

C.1.4 Monitoring Convergence in WinBUGS

node	mean	sd	MC error	5.0%	median	95.0%	start	sample
lambda	0.02962	0.01835	1.916E-4	0.007385	0.0259	0.06418	1001	9000

Figure C.7: Posterior statistics for lambda.

WinBUGS uses Markov chain Monte Carlo (MCMC) sampling to generate values directly from the target posterior distribution. However, it takes some time for the sampling to converge to the posterior distribution; any values sampled prior to convergence should not be used to estimate parameter values. For simple problems involving one parameter, such as p in the binomial distribution or lambda in the Poisson distribution, 1,000 iterations will be more than sufficient for convergence. In more complicated problems, which usually involve inference for more than one

parameter, this may not be the case, and the user will have to check for convergence. This section discusses practical checks that the user can do, using features built into WinBUGS.

Running Multiple Chains

For problems with more than one parameter, at least two chains should be run, with starting values that are dispersed around the estimated node of the posterior distribution. Usually the analysis will not be very sensitive to the initial values selected for the chains, but this is not always the case. An expert analyst should be consulted in such cases, which arise especially often when modeling population variability, as there are sophisticated approaches that can be used to derive initial values.

> **After user-specified initial values are loaded, it may still be necessary to have WinBUGS generate remaining initial values.**

After the model has been compiled (remember to specify the number of chains before compiling the model) and initial values have been loaded, specify the nodes to be monitored. All parameters should be monitored and convergence should be checked for each of these monitored nodes. Now run 1,000 samples and select History from the Inference menu to generate a trace of the first 1,000 samples for each monitored node. A plot like the one shown below, in which the two chains are well mixed, is indicative of convergence.
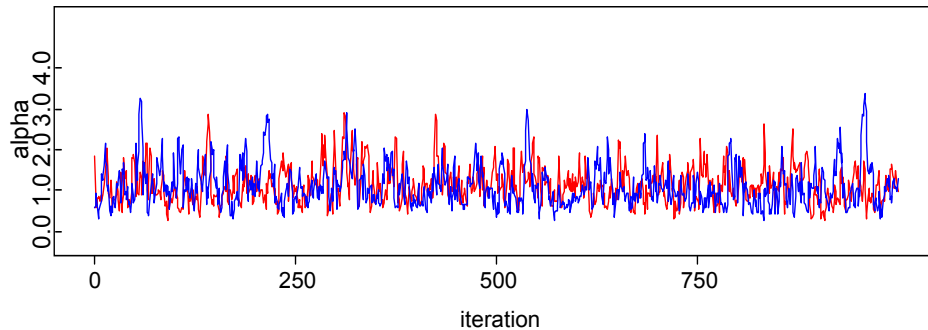

Figure C.8: History plot showing two well-mixed chains, indicative of convergence.

In contrast, the figure below shows a case in which the chain are not well mixed, which indicates that more iterations must be run to reach convergence.
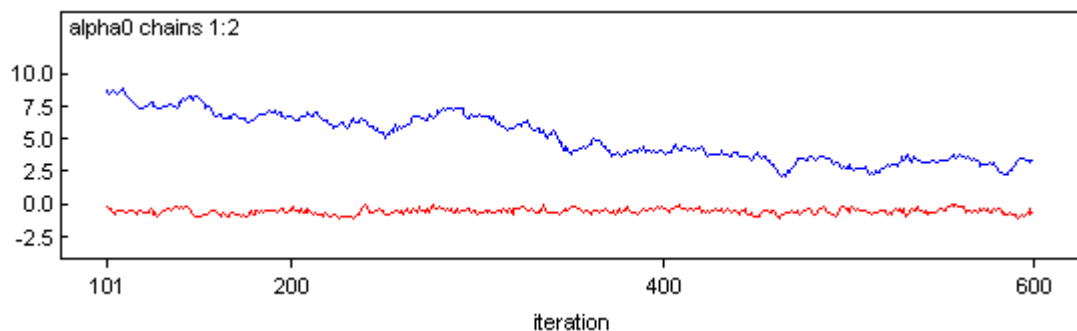

Figure C.9: History plot showing two poorly-mixed chains, indicative of failure to converge.

Brooks-Gelman-Rubin (BGR) Convergence Diagnostic

WinBUGS has a built-in convergence diagnostic that can be used in conjunction with the history plots shown above to help the user decide when burn-in samples have been taken. It requires multiple chains, and is based on an analysis of the variance within- and between-chains. If the chains have converged, all chains should have approximately the same within-chain variance, and this should be approximately equal to the between-chain variance estimate. The BGR diagnostic in WinBUGS looks at a ratio of these estimates, which is normalized to equal one when the chains have converged.

To implement the BGR diagnostic, run at least two chains, as described above, then select "bgr diag" from the Inference menu. The resulting plot will have three curves. The estimate of the within-chain variance is shown in blue, the between-chain estimate is in green, and the BGR ratio is shown in red. The BGR ratio is expected to start out greater than one. The heuristic is that this ratio should be less than about 1.2 for convergence. However, the between-chain and within-chain estimates should be stable. Double-clicking with the left mouse button on the BGR graph, followed by <Ctrl>-left click will bring up a table of the values over the history.

The first graph below shows a typical BGR plot for a problem that has converged. The second shows a plot where convergence may have been reached just at the end of the 1,000 samples, but more burn-in iterations should be taken to ensure convergence before estimating parameter values.
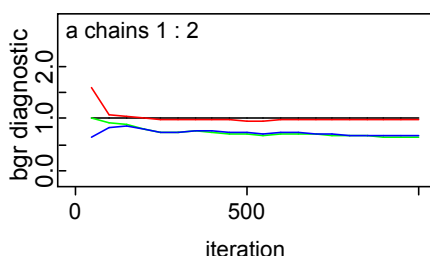

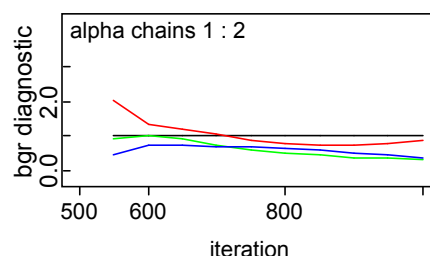**Figure C.10:** BGR plot illustrating convergence.

Figure C.11: BGR plot illustrating a questionable convergence.

C.2 References for WinBUGS/OpenBUGS

Spiegelhalter, D., A. Thomas, N. Best, D. Lunn, 2007, *OpenBUGS User Manual*, OpenBUGS project, ,http://www.mrc-bsu.cam.ac.uk/bugs

Thomas, A., 2006, *The BUGS Language*, R News, **vol 6/1**, 17-21.

D. Select Inference Equations

D.1 Introduction

This appendix provides additional details of the salient mathematics behind the Bayesian networks and WinBUGS scripts found in Sections 4.2.1, 4.2.2, and 4.2.3 of the guidebook. The network diagrams and scripts in the guidebook are intended to be the primary tools to help an analyst think through a Bayesian inference problem. However, some analysts may gain additional benefit from seeing the supporting equations, at least in those cases where it is possible to write them down. Furthermore, some problems in Bayesian inference are amenable to solution by numerical tools other than the Markov chain Monte Carlo (MCMC) sampling employed by WinBUGS, and analysts may feel more comfortable using such tools. Note that the starting point for inference, Bayes' Theorem, was described via Equation 2-2:

$$\pi_1(\theta \mid x) = \frac{f(x \mid \theta)\pi(\theta)}{\int f(x \mid \theta)\pi(\theta)d\theta}. \tag{2-2}$$

To perform inference, we need to solve equation 2-2, which requires knowing the functional form of the likelihood [$f(x \mid \theta)$], the prior [$\pi(\theta)$], and the data x.

Below, we list the applicable Appendix D equations for the topical areas found in Sections 4.2.1, 4.2.2, and 4.2.3. In addition, we provide a cross-reference to the OpenBUGS script found in the respective sections.

Script	Section	Appendix D Equations	Topical Area
n/a	4.2.1	D-1	Binomial distribution
1	4.2.1.1	D-1, D-2, D-3	Binomial with conjugate prior
n/a	4.2.1.2	D-1, D-2, D-3, D-17, D-18	Binomial Jeffreys prior
2	4.2.1.3	D-1, D-9, D-10, D-11, D-12, D-13	Binomial non-conjugate prior
3	4.2.1.3	D-1, D-9, D-10, D-11, D-12, D-13	Binomial non-conjugate prior
n/a	4.2.2	D-4	Poisson distribution
4	4.2.2.1	D-4, D-5, D-6	Poisson conjugate prior
n/a	4.2.2.2	D-4, D-5, D-6, D-17, D-18	Poisson Jeffreys prior
5	4.2.2.3	D-4, D-9, D-14	Poisson non-conjugate prior
6	4.2.3.1	D-5, D-7, D-8	Exponential conjugate prior
n/a	4.2.3.2	D-5, D-17, D-18	Exponential Jeffreys prior
7	4.2.3.3	D-8, D-14	Exponential non-conjugate prior

D.2 Inference for a Single Parameter

This section describes inference for p in the binomial distribution, typically used for failure on demand; λ in the Poisson distribution, used for initiating events and operating failures; and λ in the exponential distribution, used for random durations, such as time to recover failed equipment. The section also includes inference for the multinomial distribution, typically used to model common-cause failure of redundant equipment. While not a single-parameter problem, it is a slight extension of the binomial model, so the conceptual similarity justifies its inclusion in this section.

D.2.1 Binomial Distribution with Conjugate Prior

The likelihood function is the binomial distribution, given by

$$f(x \mid p) = \binom{n}{x} p^x (1-p)^{n-x} \tag{D-1}$$

The conjugate prior is a beta distribution, with density function

$$\pi_0(p) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha, \beta)} \tag{D-2}$$

B(α, β) is the beta function, defined as follows for α, β > 0:

$$B(\alpha, \beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} dx \tag{D-3}$$

If x failures are observed in n demands, the posterior distribution is easily shown to be a beta distribution with parameters α + x and β + n – x. The mean and variance can be found in closed form, but credible intervals must be found numerically, using a spreadsheet tool, as described in the guidebook, or another routine.

D.2.2 Poisson Distribution with a Conjugate Prior

The likelihood function is the Poisson distribution, given by

$$f(x \mid \lambda) = \frac{(\lambda t)^x e^{-\lambda t}}{x!} \tag{D-4}$$

The conjugate prior is a gamma distribution, with density function

$$\pi_0(\lambda) = \frac{\beta^\alpha \lambda^{\alpha-1} e^{-\beta\lambda}}{\Gamma(\alpha)} \tag{D-5}$$

Γ(α) is the gamma function, defined for α > 0 by

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx \tag{D-6}$$

If x events are observed in time t, the posterior is a gamma distribution with parameters α + x and β + t. As above, the mean and variance can be written in closed form but credible intervals must be found numerically, using a spreadsheet tool, as described in the guidebook, or another routine.

D.2.3 Exponential Distribution with Conjugate Prior

This is essentially identical to Poisson inference, but with a likelihood function that changes as follows. We are now observing times to occurrence. We assume that each time, T_i, i = 1, 2, …, n, is independently and identically distributed as an exponential random variable with rate λ. Thus, the density function for each T_i can be written

$$f(t_i \mid \lambda) = \lambda e^{-\lambda t_i} \tag{D-7}$$

Because of the independence assumption, the joint distribution for the likelihood factors into n terms, allowing us to write the likelihood function as the product of n exponential density functions:

$$f(\tilde{t} \mid \lambda) = \prod_{i=1}^{n} f(t_i \mid \lambda) = \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} t_i\right) \tag{D-8}$$

The prior is again a gamma distribution, with density given by Equation (D-5). The posterior is a gamma distribution with parameters α + n and β + Σt_i. As above, the mean and variance can be written in closed form but credible intervals must be found numerically, using a spreadsheet tool, as described in the guidebook, or another routine.

D.2.4 Binomial Distribution with Nonconjugate Prior

The lognormal prior for p in the binomial distribution is parameterized in terms of the logarithmic mean, μ, and logarithmic precision, τ, defined as τ = $1/\sigma^2$. The density function in this parameterization is given by

$$\pi_o(p) = \sqrt{\frac{\tau}{2\pi}} \frac{1}{p} \exp\left[-\frac{\tau}{2}(\log(p) - \mu)^2\right] \tag{D-9}$$

The likelihood function is the binomial distribution, given by Equation (D-1) above. Therefore, the posterior distribution is given by

$$\pi_1(p \mid x) = \frac{f(x \mid p)\pi_o(p)}{\int_0^1 f(x \mid p)\pi_o(p)dp} \tag{D-10}$$

The denominator of this equation does not have an analytic solution and must be evaluated numerically. As will be true for all of the cases utilizing a nonconjugate prior, the posterior distribution does not have a closed form, and is not a member of any particular functional family (e.g., lognormal). However, it can be approximated with a particular functional distribution, if desired.

The posterior mean of p is given by

$$\int_0^1 p\pi_1(\,p\,|\,x\,)dp \tag{D-11}$$

Again, this equation does not have an analytic solution and must be evaluated numerically.

Percentiles can be found by numerically solving equations such as the following:

$$\int_0^{p_\alpha} \pi_1(\,p\,|\,x\,)dp = \alpha \tag{D-12}$$

The logistic-normal prior for p is also nonconjugate. The density is given by

$$\pi_o(\,p\,) = \sqrt{\frac{\tau}{2\pi}}\,\frac{1}{p(1-p)}\,exp\left[-\frac{\tau}{2}\left(log(\frac{p}{1-p})\right)^2\right] \tag{D-13}$$

where μ and τ are the mean and precision of the underlying normal distribution. The posterior mean and credible intervals are obtained by substituting the logistic-normal density into the above equations.

D.2.5 Poisson Distribution with Nonconjugate Prior

The unknown parameter is now λ, the intensity of the Poisson process. Simply replace p with λ in Equation (D-9) above for the lognormal prior. The likelihood function is the Poisson distribution, given by Equation (D-4). The posterior distribution is given by

$$\pi_1(\,\lambda\,|\,x\,) = \frac{f(\,x\,|\,\lambda\,)\pi_o(\,\lambda\,)}{\int_0^\infty f(\,x\,|\,\lambda\,)\pi_o(\,\lambda\,)d\lambda} \tag{D-14}$$

Again, this equation does not have an analytic solution and must be evaluated numerically. The posterior mean and credible intervals are found numerically as in Equation (D-12) above using this posterior distribution.

D.2.6 Exponential Distribution with Nonconjugate Prior

As mentioned above, this is essentially the same as Poisson inference, but with a likelihood function given by Equation (D-8). The posterior distribution is obtained by substituting this likelihood function for the Poisson likelihood function into Equation (D-14) above.

D.2.7 Multinomial Distribution with Conjugate Prior

The multinomial distribution generalizes the binomial distribution to the situation where there are more than two possible outcomes of a trial. The most common application of the multinomial distribution in PRA is for common-cause failure (CCF), where it serves as an aleatory model for

common-cause failure of a group of redundant components. Given that a failure occurs in the group, the possible outcomes are one component fails, two fail due to common cause, three fail due to common cause, etc., up to all of the components in the group fail due to common cause. This model can be parameterized in different ways, with the most commonly used being one in which the parameters represent the fraction of failures of the group that involve a specific number of components in the group. These parameters are denoted α_k, and thus this parameterization is referred to as the alpha-factor model for CCF(Mosleh, Common Cause Failures: An Analysis Methodology and Examples, 1991). Each α_k represents the fraction of failures that involve k components of the group. The prior distribution for α_k, which is now a vector of dimension equal to the component group size, is almost always assumed to be a Dirichlet distribution, which is conjugate to the multinomial likelihood. Like α, the parameter of the Dirichlet distribution, which we will denote as θ, is a vector of dimension equal to that of α.

In equation form, the multinomial likelihood is

$$f(x_1, x_2, \ldots, x_n \mid \widetilde{\alpha}) = \frac{(\sum_{i=1}^{n} x_i)!}{\prod_{i=1}^{n} x_i!} \prod_{i=1}^{n} \alpha_i^{x_i}$$

(D-15)

Note the constraint that the α_ks must sum to unity, which will introduce a correlation into the joint posterior distribution. The Dirichlet prior distribution for the vector of alpha-factors has density function

$$\pi_0(\widetilde{\alpha}) = \frac{\Gamma(\sum_{i=1}^{n} \theta_i)}{\prod_{i=1}^{n} \Gamma(\theta_i)} \prod_{i=1}^{n} \alpha_i^{\theta_i - 1}$$

(D-16)

The posterior distribution is also Dirichlet, with parameters $\theta_k + x_k$.

D.2.8 Noninformative Priors

The guidebook discusses only the Jeffreys prior, which is obtained by applying Jeffreys rule:

$$\pi_{NI}(\theta) \propto \sqrt{J(\theta)}$$

(D-17)

$J(\theta)$ depends upon the likelihood function:

$$J(\theta) = -E\left[\frac{d^2(\log(f(x \mid \theta)))}{d\theta^2}\right]$$

(D-18)

For the binomial distribution, this leads to a beta(0.5, 0.5) distribution as the Jeffreys prior. For the Poisson distribution, the Jeffreys prior is an improper distribution proportional to $\lambda^{-0.5}$, which can be thought of as a (conjugate) gamma(0.5, 0) distribution, yielding a gamma posterior distribution with parameters x + 0.5 and t if x events are observed in time t. For the exponential distribution, the Jeffreys prior is an improper distribution proportional to $1/\lambda$, which can be thought of as a (conjugate) gamma(0, 0) distribution for Bayesian updating. For the case of the multinomial likelihood, the noninformative prior is a Dirichlet distribution with all parameters equal to 1. Note that this is not the Jeffreys prior; the Jeffreys prior is not often used as a noninformative prior in

multiparameter inference problems. Instead, as illustrated in the guidebook, one usually uses independent diffuse priors.

D.3 References

1. Mosleh, A. (1991). Common Cause Failures: An Analysis Methodology and Examples. *Reliability Engineering and System Safety ,* **34**, 249-292.

E. Analysis Validation

E.1 Introduction

This appendix provides an example of the possible validation approach for the analysis examples described in the main body of the report. Here we compare our calculated answers (using OpenBUGS) with other independent analysis tools including Excel, RADS, and R.

The table below provides validation results for the first eight scripts in the report.

Script	Section	Model Type	Validation Method	Results OpenBUGS	Results Validation Model	Pass or Fail
1	4.2.1.1	Binomial likelihood with Beta conjugate prior	Spreadsheet calculations based on conjugate distribution properties	**p** mean = 1.70×10^{-5} 5th % = 4.99×10^{-6} 95th % = 3.50×10^{-5}	**p** mean = 1.71×10^{-5} 5th % = 4.98×10^{-6} 95th % = 3.52×10^{-5}	Pass
2	4.2.1.3	Binomial likelihood with Lognormal prior	RADS[1] calculator	**p** mean = 4.7×10^{-5} 5th % = 1.9×10^{-6} 95th % = 1.8×10^{-4}	**p** mean = 4.69×10^{-5} 5th % = 1.87×10^{-6} 95th % = 1.78×10^{-4}	Pass
3	4.2.1.3	Binomial likelihood with Logistic-Normal prior	RADS calculator	**p** mean = 4.8×10^{-5} 5th % = 1.9×10^{-6} 95th % = 1.8×10^{-4}	**p** mean = 4.69×10^{-5} 5th % = 1.87×10^{-6} 95th % = 1.78×10^{-4}	Pass
4	4.2.2.1	Poisson likelihood with Gamma conjugate prior	Spreadsheet calculations based on conjugate distribution properties	**lambda** mean = 4.3×10^{-6} 5th % = 5.6×10^{-7} 95th % = 1.1×10^{-5}	**lambda** mean = 4.33×10^{-6} 5th % = 5.63×10^{-7} 95th % = 1.10×10^{-5}	Pass

---

[1] RADS is the "Reliability and Availability Data System" developed for the NRC by the INL.

5	4.2.2.3	Poisson likelihood with Lognormal prior	RADS calculator	**lambda** mean = 1.6×10^{-6} 5th % = 3.8×10^{-8} 95th % = 6.5×10^{-6}	**lambda** mean = 1.8×10^{-6} 5th % = 3.56×10^{-8} 95th % = 7.04×10^{-6}	Pass
6	4.2.3.1	Exponential likelihood with Gamma conjugate prior	Spreadsheet calculations based on conjugate distribution properties	**lambda** mean = 6.5×10^{-7} 5th % = 3.4×10^{-7} 95th % = 1.1×10^{-6}	**lambda** mean = 6.54×10^{-7} 5th % = 3.35×10^{-7} 95th % = 1.06×10^{-6}	Pass
7	4.2.3.3	Exponential likelihood with Lognormal prior	RADS calculator (treat as 7 Poisson events in 12782180 hours)	**lambda** mean = 5.5×10^{-7} 5th % = 2.6×10^{-7} 95th % = 9.2×10^{-7}	**lambda** mean = 5.47×10^{-7} 5th % = 2.64×10^{-7} 95th % = 9.16×10^{-7}	Pass
8	4.2.4	Multinomial likelihood with Dirichlet prior	Mcmultinomdirichlet function in MCMCpack package in R	**Alpha-1** mean = 0.72 5th % = 0.58 95th % = 0.85 **Alpha-2** mean = 0.21 5th % = 0.10 95th % = 0.34 **Alpha-3** mean = 0.07 5th % = 0.01 95th % = 0.16 **Beta** mean = 0.28 5th % = 0.15 95th % = 0.42 **Gamma** mean = 0.25 5th % = 0.05 95th % = 0.52	**Alpha-1** mean = 0.72 5th % = 0.58 95th % = 0.85 **Alpha-2** mean = 0.21 5th % = 0.10 95th % = 0.34 **Alpha-3** mean = 0.07 5th % = 0.01 95th % = 0.16 **Beta** mean = 0.28 5th % = 0.15 95th % = 0.43 **Gamma** mean = 0.25 5th % = 0.05 95th % = 0.52	Pass