Lance Fiondella
University of Massachusetts Dartmouth

Introduction to Software Reliability Growth Models

Acknowledgement: The author gratefully acknowledges the support of the National Science Foundation under Grant Numbers #1526128 and 1749635.

Contents

List of Figures

List of Tables

Chapter 1
Parameter Estimation Methods

Parameter estimation [Min, 1991] is the process of finding numerical values of a model's parameters that best fit the data. This section discusses two widely used estimation techniques

- Least-squares estimation (LSE)
- Maximum likelihood estimation (MLE)

To estimate a model's parameters, first obtain the software fault detection time data, which consists of individual failure times $\mathbf{T} = \langle t_1, t_2, \ldots, t_n \rangle$, where n is the number of faults observed. Alternatively, failure count data is represented as $\langle \mathbf{T}, \mathbf{K} \rangle = ((t_1, k_1), (t_2, k_2), \ldots, (t_n, k_n))$, where t_i is the time at which the i^{th} interval ended and k_i is the number of faults detected in interval i.

1.1 Least-squares estimation (LSE)

LSE [Kleinbaum et al., 2008] determines the parameters that best fit a data set by minimizing the sum of squares of the vertical distances between the observed data points and the fitted curve. Given n tuples (t_i, y_i) for failure time or failure count data, LSE minimizes

$$\sum_{i=1}^{n} = (y_i - f(t_i|\boldsymbol{\Theta}))^2 \tag{1.1}$$

where y_i is the cumulative number of faults detected and t_i is the i^{th} failure time or time at the end of the i^{th} interval. The function $f(t_i)$ is the model possessing the vector of parameters $\boldsymbol{\Theta}$ to be estimated. Given a specific mean value function (MVF), $m(t)$, of a software reliability, LSE minimizes

$$LSE(\boldsymbol{\Theta}) = \sum_{i=1}^{n} (y_i - m(t_i;\boldsymbol{\Theta}))^2 \tag{1.2}$$

where $m(t_i;\boldsymbol{\Theta})$ the number of faults predicted to occur by time t_i, given specific numerical values for $\boldsymbol{\Theta}$.

1.2 Maximum likelihood estimation (MLE)

Maximum likelihood estimation [Elsayed, 2012, Leemis, 1995] is a procedure to identify numerical values of model parameters that best fit the observed failure data. Maximum likelihood estimation maximizes the likelihood function, also known as the joint distribution of the failure data. Commonly, the logarithm of the likelihood function is maximized because this simplifies the mathematics and the monotonicity of the logarithm ensures that the maximum of the log-likelihood function is equivalent to maximizing the likelihood function.

To obtain maximum likelihood estimates for a models parameters, first specify the likelihood function, which is the joint density function of the observed data

$$L(\boldsymbol{\Theta};t_1,\ldots,t_n) = f(t_1,\ldots,t_n|\boldsymbol{\Theta}) = \prod_{i=1}^{n} f(t_i|\boldsymbol{\Theta}) \tag{1.3}$$

where $f(t)$ is the probability density function.

Consider the motivating example where an individual flips a coin five time and observes three heads and two tails. Based on these experiments, what is the most likely value of observing a head? Assuming the coin flips are independent and identically distributed, the probability may be characterized by a binomial distribution. Specifically, we can use our experimental results to write the probability of three heads and two tails as the following function of p

$$L(p) = \binom{5}{3} p^3 (1-p)^2 \tag{1.4}$$

where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is a binomial coefficient, p denotes a head, and $(1-p)$ tails. Now consider a discrete set of values for $p \in \{\frac{0}{5}, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, \frac{5}{5}\}$. Table 1.1 shows the probability of observing three heads and two tails for each value of p.

Table 1.1: Probability of three heads and two tails.

p	Probability
$\frac{0}{5}$	$\binom{5}{3}\left(\frac{0}{5}\right)^3\left(\frac{5}{5}\right)^2 = 0$
$\frac{1}{5}$	$\binom{5}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^2 = \frac{32}{625}$
$\frac{2}{5}$	$\binom{5}{3}\left(\frac{2}{5}\right)^3\left(\frac{3}{4}\right)^2 = \frac{144}{625}$
$\frac{3}{5}$	$\binom{5}{3}\left(\frac{3}{5}\right)^3\left(\frac{2}{5}\right)^2 = \frac{216}{625}$
$\frac{4}{5}$	$\binom{5}{3}\left(\frac{4}{5}\right)^3\left(\frac{1}{5}\right)^2 = \frac{128}{625}$
$\frac{5}{5}$	$\binom{5}{3}\left(\frac{5}{5}\right)^3\left(\frac{0}{5}\right)^2 = \frac{0}{625}$

Figure 1.1 shows a plot of the likelihood function for values of $p \in (0,1)$ for the motivating example.

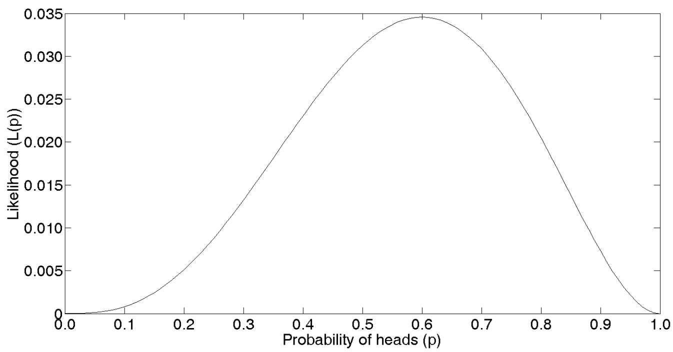

Fig. 1.1: Likelihood function of motivating example

Figure 1.1 indicates that the value $p = \frac{3}{5}$ produces a higher probability than any of the other values considered, agreeing with the intuition that a reasonable estimate of p may be the number of heads divided by the total number of coin flips. Calculus based techniques enable algebraic derivation of the MLE of the Binomial distribution analytically. We demonstrate this next.

1.2.1 Binomial distribution

The Binomial distribution is built from a sequence of binomial random variables possessing probability mass function (pmf)

$$p_x(x) = p^x(1-p)^{1-x}, \tag{1.5}$$

where $0 \leq p \leq 1$ and $x = \{0,1\}$ with $x = 1$ denoting success and $x = 0$ failure. The joint probability is

$$\begin{aligned} \Pr\{X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\} &= \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} \\ &= p^{\Sigma x_i}(1-p)^{n-\Sigma x_i} \end{aligned} \tag{1.6}$$

where the last equality follows from Equation (A.7). Therefore, the likelihood equation is

$$L(p) = p^{\Sigma x_i}(1-p)^{n-\Sigma x_i} \tag{1.7}$$

Due to the monotonicity of the logarithm, one can maximize $\ln(L(p))$ is in place of $L(p)$, which enables simplification

$$\ln(L(p)) = \sum_{i=1}^{n} x_i \ln p + \left(n - \sum_{i=1}^{n} x_i\right) \ln(1-p) \tag{1.8}$$

which follows from Equation (A.13). Differentiating with respect to p

$$\frac{\partial \ln(L(p))}{\partial p} = \frac{1}{p}\sum_{i=1}^{n} x_i - \frac{1}{1-p}\left(n - \sum_{i=1}^{n} x_i\right) \tag{1.9}$$

which follows from Equation (A.5). Setting the result equal to zero and solving for p

$$\begin{aligned} \frac{n}{1-p} &= \frac{1}{p}\sum_{i=1}^{n} x_i + \frac{1}{1-p}\sum_{i=1}^{n} x_i \\ np &= (1-p)\sum_{i=1}^{n} x_i + p\sum_{i=1}^{n} x_i \\ \hat{p} &= \frac{1}{n}\sum_{i=1}^{n} x_i \end{aligned} \tag{1.10}$$

where $\hat{p}$ is the maximum likelihood estimate of parameter p. Equation (1.10) is indeed the sum of the successes divided by the total number of experiments as conjectured from the numerical example with five coin flips introduced above.

General procedure

The general steps of the estimation process are as follows.

$$L(\boldsymbol{\Theta}) = \Pr\{X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n | \boldsymbol{\Theta}\} = \prod_{i=1}^{n} p_{x_i}(x_i | \boldsymbol{\Theta}) \quad (1.11)$$

$$\ln L(\boldsymbol{\Theta}) = \ln \prod_{i=1}^{n} f_{x_i}(x_i | \boldsymbol{\Theta}) = \sum_{i=1}^{n} \ln f_{x_i}(x_i | \boldsymbol{\Theta}) \quad (1.12)$$

$$\frac{\partial \ln L(\boldsymbol{\Theta})}{\partial \theta_i} = 0 \quad (1.13)$$

1.3 Exercise

1. Compute the maximum likelihood estimate of the following distribution
   a. Poisson distribution:
   $$p_x(x) = e^{-\lambda} \frac{\lambda^x}{x!}$$
   b. Geometric distribution:
   $$p_x(x) = p(1-p)^{x-1}$$
   c. Negative binomial distribution:
   $$p_x(x) = p^{\sigma}(1-p)^{x-\sigma}$$
   d. Normal distribution:
   $$f_x(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Chapter 2
The Exponential Distribution

The probability density function (PDF) of a continuous random variable X possessing an exponential distribution with parameter $\lambda > 0$ is

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & : \quad x \geq 0 \\ 0 & : \quad x < 0 \end{cases} \tag{2.1}$$

The corresponding cumulative distribution function (CDF) is

$$F(x) = \int_{-\infty}^{x} f(y)dy = \begin{cases} 1 - e^{-\lambda x} & : \quad x \geq 0 \\ 0 & : \quad x < 0 \end{cases} \tag{2.2}$$

with $F(0) = 0$ and $F(\infty) = 1$.

The expected value is

$$E[X] = \int_{-\infty}^{\infty} x f(x)dx = \int_{0}^{\infty} \lambda x e^{-\lambda x} dx \tag{2.3}$$

Integration by parts with $u = x$ and $dv = \lambda e^{-\lambda x}dx$ produces

$$\begin{aligned} E[X] &= uv\Big|_0^\infty - \int_0^\infty v du \\ &= -xe^{-\lambda x}\Big|_0^\infty + \int_0^\infty e^{-\lambda x} dx \\ &= \frac{1}{\lambda} \end{aligned} \tag{2.4}$$

The probability that $t_1 \leq X \leq t_2$ is

$$\begin{aligned}
\Pr\{t_1 \le X \le t_2\} &= \int_{t_1}^{t_2} f(x)dx \\
&= F(t_2) - F(t_1) \\
&= \left(1 - e^{-\lambda t_2}\right) - \left(1 - e^{-\lambda t_1}\right) \\
&= e^{-\lambda t_1} - e^{-\lambda t_2}
\end{aligned} \tag{2.5}$$

Thus, the probability that $X \ge t$ is

$$\begin{aligned}
\Pr\{X \ge t\} &= \int_{t}^{\infty} f(x)dx \\
&= F(\infty) - F(t) \\
&= 1 - \left(1 - e^{-\lambda t}\right) \\
&= e^{-\lambda t}
\end{aligned} \tag{2.6}$$

The *memorylessness property* may be written as

$$\begin{aligned}
\Pr\{X > t_1 + t_2 | X > t_1\} &= \frac{\Pr\{X > t_1 + t_2, X > t_1\}}{\Pr\{X > t_1\}} \\
&= \frac{e^{-\lambda(t_1+t_2)}}{e^{-\lambda t_1}} \\
&= \frac{e^{-\lambda t_1} e^{-\lambda t_2}}{e^{-\lambda t_1}} \\
&= e^{-\lambda t_2} \\
&= \Pr\{X > t_2\}
\end{aligned} \tag{2.7}$$

2.1 Reliability and failure rate

If the time to failure of a system is a random variable X, the reliability of the system is

$$R(t) = \Pr\{X > t\} = 1 - F(t) \tag{2.8}$$

It follows from the continuity of $F(t)$ that

$$R'(t) = -f'(t) \tag{2.9}$$

The probability of failure in the interval $t + \Delta$ given that the system has survived to time t is

$$\frac{\Pr\{t < X \leq t + \Delta\}}{\Pr\{X > t\}} = \frac{F(t+\Delta) - F(t)}{R(t)} \tag{2.10}$$

Therefore, the instantaneous failure rate (also know as the hazard rate) is

$$\begin{aligned}
h(t) &= \lim_{\Delta \to 0} \frac{1}{\Delta} \frac{F(t+\Delta) - F(t)}{R(t)} \\
&= \frac{1}{R(t)} \lim_{\Delta \to 0} \frac{(1 - R(t+\Delta)) - (1 - R(t))}{\Delta} \\
&= \frac{1}{R(t)} \lim_{\Delta \to 0} \frac{R(t) - R(t+\Delta)}{\Delta} \\
&= -\frac{1}{R(t)} \lim_{\Delta \to 0} \frac{R(t+\Delta) - R(t)}{\Delta} \\
&= -\frac{R'(t)}{R(t)} \\
&= \frac{f(t)}{R(t)}
\end{aligned} \tag{2.11}$$

The hazard rate of the exponential distribution is

$$h(t) = \frac{f(t)}{R(t)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda \tag{2.12}$$

Expressing the hazard rate as

$$\begin{aligned}
h(t) &= \frac{f(t)}{R(t)} \\
&= \frac{\frac{d}{dt} F(t)}{1 - F(t)}
\end{aligned} \tag{2.13}$$

and integrating over the interval $[0,t]$

$$\begin{aligned}
-\int_0^t h(x)dx + c_0 &= \log(1 - F(t)) \\
c_1 e^{-\int_0^t h(x)dx} &= 1 - F(t) \\
1 - R(t) &= 1 - e^{-\int_0^t h(x)dx} \\
R(t) &= e^{-\int_0^t h(x)dx} \\
R(t) &= e^{-H(t)}
\end{aligned} \tag{2.14}$$

where $c_1 = 1$ follows from $F(0) = 0$ and $H(t) = \int_0^t h(x)dx$ is the cumulative hazard rate. A constant hazard $h(x) = \lambda$ possess cumulative hazard rate $H(t) = \lambda t$, which is clearly the exponential distribution.

2.2 Maximum likelihood estimation

To illustrate maximum likelihood estimation in the context of a continuous distribution, the method is applied to the exponential distribution, which commonly occurs in reliability mathematics. The pdf of the exponential distribution is

$$f_x(x) = \lambda e^{-\lambda x} \quad x > 0 \ldots \tag{2.15}$$

Therefore, by Equation (1.3) the likelihood equation is

$$\begin{aligned} L(\lambda) &= \prod_{i=1}^{n} \lambda e^{-\lambda x_i} \\ &= \lambda^n e^{-\lambda \sum_{i=1}^{n} x_i} \end{aligned} \tag{2.16}$$

and the log-likelihood is

$$\ln(L(\lambda)) = n \ln \lambda - \lambda \sum_{i=1}^{n} x_i \tag{2.17}$$

Next, differentiate with respect to λ

$$\frac{\partial \ln(L(p))}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i \tag{2.18}$$

Finally, set the result equal to zero and solve for λ

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} x_i} \tag{2.19}$$

Rearranging Equation (2.19)

$$\frac{1}{\hat{\lambda}} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{2.20}$$

When the x_i in Equation (2.20) are interpreted as failure times, this suggest that the MLE of the mean time to failure $\frac{1}{\hat{\lambda}}$ is the sum of the failure times divided by the number of items, which agrees with intuition as the simple average of the time to failure.

2.3 Exercises

1. Given the following eight values compute the MLE assuming the data came from an exponential distribution

   $$K = \{0.7467, 0.3653, 0.2893, 2.0946, 0.6189, 0.0338, 1.0123, 2.3602\}$$

2. Compute the value of the fitted exponential and true ($\lambda = 1$) PDF at $x = 3$.
3. Repeat exercise 2 for the CDF.
4. Plot the value of the fitted exponential and true ($\lambda = 1$) pdf in the interval $x \in (0, 5)$.
5. Repeat exercise 4 for the CDF.
6. Compute the value of the fitted and true ($\lambda = 1$) expected value.
7. Compute the probability $1 \leq x \leq 2$ for the fitted and true ($\lambda = 1$) value of the exponential parameter. Repeat this exercise for $2 \leq x \leq 3$.
8. Compute the probability $x \geq 2$ for the fitted and true ($\lambda = 1$) value of the exponential parameter. Repeat this exercise for $x \geq 3$.

Chapter 3
Failure Rate Models - Theory

Failure rate models characterize software reliability in terms of failure data collected during software testing. Some of the estimates enabled by failure rate models include:

1. Number of initial faults.
2. Number of remaining faults.
3. Mean time to failure (MTTF) of the $(n+1)$st fault.
4. Testing time needed to remove the next k faults.
5. Probability that the software does not fail before a mission of fixed duration is complete.

Intuitively, the failure rate decreases as the number of fault detected increases. Thus, additional testing time should lower the failure rate.

3.1 Jelinski-Moranda

The Jelinski-Moranda (JM) [Jelinski and Moranda, 1972] model is was one of the earliest failure rate models. It has also been extended by many subsequent studies. The assumptions of the JM model are:

1. The number of initial software faults $N_0 > 0$ is a unknown but fixed constant.
2. Each fault contributes equally to software failure.
3. Times between failures are independent an exponentially distributed with rate proportional to the number of remaining faults.
4. A detected fault is removed immediately and no new faults are introduced during the fault removal process.

Let **T** be the vector of n observed failure times. The time between the $(i-1)$st and ith failure is $t_i = (T_i - T_{i-1})$, which is exponentially distributed with rate

$$\lambda(i) = \phi\left(N_0 - (i-1)\right), \qquad i = 1, 2, \ldots, n. \tag{3.1}$$

Here, $\phi > 0$ is a constant of proportionality representing the contribution of individual faults to the overall failure rate.

The probability distribution function of the ith failure is

$$\begin{aligned} f_i(t) &= \lambda(i) e^{-\int_0^t \lambda(i) dx} \\ &= \phi\left(N_0 - (i-1)\right) e^{-\int_0^t \phi(N_0-(i-1))dx} \\ &= \phi\left(N_0 - (i-1)\right) e^{-\phi(N_0-(i-1))t} \end{aligned} \tag{3.2}$$

where the second equality is the hazard rate representation of the pdf $\left(h(\tau)e^{-\int_0^t h(\tau)d\tau}\right)$ and $H(\tau) = \int_0^t h(\tau)d\tau$ is the cumulative hazard rate.

The cumulative distribution function (CDF) of the i^{th} failure is

$$\begin{aligned} F_i(t) &= \int_0^t f_i(x)dx \\ &= \int_0^t \phi\left(N_0 - (i-1)\right) e^{-\phi(N_0-(i-1))x} dx \\ &= 1 - e^{-\phi(N_0-(i-1))t} \end{aligned} \tag{3.3}$$

where the final equality follows from $\int_0^t \lambda e^{-\lambda x} dx = 1 - e^{-\lambda t}$

The reliability of the software prior to the i^{th} failure is therefore

$$R_i(t) = e^{-\phi(N_0-(i-1))t} \tag{3.4}$$

and the mean time to the $(n+1)$st failure is

$$\begin{aligned} MTTF(n+1) &= \int_0^\infty R(t|i = n+1)dt \\ &= \int_0^\infty e^{-\phi(N_0-n)t} dt \\ &= \frac{1}{\phi\left(N_0 - n\right)} \end{aligned}$$

where the final equality follows from $\int_0^\infty e^{-\lambda t} = \frac{1}{\lambda}$.

More generally, the amount of testing required to remove the next k faults is

$$\sum_{i=1}^{k} MTTF(n+i) = \sum_{i=1}^{k} \frac{1}{\phi\,(N_0-(n+(i-1)))} \tag{3.5}$$

The likelihood function of the JM model parameters is

$$\begin{aligned} L(t_1,t_2,\ldots,t_n;N_0,\phi) &= \prod_{i=1}^{n} \phi\,(N_0-(i-1))\,e^{-\phi(N_0-(i-1))t_i} \\ &= \phi^n \left(\prod_{i=1}^{n} N_0-(i-1)\right) e^{-\phi\sum_{i=1}^{n}(N_0-(i-1))t_i} \end{aligned} \tag{3.6}$$

The log likelihood function is

$$\begin{aligned} \ln L &= \ln\left(\phi^n \left(\prod_{i=1}^{n} N_0-(i-1)\right) e^{-\phi\sum_{i=1}^{n}(N_0-(i-1))t_i}\right) \\ &= \ln\phi^n + \ln\left(\prod_{i=1}^{n} N_0-(i-1)\right) - \phi\sum_{i=1}^{n}(N_0-(i-1))\,t_i \\ &= n\ln\phi + \sum_{i=1}^{n}\ln(N_0-(i-1)) - \phi\sum_{i=1}^{n}(N_0-(i-1))\,t_i \end{aligned} \tag{3.7}$$

The partial derivatives with respect to N_0 and ϕ are

$$\frac{\partial \ln L}{\partial N_0} = \sum_{i=1}^{n} \frac{1}{N_0-(i-1)} - \phi\sum_{i=1}^{n} t_i \tag{3.8}$$

and

$$\frac{\partial \ln L}{\partial \phi} = \frac{n}{\phi} - \sum_{i=1}^{n}(N_0-(i-1))\,t_i \tag{3.9}$$

Equating Equation (3.8) to zero and solving for ϕ,

$$\hat{\phi} = \frac{\sum_{i=1}^{n}\frac{1}{N_0-(i-1)}}{\sum_{i=1}^{n} t_i} \tag{3.10}$$

Next, equating Equation (3.9) to zero and substitute Equation (3.10) for ϕ reduces to the following fixed point problem with N_0 as the only unknown.

$$\frac{n\sum_{i=1}^{n} t_i}{\sum_{i=1}^{n}\frac{1}{\hat{N}_0-(i-1)}} - \sum_{i=1}^{n}\left(\hat{N}_0-(i-1)\right)t_i = 0 \tag{3.11}$$

After solving Equation (3.11) for $\hat{N}_0$, substituting this value into Equation (3.10) provides the estimate $\hat{\phi}$.

Alternatively, one may equate Equation (3.9) to zero and solve for ϕ,

$$\hat{\phi} = \frac{n}{\sum_{i=1}^{n} \left(N_0 - (i-1)\right) t_i} \tag{3.12}$$

Now, substitute Equation (3.12) into Equation (3.8) and equate to zero, to give

$$\sum_{i=1}^{n} \frac{1}{N_0 - (i-1)} - \frac{n}{\sum_{i=1}^{n} \left(N_0 - (i-1)\right) t_i} \sum_{i=1}^{n} t_i = 0 \tag{3.13}$$

Table 6.1 summarizes some additional failure rate models derived from the JM.

Table 3.1: Additional failure rate models.

Model name	Failure rate $\lambda(i)$	Parameter constraints
Goel-Okumoto imperfect	$\phi\left(N_0 - p(i-1)\right)$	$N_0, \phi > 0, 0 < p < 1$
Power-type function	$\phi\left(N_0 - (i-1)\right)^{\alpha}$	$N_0, \phi, \alpha > 0$
Exponential-type function	$\phi\left(e^{-\beta(N_0-(i-1))} - 1\right)$	$N_0, \phi, \beta > 0$
Schick-Wolverton	$\phi\left(N_0 - (i-1)\right) t_i$	$N_0, \phi > 0$
Modified Schick-Wolverton	$\phi\left(N_0 - n_{i-1}\right) t_i$	$N_0, \phi > 0$
Lipow	$\phi\left(N_0 - (i-1)\right)\left(\frac{1}{2}t_i + \sum_{j=1}^{i-1} t_j\right)$	$N_0, \phi > 0$
Geometric model	ϕp^i	$\phi > 0, 0 < p < 1$
Fault exposure coefficient	$k(j)\left(N_0 - j\right)$, $k(j) = k_i + \frac{(k_f - k_i) j}{N_0}$	$N_0, k_i, k_j > 0$

Chapter 4
Failure Rate Models - Application

4.1 Illustration of Jelinski-Moranda Model

This section illustrates the Jelinski-Moranda model for the SYS1 [Lyu, 1996] failure data set which consists of $n = 136$ faults. Figure 4.1 shows the plot of the log-likelihood function, which has been reduced to N_0 by substituting Equation (3.10) into Equation (3.7). Examination of Figure 4.1 suggests that the log-likelihood function is maximized at $N_0 \approx 142$, which can serve as the initial input to a numerical procedure to find the exact maximum.

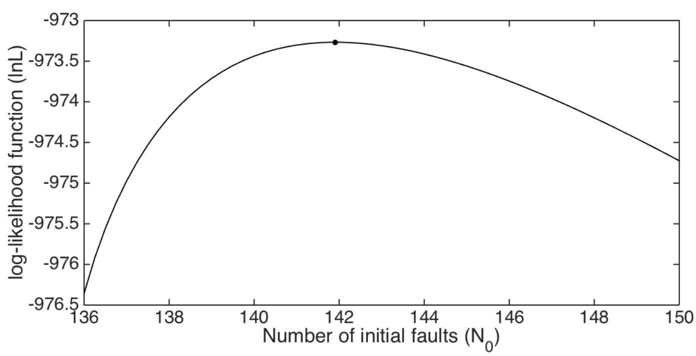

Fig. 4.1: Log-likelihood function of JM model

Figure 4.2 shows a plot of the MLE of N_0 specified by Equation (3.11). The value of N_0 that satisfies Equation (3.11) is the maximum likelihood estimate $\hat{N}_0$. This value is indicated with a dot. The grid lines suggest that this value is slightly less than 142.

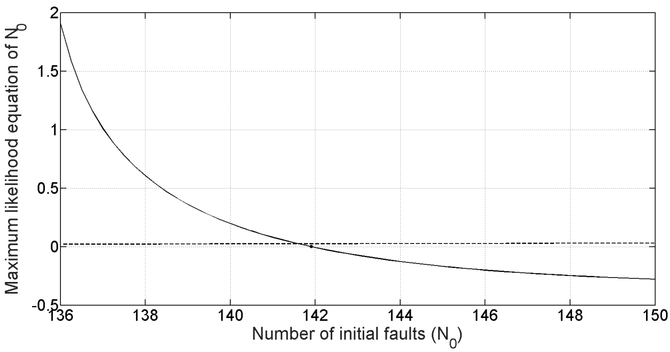

Fig. 4.2: MLE of N_0

The parameter estimates for the SYS1 data set are

1. Estimated number of initial faults using Equation (3.11) or (3.13): $\hat{N}_0 = 141.903$
2. Failure constant using Equation (3.10) or (3.12): $\hat{\phi} = 0.0000349665$
3. Estimated number of faults remaining: 5.902

$$(\hat{N}_0 - n = 141.903 - 136)$$

4. Reliability function after 136^{th} failure using Equation (3.4):

$$R(t) = e^{(-0.00003497(141.903-(137-1))t)}$$

Figure 4.3 shows the plot of the reliability after 136th failure
Figure 4.3 suggests that the probability that the software operates without failure for 6,000 seconds is approximately 0.3. The MTTF of the 137th fault is $\int_0^\infty R(t)dt = 4844.89$ or $\frac{4844.89}{60} = 80.7482$ minutes.

Figure 4.4 shows the plot of the MLE of the failure rate.
Figure 4.4 shows that the failure rate decreases by a constant of size $\hat{\phi}$ at each failure time t_i. Early testing decreases the failure rate more quickly because there are more faults to detect. Later testing lowers the failure rate more slowly because fewer faults remain.

Figure 4.5 shows the plot of the MLE of the MTTF

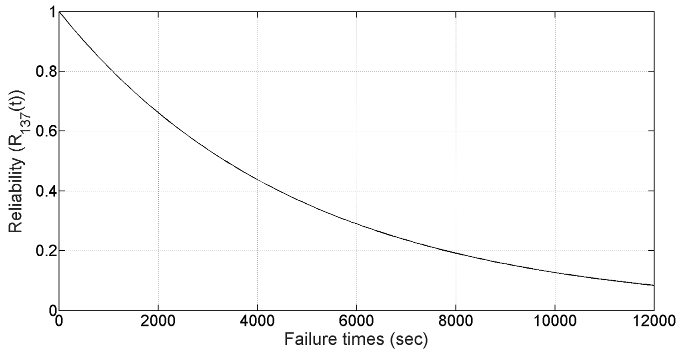

Fig. 4.3: Reliability after 136th failure

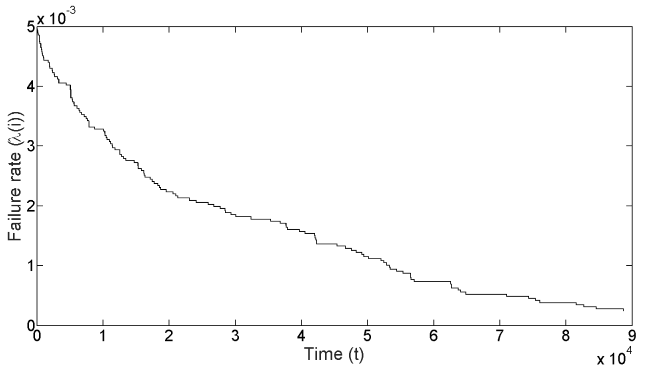

Fig. 4.4: MLE of failure rate

Figure 4.5 indicates that the initial MTTF is low because there are frequent failures. The MTTF increases as faults are removed because fewer faults remain and the failure rate has decreased.

4.2 Exercises

1. In the Geometric Model, the time between the $(i-1)$st and ith failure is exponential with rate

$$\lambda(i) = D\phi^{i-1}$$

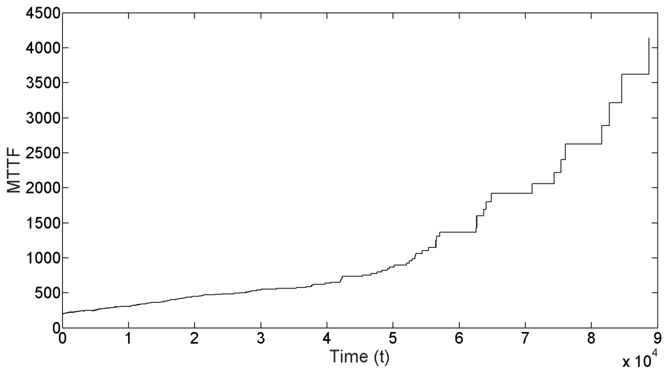

Fig. 4.5: MLE of MTTF

where D is the initial failure rate and ϕ is geometrically decreasing. Thus, the probability distribution function of the ith failure is

$$f(t_i) = D\phi^{i-1}e^{-D\phi^{i-1}t_i}$$

a. Write the likelihood function of the Geometric Model similar to Equation (3.6).
b. Write the log likelihood function of the Geometric Model similar to Equation (4.1).
c. Derive the maximum likelihood estimators of model parameters D and ϕ.

2. Consider the SYS2 data set consisting of 86 time between failures [Lyu, 1996]

a. Estimate the parameters D and ϕ of the Geometric Model. Use MLEs derived in Exercise 1c to compute numerical values.
b. Based on the MLEs, calculate the purification level $(1 - \phi^n)$. Use the value of ϕ computed in Exercise 2a and n indicates length of the SYS2 data.
c. Compute the reliability function for the 87th fault (use Equation (3.4)).
d. Compute the Mean time to the 87th fault (use Equation (3.5)).
e. Compute the log likelihood of the Geometric Model at the MLE. Substitute values of D and ϕ computed in Exercise 2a in Equation (3.7).
f. Plot the failure rate in Equation (3.1) as a function of time.

Chapter 5
Non-homogeneous Poisson Process Models

5.1 Homogeneous Poisson process

A stochastic process $\{N(t), t \geq 0\}$ is a *counting process* [Min, 1991] if $N(t)$ represents the cumulative number of events observed by time t. A counting process must satisfy the following properties:

1. $N(t) \geq 0$.
2. $N(t)$ is a natural number.
3. $t_1 < t_2$ implies $N(t_1) \leq N(t_2)$.
4. For $t_1 < t_2$, $N(t_2) - N(t_1)$ is the number of events occuring in the interval $[t_1, t_2]$.

A counting process possesses *independent increments* if the number of events that occur in disjoint time intervals are independent. A counting process possesses *stationary increments* if the distribution of the number of events that occur in any time interval depends only on the length of the interval.

The *Poisson process* is a counting process with rate $\lambda > 0$, possessing the following properties

1. N(0)=0.
2. The process has independent increments.
3. The number of events in any interval of length t is Poisson with mean λt

$$\Pr\{N(t_2) - N(t_1) = k\} = e^{-\lambda t} \frac{(\lambda t)^k}{k!} \tag{5.1}$$

Condition 3 implies that $E[N(t)] = \lambda t =: m(t)$, which is known as the mean value function (MVF). It also follows that the time between events of a Poisson process are exponentially distributed, because the probability of no events in an interval $[t_1, t_2]$ of length t is

$$\begin{aligned}\Pr\{N(t_2) - N(t_1) = 0\} &= e^{-\lambda t}\frac{(\lambda t)^0}{0!} \\ &e^{-\lambda t}\end{aligned} \tag{5.2}$$

Hence,

$$F(t) = 1 - e^{-\lambda t} \tag{5.3}$$

An alternative formulation of the Poisson process with rate λ

1. N(0)=0.
2. The process has independent increments.
3. $\Pr\{N(\Delta) = 1\} = \lambda\Delta + o(\Delta)$.
4. $\Pr\{N(\Delta) \geq 2\} = o(\Delta)$.

where a function f is $o(\Delta)$ if

$$\lim_{\Delta \to 0} \frac{f(\Delta)}{\Delta} = 0 \tag{5.4}$$

Condition 1 states that the number of events at time $t = 0$ is 0 [Trivedi, 2001]. Condition 2 is as defined previously. Condition 3 states that the probability of an event in a small interval of length Δ is $\lambda\Delta$, while condition 4 states that the probability of two or more events in a small interval is negligible.

5.2 Non-homogeneous Poisson process

A counting process is a *nonhomogeneous Poisson process* (NHPP) with intensity function $\lambda(t), t \geq 0$ if

1. $N(0) = 0$.
2. $\{N(t), t \geq 0\}$ has independent increments.
3. $\Pr\{N(t+\Delta) - N(t) \geq 2\} = o(\Delta)$
4. $\Pr\{N(t+\Delta) - N(t) = 1\} = \lambda(t)\Delta + o(\Delta)$

If the *mean value function* of an NHPP is

$$m(t) = \int_0^t \lambda(x)dx \tag{5.5}$$

it can be shown that

$$\Pr\{N(t_2) - N(t_1) = k\} = e^{-(m(t_2)-m(t_1))}\frac{(m(t_2) - m(t_1))^k}{k!} \tag{5.6}$$

Substituting the mean value function $m(t) = \lambda t$ the Equation (5.6) simplifies to the Poisson process, also referred to as the *Homogeneous Poisson process* (HPP).

5.3 NHPP likelihood function

The following theorem will be used to derive the maximum likelihood estimates of NHPP software reliability growth models.

Theorem 5.1. *Given n observed event times* $\mathbf{T} = \langle t_1, t_2, \ldots, t_n \rangle$, *the likelihood function an NHPP with mean value function* $m(t)$ *and intensity function* $\lambda(t)$ *possessing parameters* $\boldsymbol{\Theta}$ *is*

$$L(\boldsymbol{\Theta}|\mathbf{T}) = e^{-m(t_n)} \prod_{i=1}^{n} \lambda(t_i)$$

Proof. Let X_i be a random variable representing the time between the $(i-1)$st and ith failures and $\Phi_{X_i}(x_i)$ denoted the CDF of X_i. Then

$$\Phi_{X_1}(x_1) = \Pr\{X_1 \leq x_1\}$$

with corresponding reliability is

$$R_{X_1} = \Pr\{X_1 > x_1\} = \Pr\{N(x_1) = 0\} = e^{-m(x_1)}$$

so that

$$\Phi_{X_1}(x_1) = 1 - R_{X_1} = 1 - e^{-m(x_1)}.$$

The pdf is thus

$$\phi_{X_1}(x_1) = \frac{d}{dx_1}\Phi_{X_1}(x_1) = \frac{d}{dx_1}\left(1 - e^{-m(x_1)}\right) = m'(x_1)e^{-m(x_1)} = \lambda(x_1)e^{-m(x_1)}.$$

Now, the conditional reliability of the second failure given the first is

$$\begin{aligned}
R_{X_2|X_1}(x_2|x_1) &= \Pr\{X_2 > x_2 | X_1 = x_1\} \\
&= \Pr\{\texttt{No failures in } (x_1, x_1 + x_2]\} \\
&= \Pr\{N(x_1 + x_2) - N(x_1) = 0\} \\
&= e^{-(m(x_1+x_2)-m(x_1))}
\end{aligned}$$

so that

$$\Phi_{X_2|X_1}(x_2|x_1) = 1 - R_{X_2|X_1}(x_2|x_1) = 1 - e^{-(m(x_1+x_2)-m(x_1))}$$

Hence, the conditional pdf is

$$\begin{aligned}
\phi_{X_2|X_1}(x_2|x_1) &= \frac{d}{dx_2}\Phi_{X_2|X_1}(x_2|x_1) \\
&= \frac{d}{dx_2}\left(1 - e^{-(m(x_1+x_2)-m(x_1))}\right) \\
&= m'(x_1+x_2)e^{-(m(x_1+x_2)-m(x_1))} \\
&= \lambda(x_1+x_2)e^{-(m(x_1+x_2)-m(x_1))}
\end{aligned}$$

The joint distribution of X_1 and X_2 is

$$\begin{aligned}
\phi_{X_1,X_2}(x_1,x_2) &= \phi_{X_2|X_1}(x_2|x_1)\phi_{X_1}(x_1) \\
&= \lambda(x_1+x_2)e^{-(m(x_1+x_2)-m(x_1))}\lambda(x_1)e^{-m(x_1)} \\
&= \lambda(x_1)\lambda(x_1+x_2)e^{-(m(x_1+x_2))}
\end{aligned}$$

This suggests that

$$\phi_{X_1,X_2,\ldots,X_k}(x_1,x_2,\ldots,x_k) = e^{-m(t_k)}\prod_{i=1}^{k}\lambda(t_i), \tag{5.7}$$

where $t_i = \sum_{j=1}^{i} x_i$ is the time of the ith failure. This will serve as the inductive hypothesis. We first show that

$$\begin{aligned}
\phi_{X_n|X_1,X_2,\ldots,X_{n-1}}(x_n|x_1,x_2,\ldots,x_{n-1}) &= \frac{d}{dx_n}\Phi_{X_n|X_1,X_2,\ldots,X_{n-1}}(x_n|x_1,x_2,\ldots,x_{n-1}) \\
&= \frac{d}{dx_n}\left(1 - e^{-(m(t_{n-1}+x_n)-m(t_{n-1}))}\right) \\
&= m'(t_{n-1}+x_n)e^{-(m(t_{n-1}+x_n)-m(t_{n-1}))} \\
&= \lambda(t_{n-1}+x_n)e^{-(m(t_{n-1}+x_n)-m(t_{n-1}))} \\
&= \lambda(t_n)e^{-(m(t_n)-m(t_{n-1}))}
\end{aligned}$$

and hence

$$\begin{aligned}
\phi_{X_1,X_2,\ldots,X_n}(x_1,x_2,\ldots,x_n) &= \phi_{X_n|X_1,X_2,\ldots,X_{n-1}}(x_n|x_1,x_2,\ldots,x_{n-1}) \\
&\quad \times \phi_{X_1,X_2,\ldots,X_{n-1}}(x_1,x_2,\ldots,x_{n-1}) \\
&= \lambda(t_n)e^{-(m(t_n)-m(t_{n-1}))} \\
&\quad \times e^{-m(t_{n-1})}\prod_{i=1}^{n-1}\lambda(t_i) \qquad (5.8) \\
&= e^{-m(t_n)}\prod_{i=1}^{n}\lambda(t_i)
\end{aligned}$$

where Equation (5.8) follows from the inductive hypothesis given in Equation (5.7), completing the proof.

The log likelihood function is

$$\begin{aligned}
\log L(\boldsymbol{\Theta}|\mathbf{T}) &= \log\left(e^{-m(t_n)}\prod_{i=1}^{n}\lambda(t_i)\right) \\
&= -m(t_n)+\sum_{i=1}^{n}\log(\lambda(t_i)) \qquad (5.9)
\end{aligned}$$

The maximum likelihood estimate of a model with p parameters is obtained by solving the system of equations

$$\frac{\partial}{\partial\theta_i}\log L(\boldsymbol{\Theta}|\mathbf{T}) = 0, \qquad (1 \le i \le p) \qquad (5.10)$$

Chapter 6
Goel-Okumoto NHPP Software Reliability Growth Model

This section introduces the Goel-Okumoto (GO) [Goel and Okumoto, 1979] software reliability growth model (SRGM). The mean value function is developed from its formulation as a differential equation and the and maximum likelihood estimates for failure time and failure count data are derived.

6.1 Model formulation

The Goel-Okumoto is formulated as

$$m(t+\Delta t)-m(t)=b(a-m(t))\Delta t \tag{6.1}$$

This expression suggests that the rate of fault detection in the time interval from $[t,t+\Delta t)$ is equal to the product of: a constant fault detection rate (b), the number of faults remaining at time t $(a-m(t))$, and the length of the interval (Δt).

$$
\begin{aligned}
\frac{m(t+\Delta t)-m(t)}{\Delta t} &= b(a-m(t)) \\
\lim_{\Delta t\to 0}\frac{m(t+\Delta t)-m(t)}{\Delta t} &= \lim_{\Delta t\to 0} b(a-m(t)) \\
m'(t) &= b(a-m(t)) \\
\frac{m'(t)}{a-m(t)} &= b \\
-\frac{du}{u} &= b \qquad (u=a-m(t), \quad du=-m'(t)) \\
-\log_e u &= bt+c_0 \\
u &= e^{-bt+c_0} \\
a-m(t) &= e^{-bt}e^{c_0} \\
m(t) &= a-c_1e^{-bt}
\end{aligned}
\tag{6.2}
$$

Given the initial condition $m(0)=0$, it follows that

$$
\begin{aligned}
m(0) &= a-c_1e^{-b\cdot 0} \\
0 &= a-c_1e^{0} \\
c_1 &= a
\end{aligned}
\tag{6.3}
$$

Hence the mean value function of the GO model is

$$
\begin{aligned}
m(t) &= a-ae^{-bt} \\
&= a\left(1-e^{-bt}\right)
\end{aligned}
\tag{6.4}
$$

$a>0$, $b>0$.

Note that all faults are detected as t tends to infinity

$$\lim_{t\to\infty} m(t)=a \tag{6.5}$$

6.2 Failure times MLE

Recall that the definition of the mean value function is

$$m(t)=\int_0^{\infty}\lambda(x)dx \tag{6.6}$$

where $\lambda(t)$ is the instantaneous failure intensity at time t.

It follows that the instantaneous failure intensity is

$$\frac{dm(t)}{dt} = \lambda(t). \tag{6.7}$$

Thus, the instantaneous failure intensity of the GO SRGM is

$$\begin{aligned}\lambda(t) &= \frac{d}{dt}\left[a\left(1-e^{-bt}\right)\right] \\ &= abe^{-bt}\end{aligned} \tag{6.8}$$

Note that Equation (6.8) is monotonically decreasing with

$$\lim_{t\to\infty}\lambda(t) = 0 \tag{6.9}$$

The log likelihood function is

$$\begin{aligned}\log L(\blacksquare|\mathbf{T}) &= \log\left(e^{-m(t_n)}\prod_{i=1}^{n}\lambda(t_i)\right) \\ &= -m(t_n) + \sum_{i=1}^{n}\log\left(\lambda(t_i)\right)\end{aligned} \tag{6.10}$$

The maximum likelihood estimate of a model with p parameters is obtained by solving the system of equations

$$\frac{\partial}{\partial\theta_i}\log L(\blacksquare|\mathbf{T}) = 0, \qquad (1 \le i \le p) \tag{6.11}$$

From Equation (6.10), the log likelihood function of the GO SRGM given the failure times is

$$\begin{aligned}\log L(a,b|\mathbf{T}) &= -a\left(1-e^{-bt_n}\right) + \sum_{i=1}^{n}\log\left(abe^{-bt_i}\right) \\ &= -a\left(1-e^{-bt_n}\right) + \sum_{i=1}^{n}\log a + \sum_{i=1}^{n}\log b + \sum_{i=1}^{n}\log e^{-bt_i} \\ &= -a\left(1-e^{-bt_n}\right) + n\log a + n\log b - b\sum_{i=1}^{n}t_i\end{aligned} \tag{6.12}$$

Differentiating Equation (6.12) with respect to a,

$$
\begin{aligned}
\frac{\partial}{\partial a}[\log L(a,b|\mathbf{T})] &= \frac{\partial}{\partial a}\left[-a\left(1-e^{-bt_n}\right)+n\log a+n\log b-b\sum_{i=1}^{n}t_i\right] \\
&= \frac{\partial}{\partial a}\left[-a\left(1-e^{-bt_n}\right)\right]+\frac{\partial}{\partial a}[n\log a]+\frac{\partial}{\partial a}[n\log b]-\frac{\partial}{\partial a}\left[b\sum_{i=1}^{n}t_i\right] \\
&= -\left(1-e^{-bt_n}\right)+\frac{n}{a} \qquad (6.13)
\end{aligned}
$$

Equating Equation (6.13) to zero and solving for a,

$$
\begin{aligned}
1-e^{-bt_n} &= \frac{n}{a} \\
a &= \frac{n}{1-e^{-bt_n}} \qquad (6.14)
\end{aligned}
$$

Differentiating Equation (6.12) with respect to b,

$$
\begin{aligned}
\frac{\partial}{\partial b}[\log L(a,b|\mathbf{T})] &= \frac{\partial}{\partial b}\left[-a\left(1-e^{-bt_n}\right)+n\log a+n\log b-b\sum_{i=1}^{n}t_i\right] \\
&= \frac{\partial}{\partial b}\left[-a+ae^{-bt_n}\right]+\frac{\partial}{\partial b}[n\log a]+\frac{\partial}{\partial b}[n\log b]-\frac{\partial}{\partial b}\left[b\sum_{i=1}^{n}t_i\right] \\
&= -at_ne^{-bt_n}+\frac{n}{b}-\sum_{i=1}^{n}t_i \qquad (6.15)
\end{aligned}
$$

Equating Equation (6.15) to zero and substituting Equation (6.14) for a,

$$
\frac{nt_ne^{-bt_n}}{1-e^{-bt_n}}+\sum_{i=1}^{n}t_i=\frac{n}{b} \qquad (6.16)
$$

6.3 Failure counts MLE

In some instances, failure data is grouped into n consecutive, nonoverlapping, and possibly unequal intervals $[t_{i-1},t_i]$, $1\le i\le n$. The maximum likelihood estimates can also be determined from this vector of failure counts data $\mathbf{K}=\langle k_1,k_2,\ldots,k_n\rangle$. The probability of k failures in an interval i is

$$
\Pr\{m(t_i)-m(t_{i-1})=k_i\}=\frac{(m(t_i)-m(t_{i-1}))^{k_i}e^{-(m(t_i)-m(t_{i-1}))}}{k_i!} \qquad (6.17)
$$

and the likelihood function of a vector of parameters $\boldsymbol{\Theta}$ given n failure time intervals $\mathbf{T}$ and failure counts $\mathbf{K}$ is

$$\begin{aligned} L(\boldsymbol{\Theta}|\mathbf{T},\mathbf{K}) &= f(\mathbf{T},\mathbf{K}|m(t)) \\ &= \prod_{i=1}^{n} \frac{(m(t_i)-m(t_{i-1}))^{k_i} e^{-(m(t_i)-m(t_{i-1}))}}{k_i!} \end{aligned} \tag{6.18}$$

The log likelihood function is

$$\begin{aligned} \log L(\boldsymbol{\Theta}|\mathbf{T},\mathbf{K}) &= \log\left(\prod_{i=1}^{n} \frac{(m(t_i)-m(t_{i-1}))^{k_i} e^{-(m(t_i)-m(t_{i-1}))}}{k_i!}\right) \\ &= \sum_{i=1}^{n} \log\left(\frac{(m(t_i)-m(t_{i-1}))^{k_i} e^{-(m(t_i)-m(t_{i-1}))}}{k_i!}\right) \\ &= \sum_{i=1}^{n} k_i \log(m(t_i)-m(t_{i-1})) - \sum_{i=1}^{n}(m(t_i)-m(t_{i-1})) - \sum_{i=1}^{n} \log k_i! \end{aligned} \tag{6.19}$$

To determine the log likelihood function of the GO SRGM from Equation (6.19), first substitute the MVF into

$$\begin{aligned} m(t_i)-m(t_{i-1}) &= a\left(1-e^{-bt_i}\right) - a\left(1-e^{-bt_{i-1}}\right) \\ &= a\left(e^{-bt_{i-1}} - e^{-bt_i}\right) \end{aligned} \tag{6.20}$$

Substituting Equation (6.20) into Equation (6.19) and simplifying produces the

$$\begin{aligned} \log L(a,b|\mathbf{T},\mathbf{K}) &= \sum_{i=1}^{n} k_i \log\left(a\left(e^{-bt_{i-1}} - e^{-bt_i}\right)\right) - \sum_{i=1}^{n}(a\left(e^{-bt_{i-1}} - e^{-bt_i}\right)) - \sum_{i=1}^{n} \log k_i! \\ &= \sum_{i=1}^{n} k_i \log a + \sum_{i=1}^{n} k_i \log\left(e^{-bt_{i-1}} - e^{-bt_i}\right) + a\sum_{i=1}^{n}\left(e^{-bt_i} - e^{-bt_{i-1}}\right) \\ &= \sum_{i=1}^{n} k_i \log a + \sum_{i=1}^{n} k_i \log\left(e^{-bt_{i-1}} - e^{-bt_i}\right) - a\left(1-e^{-bt_n}\right) \end{aligned} \tag{6.21}$$

The term $\sum_{i=1}^{n} \log k_i!$ can be dropped because it is constant with respect to all model parameters. The final step follows from the telescoping series in the denominator and $t_0 = 0$. Furthermore,

Differentiating Equation (6.21) with respect to a,

$$\frac{\partial}{\partial a}[\log L(a,b|\mathbf{T},\mathbf{K})] = \frac{\sum_{i=1}^{n} k_i}{a} - \left(1 - e^{-bt_n}\right) \tag{6.22}$$

Equating Equation (6.22) to zero and solving for a,

$$\frac{\sum_{i=1}^{n} k_i}{a} + \left(1 - e^{-bt_n}\right) = 0$$

$$a = \frac{\sum_{i=1}^{n} k_i}{1 - e^{-bt_n}} \tag{6.23}$$

Differentiating Equation (6.21) with respect to b,

$$\frac{\partial}{\partial b}[\log L(a,b|\mathbf{T},\mathbf{K})] = \sum_{i=1}^{n} \frac{k_i\left(t_i e^{-bt_i} - t_{i-1}e^{-bt_{i-1}}\right)}{e^{-bt_{i-1}} - e^{-bt_i}} - at_n e^{-bt_n} \tag{6.24}$$

Equating Equation (6.24) to zero and substituting Equation (6.23) for a,

$$\sum_{i=1}^{n} \frac{k_i\left(t_i e^{-bt_i} - t_{i-1}e^{-bt_{i-1}}\right)}{e^{-bt_{i-1}} - e^{-bt_i}} = \frac{t_n e^{-bt_n}\sum_{i=1}^{n} k_i}{1 - e^{-bt_n}} \tag{6.25}$$

The following list of software reliability growth models (SRGM) in Table 6.1 are a few of the generalizations of the Goel-Okumoto software reliability growth model.

Table 6.1: Additional NHPP models.

Model name	Mean value function ($m(t)$)	Parameter constraints
Hyperexponential	$a\sum_{i=1}^{k} p_i\left(1 - e^{b_i t}\right)$	$a, b_i > 0, 0 < p_i < 1, \sum_{i=1}^{k} p_i = 1$
Weibull	$a\left(1 - e^{-bt^c}\right)$	$a, b, c > 0$
Delayed S-shaped	$a\left(1 - (1+bt)\,e^{-bt}\right)$	$a, b > 0$
Inflection S-shaped	$\frac{a\left(1-e^{-bt}\right)}{1+\psi e^{-bt}}$	$a, b > 0, \psi(r) = \frac{1-r}{r}, r \in (0,1]$
Bathtub	$a\left(1 - e^{-bct^c e^{\phi t}}\right)$	$a, b, \phi > 0, 0 < c < 1$

6.4 Exercises

1. The Weibull SRGM is a generalization of the Goel-Okumoto model with mean value function

$$m(t) = a(1 - e^{-bt^c})$$

where parameter a is interpreted as the initial number of faults, b the fault detection rate, and c a shape parameter. The case where $c = 1$ simplifies to the Goel-Okumoto SRGM.

a. Derive the intensity of the Weibull SRGM.
b. Write the failure times log likelihood function of the Weibull SRGM.
c. Show that the maximum likelihood estimators of parameters a, b, and c are respectively:

$$a = \frac{n}{1 - e^{-bt_n^c}}$$

$$-at_n^c e^{-bt_n^c} + \frac{n}{b} - \sum_{i=1}^{n} t_i^c$$

and

$$-abt_n^c \ln(t_n) e^{-bt_n^c} + \frac{n}{c} + \sum_{i=1}^{n} \ln(t_i) - b \sum_{i=1}^{n} t_i^c \ln(t_i)$$

2. Consider the SYS1 [Lyu, 1996] data set consisting of 136 time between failures (available at [Lyu, 1996]).

a. Estimate the parameters b and c of the Weibull SRGM.
b. Based on the MLEs of parameters b and c, calculate the MLE of parameter a.
c. Construct a plot showing the observed faults and MVF.

Chapter 7
Goodness-of-fit measures

This section describes some theoretical and practical methods to assess competing software reliability models.

7.1 Number of faults remaining

For models where the number of failures is finite as t goes to infinity, the number of remaining fault may be estimated as

$$\begin{aligned} E[\overline{N}(t)] &= N(\infty) - N(t) \\ &= a - m(t_n) \end{aligned} \tag{7.1}$$

For example, according to the GO model, the number of remaining faults is

$$\begin{aligned} \overline{N}(t) &= a - a\left(1 - e^{-bt}\right) \\ &= a\left(1 - \left(1 - e^{-bt}\right)\right) \\ &= ae^{-bt} \end{aligned} \tag{7.2}$$

7.2 Software reliability

Given a fixed mission duration t_d, software reliability may be defined as the probability of zero failures

$$R(t_d) = \Pr\{N(t_n + t_d) - N(t_n) = 0\} = e^{-(m(t_n+t_d)-m(t_n))} \tag{7.3}$$

Since, the probability of observing k faults by time t is

$$\Pr\{N(t) = k\} = \frac{(m(t))^k e^{-m(t)}}{k!} \tag{7.4}$$

For example, from Equation (6.20), the reliability estimate of the GO SRGM is

$$R(t_d) = e^{-a\left(e^{-bt_n} - e^{-b\left(t_n+t_d\right)}\right)} \tag{7.5}$$

Another useful inference is to estimate the testing time t^* needed to achieve a target reliability of R^* for a mission of duration t_d by solving

$$R^* = e^{-(m(t^*+t_d)-m(t^*))} \tag{7.6}$$

for t^*.

From Equation (7.5), it follows that the estimated testing time for the GO SRGM is

$$\begin{aligned}
R^* &= e^{-\hat{a}\left(e^{-\hat{b}t^*} - e^{-\hat{b}\left(t^*+t_d\right)}\right)} \\
&= e^{-\hat{a}\left(1-e^{-\hat{b}t_d}\right)e^{-\hat{b}t^*}} \\
&= e^{-m(t_d)e^{-\hat{b}t^*}} \\
-\log(R) &= m(t_d)e^{-\hat{b}t^*} \\
\log\log\left(\frac{1}{R}\right) &= \log\left(m(t_d)\right) - \hat{b}t^* \\
\hat{t}^* &= \frac{1}{b}\left[\log\left(m(t_d)\right) - \log\log\left(\frac{1}{R}\right)\right] \qquad (7.7)
\end{aligned}$$

7.3 Trend tests

This section describes trend tests including the Laplace trend test (LTT) and running arithmetic average test, which are used to determine if the data exhibits reliability growth with specified level of confidence.

7.3.1 Laplace trend test

The Laplace trend test (LTT) [Ascher and Feingold, 1984] is a commonly used statistical test for reliability growth for times between failures data. The LTT of the given data is

$$u(i) = \frac{\frac{1}{i-1}\sum_{n=1}^{i-1}\sum_{j=1}^{n} t_j - \frac{\sum_{j=1}^{i} t_j}{2}}{\sum_{j=1}^{i} t_j \sqrt{\frac{1}{12(i-1)}}} \tag{7.8}$$

where, $u(i)$ is is the Laplace test statistic value after i^{th} failure and t_j is the test time between the j^{th} and $(j-1)^{st}$ failures. Small values suggest reliability growth.

7.3.2 Running arithmetic average

The running arithmetic average is computed using

$$\tau(i) = \frac{1}{i}\sum_{j=1}^{i} t_j \tag{7.9}$$

where, t_j denotes the observed inter-failure times, $j = 1, 2, \ldots, i$. An increasing average suggests reliability growth.

7.4 Goodness of fit measures

We now discuss some additional theoretical and practical methods to assess competing software reliability models, in addition to the likelihood ratio test (LRT) and Kolmogorov-Smirnov (KS) test

7.4.1 Akaike information criterion

The Akaike information criterion (AIC) is [Pham]

$$AIC = 2p - 2\ln L(\hat{\boldsymbol{\Theta}}), \tag{7.10}$$

where p is the number of model parameters and $\ln L$ is the maximum of the log likelihood function. Given the AIC of two alternative models i and j, model i is preferred over model j if $AIC_i < AIC_j$. The AIC enables a more equitable comparison of models with different numbers of parameters because each model is "penalized" two points for each of its parameters. Thus, a model with more parameters much achieve a higher maximum likelihood to offset this penalty factor.

7.4.2 Corrected Akaike Information Criterion

The corrected Akaike Information Criterion (AICc) is

$$AICc = AIC + \frac{2p(p+1)}{n-p-1}, \tag{7.11}$$

where p is the number of parameters and n is the sample size. The AICc reduces the risk of overfitting a small amount of data to a model with more parameters and approaches the AIC as the sample size increases.

$$\lim_{n\to\infty} AICc = AIC \tag{7.12}$$

7.4.3 Bayesian information criterion

The Bayesian information criterion (BIC) is an asymptotic result derived under the assumption that the data distribution is in the exponential family.

$$BIC = -2\ln L(\hat{\boldsymbol{\Theta}}) + p\ln n, \tag{7.13}$$

Equation (7.13) indicates that the BIC exacts a penalty proportional to the number of parameters p multiplied by the logarithm of the sample size n.

7.4.4 Sum of squares error

The sum of squares error (SSE), also know as the residual sum of squares (RSS) for failure time data is

$$SSE_{ft} = \sum_{i=1}^{n} (\hat{m}(t_i) - i)^2 \tag{7.14}$$

where $\hat{m}(t_i)$ is the number of faults predicted by an SRGM evaluated at the maximum likelihood estimate, while i is the actual number of faults detected, and n is the number of faults observed.

The sum of squares error (SSE) for failure count data is

$$SSE_{fc} = \sum_{i=1}^{n} (\hat{m}(t_i) - Y_i)^2 , \tag{7.15}$$

where

$$Y_i = \sum_{j=1}^{i} y_i \tag{7.16}$$

is the cumulative number of failures observed in the first i time intervals.

7.4.5 Mean square error

The respective expressions for the mean square error (MSE) of failure time and failure count data are

$$MSE_{ft} = \frac{1}{n} \sum_{i=1}^{n} (\hat{m}(t_i) - i)^2 \tag{7.17}$$

$$MSE_{fc} = \frac{1}{n} \sum_{i=1}^{n} (\hat{m}(t_i) - Y_i)^2 . \tag{7.18}$$

7.4.6 Root mean square error

The root mean square error (RMSE) is simply

$$RMSE = \sqrt{MSE}. \tag{7.19}$$

7.4.7 Predictive error

Predictive analogs of the SSE, MSE, and RMSE compare the predictions of a model to previously unobserved data. For example the predictive sum of squares error (PSSE) for failure time data is

$$PSSE_{ft} = \sum_{i=(n-k)+1}^{n} \left(\hat{m}(t_i) - i\right)^2, \tag{7.20}$$

where the maximum likelihood estimates of the model parameters are determined from the first $n-k$ failure times. A common choice of k is approximately 10% of the available data, so that $k/n \approx 0.10$.

7.4.8 Predictive ratio risk

The predictive ratio risk (PRR) for failure time data is

$$PRR_{ft} = \sum_{i=k+1}^{n} \left(\frac{\hat{m}(t_i) - i}{\hat{m}(t_i)}\right)^2. \tag{7.21}$$

Note that the term in the denominator penalizes underestimation of the number of faults more heavily than an overestimate.

7.4.9 Predictive power

The predictive power (PP) for failure time data is

$$PP_{ft} = \sum_{i=k+1}^{n} \left(\frac{\hat{m}(t_i) - i}{i}\right)^2. \tag{7.22}$$

7.5 Comparison of GOF measures

Figure 7.1 shows the three-dimensional plot of the failure times log-likelihood function of the Goel-Okumoto model for the SYS1 [Lyu, 1996] data set consisting of $n = 136$ failures.

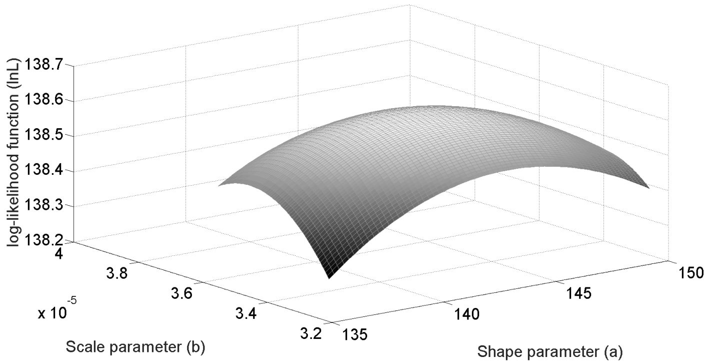

Fig. 7.1: Log-likelihood function

The maximum appears to occur for a between 140 and 145 and b between 3.3×10^{-5} and 3.5×10^{-5}, which can serve as initial estimates for an algorithm to identify the MLEs.

The parameter estimates of the failure times model are:

- Fault detection rate: $\hat{b} = 0.0000342038$
- Number of initial faults: $\hat{a} = 142.881$
- Number of faults remaining: 6.827

$$(\hat{a} - n = 142.881 - 136 = 6.827)$$

Figure 7.2 shows the fit of the GO model to failure times data along with the observed failure data.

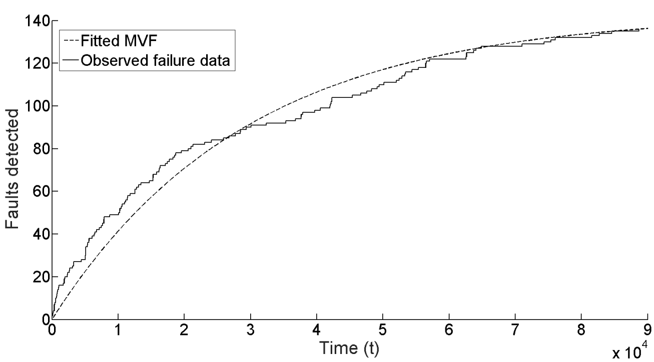

Fig. 7.2: Fit of GO model to failure times data

Figure 7.2 indicates that the Goel-Okumoto model underpredicts the data until approximately $25,000$ time units and overpredicts afterwards. However, in later stages, its predictions closely matches the data, hence it is widely used.

Figure 7.3 shows the fit of the GO model to failure counts data along with the observed failure count data for SYS1 data set, which was converted to failure counts by grouping faults into $n = 25$ intervals of one hour.

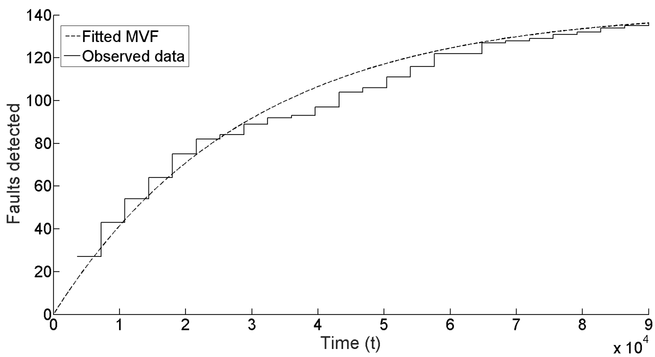

Fig. 7.3: Fit of GO model to failure counts data

Figure 7.3 indicates that the GO model under predicts the data between $30,000$ to $60,000$ time units. Apart from this, it matches the observed data closely.

The parameter estimates of the failure counts model are:

- Fault detection rate: $\hat{b} = 0.0000342879$
- Number of initial faults: $\hat{a} = 142.827$
- Number of faults remaining: 117.827

$$(\hat{a} - \sum_{i=1}^{n} k_i = 142.827 - 136 = 6.827)$$

Table 7.1 lists the AIC values for different SRGM.

Table 7.1 also ranks the models based on their respective AIC values. The Geometric model achieves the lowest AIC, indicating that it is the preferred model with respect to the AIC. The Weibull model comes close to the AIC of the Geometric model, but the other three models do not.

Figure 7.4 illustrates the fit of the GO and Weibull models for 90% of the SYS1 data set.

Table 7.1: Ranking of SRGM according to AIC

Model name	AIC	Ranking
Geometric	1937.03	1
Weibull	1938.16	2
Jelinski-Moranda	1950.53	3
Goel-Okumoto	1953.61	4
Delayed S-shaped	2075.15	5

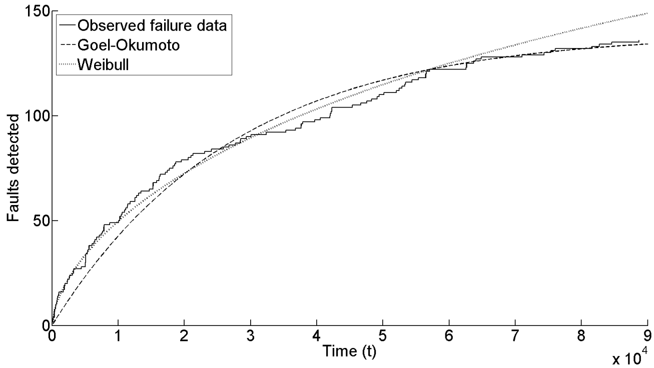

Fig. 7.4: Comparison of GO and Weibull model predictions

Figure 7.4 indicates that the GO model predicts the data better than the Weibull model, which is in contrast to the AIC assessment. Thus, no one measure of goodness of fit is sufficient to select a model and multiple measures should be considered.

Table 7.2 reports the numerical values of the PSSE for each of the three NHPP models, which shows that the error of the GO model predictions is far less than the Weibull model.

Table 7.2: Ranking of SRGM according to PSSE

Model name	PSSE	Ranking
Goel-Okumoto	23.17	1
Jelinski-Moranda	24.39	2
Weibull	74.94	3
Geometric	84.37	4
Delayed S-shaped	296.35	5

Table 7.3 reports the numerical values of AICc, BIC, SSE, PRR, and PP for GO and Weibull model.

Table 7.3: Other goodness of fit measures

Model name	AICc	BIC	SSE	PRR	PP
Goel-Okumoto	1953.70	1959.44	9346.13	0.00138149	0.00136917
Weibull	1938.34	1946.90	906.76	0.0325993	0.0370573

Chapter 8
Software Failure and Reliability Assessment Tool (SFRAT)

This chapter presents an open source software reliability tool to automatically apply methods from software reliability engineering, including visualization of failure data, application of software reliability growth models, inferences made possible by these models, and assessment of goodness of fit. The application and source code are available on the web. The quantitative assessments enabled by the tool should promote greater communication between government organizations acquiring software intensive systems and the defense contracting community.

8.1 Introduction

Software intensive systems have become the norm in defense acquisition. The size and complexity of these systems require extensive software development and testing, which can consume months or years to achieve acceptable failure rates. This complexity imposes both schedule and cost risks. Major delays in delivery to the warfighter have been experienced and trends are increasing at an alarming rate. This trend will continue to pose a serious threat to national security, if steps are not taken to accelerate defense acquisition. Despite the emphasis on capabilities to overcome adversaries, nonfunctional requirements are key to program success as they ultimately influence availability, maintainability, and lifecycle cost. Moreover, reliable software is critical to ensure systems are resilient to electronic and cyber warfare in hostile environments.

Software reliability growth models (SRGM) can be used to mitigate the cost and schedule risks identified above. SRGMs apply techniques from probability and stochastic processes to characterize reliability growth and failure rate reduction during test. Because many software, reliability, and system engineers are extremely busy it is often difficult to learn the underlying mathematical principles on top of a demanding work schedule. Therefore, some researchers have made efforts to transition these models and related decision making in the form of a software tool to automatically apply SRGM to facilitate quantitative software reliability test and evaluation. Previous automated tools for software reliability engineering include: SMERFS (Statistical Modeling and Estimation of Reliability Functions for Software) [Farr and Smith, 1984] which was incorporated into CASRE (Computer-Aided Software Reliability Estimation tool) [Lyu and Nikora, 1992] and SRATS (Software Reliability Assessment Tool on Spreadsheet) [Okamura and Dohi, 2013]. Although popular among the user community, the primary shortcoming of these previous tools is that they were implemented as desktop applications by a single researcher, which limited the models incorporated to those the individual researcher was familiar with.

To overcome the limitations of previous software reliability tools, this paper presents a free and open source software failure assessment and reliability tool (SFRAT) to promote quantitative assessment of software reliability and improved communication between organization acquiring software and those conducting developing and test. SFRAT is an application to estimate and predict the reliability of a software system during test and operation. Some of the questions it allows a user to answer about a software system during test are:

1. Is the software ready to release (has it achieved a specified reliability goal)?
2. How much more time and test effort will be required to achieve a specified goal?
3. What will be the consequences to the systems operational reliability if not enough testing resources are available?

SFRAT is implemented in the R statistical programming language, an open source environment for statistical computing and graphics that can be used on computers running Windows, Mac OSX, or Linux, with the user interface implemented with the Shiny. It is also accessible through a web-enabled framework web at *http://sasdlc.org/*http://sasdlc.org/ (System and software development life cycle). The code is accessible through the GitHub repository at *https://github.com/LanceFiondella/srt.core*, which can be downloaded and run using RStudio to use the application locally. In doing so, an organization can easily perform information assurance assessment of the code prior to use on sensitive failure data.

The architecture will enable incorporation of existing software reliability models into a single tool, enabling more systematic comparison of models than ever before. Presently, inter failure time, failure time, and failure count data formats are supported. Implemented functionality includes: two trend tests for reliability growth, two failure rate [Lyu, 1996], Jelinski-Moranda and geometric, and three failure counting models, Goel-Okumoto [Goel, 1985], delayed S-shaped [Yamada et al., 1986], and Weibull [Yamada and Osaki, 1983] models as well as two measures of goodness of fit. Inferences such as the time to achieve a target reliability or detect a specified number of additional failures have also been implemented. The free and open source nature of the tool architecture enables additional models and goodness of fit measures to be included.

8.2 SFRAT: Graphical User Interface

SFRAT can be accessed either by downloading the source code from Github and run using RStudio or through the web instance. Once the application is launched, the initial SFRAT screen (Figure 8.1) is shown.

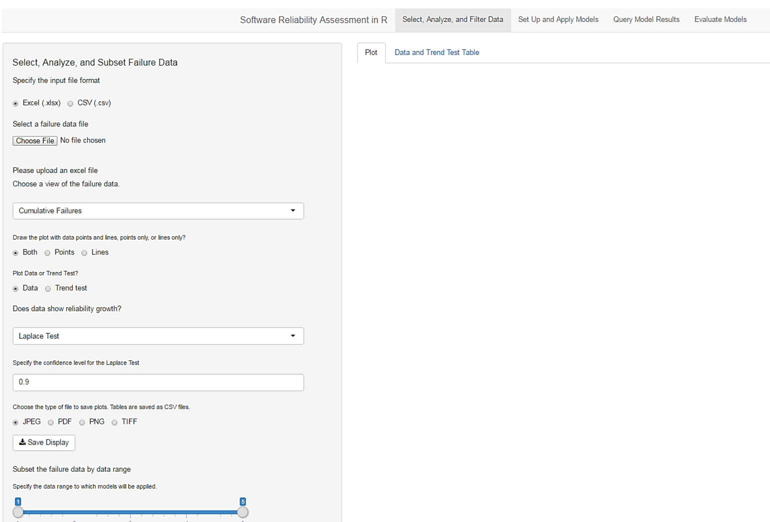

Fig. 8.1: Initial View within SFRAT

Figure 8.1 shows that SFRAT divides the workflow into the following four primary subtasks:

- **Select, Analyze, and Filter Data**: Open a failure data history file and determine whether the data exhibits reliability growth that would justify model application.
- **Set Up and Apply Models**: Apply one or more software reliability models and view the results including reliability growth.
- **Query Model Results**: Query applied models to answer the following questions:
  - How many additional failures will be observed if testing is continued for an additional amount of time?
  - How much additional time is required to observe a given number of additional failures?
  - How much additional testing time will be needed to achieve a specified reliability goal (or has that goal already been achieved)?
- **Evaluate Models**: Compare the models with one another to determine which one is the most likely to characterize the data well and provide accurate predictions.

8.2.1 Input Data

SFRAT accepts input data in either excel (.xlsx) or CSV (Comma Separated Values) file format. An input data file contains the failure data collected during the software testing phase in one of the following formats:

- Failure time data (FT): records the time of individual failures. A vector of individual failure times $\mathbf{T} = \langle t_1, t_2, \ldots, t_n \rangle$, where, n is the number of faults observed.
- Inter-failure data (IF): records the length of time between two successive failures. It is equivalent to the time between $(i-1)^{st}$ and i^{th} failure, defined as $t_i - t_{i-1}$.
- Failure count data (FC): records the length of each interval and number failures observed within that interval
  $\langle \mathbf{T}, \mathbf{K} \rangle = \langle (t_2, k_2), \ldots, (t_n, k_n) \rangle$
  where, t_i is the time at which the i^{th} interval ended and k_i is the number of faults detected in that interval.

Any of these types of failure data can be read by SFRAT. Examples of each data format are shown in Table 8.1 below.

Table 8.1: Inter-failure, failure times, and failure counts data formats in Excel file

Inter-failure		Failure times		failure counts		
FN	IF	FN	FT	T	FC	CFC
1	3	1	3	1	6	6
2	30	2	33	3	1	7
3	113	3	146	4.5	1	8
4	81	4	227	6	0	8
5	115	5	342	7	1	9
6	9	6	351	8	3	12
7	2	7	353	9	0	12
8	91	8	444	10.5	5	17
9	112	9	556	11	6	23
10	15	10	571	12	1	24

Table 8.1 shows part of a set of input data in the form of times between successive failures, which are inter-failure times, failure times, and failure counts data labeled as "IF" , "FT" , and "FC". The column labeled "FN" indicates the failure number for the failure time and inter-failure data, whereas the column labeled "T" denotes the time interval for the failure count data and "CFC" denotes the cumulative failure count, which is the total number of failures observed over time. The input data should be entered in one of the three formats shown in Table 8.1. The tool then automatically converts the data to failure time and inter-failure data formats.

Once the data file has been prepared, the file format can be specified under the option "*Specify the input file format*". Then, the user can browse and specify the file for upload by clicking on "*Choose File*" option under "*Select a failure data file*" option.

Figure 8.2 and 8.3 shows SFRAT before and after the input file upload. Once the data upload is complete (Figure 8.3), the user can select a specific data set under the "*Choose Sheet*" option if the excel file contains more than one data set. The file model_data.xlsx available from (http://sasdlc.org/lab/projects/srt.html) contains 31 data sets from the Handbook of Software Reliability, which consist of 10 failure times and 21 failure counts data sets. These data sets and the web version of the tool (http://sasdlc.org/) have been provided so that users can experiment without having to download and install the tool or possess and format their own failure data to learn how to use the tool. For the sake of illustration, we have selected the SYS1 data set which will serve as a running example throughout this document. The data set has been used in many research papers and is therefore used here to familiarize readers who may

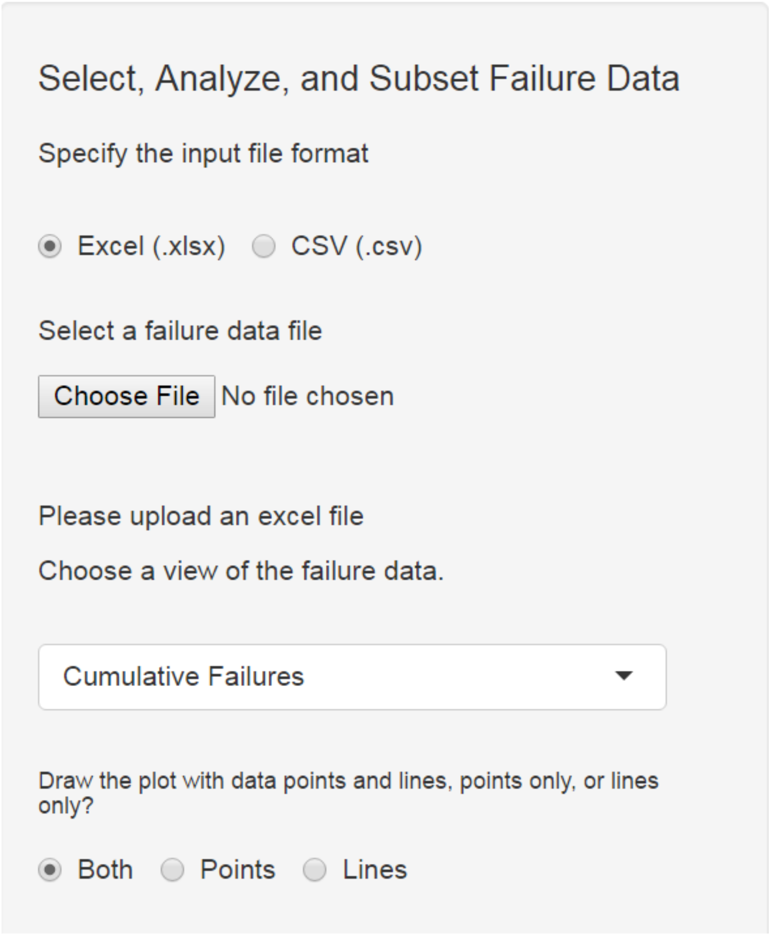

Fig. 8.2: Before input data file upload

ultimately choose to explore previous published papers to understand the various models and applications of software reliability growth models.

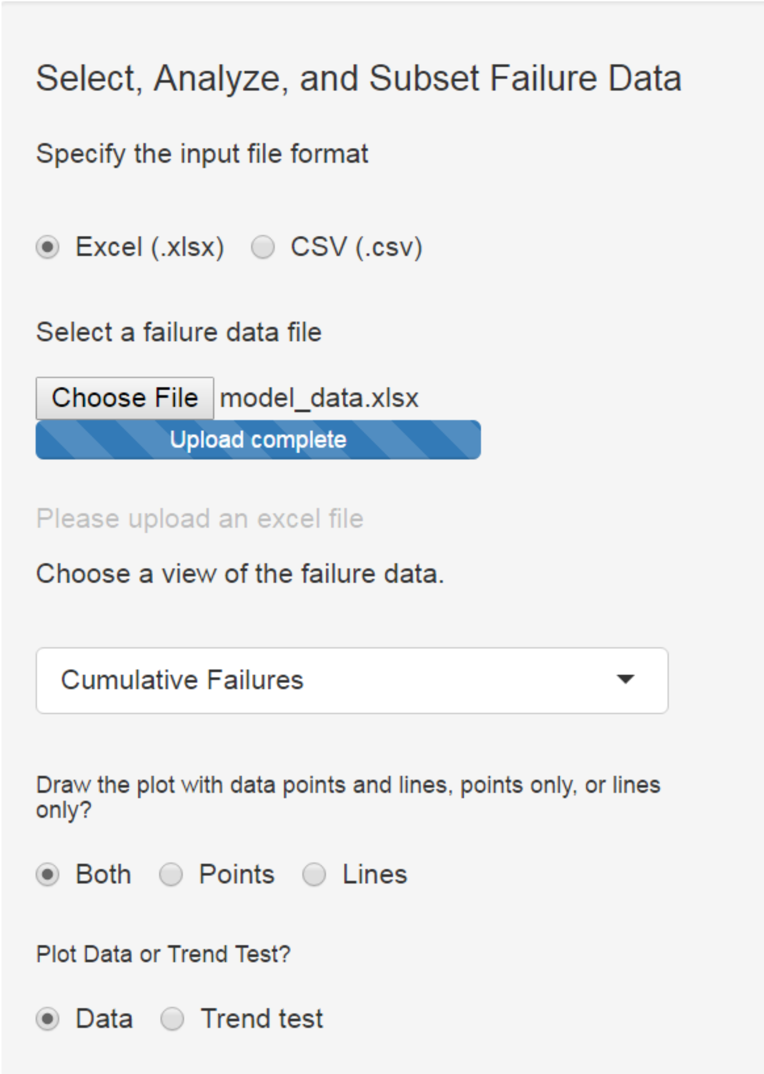

Fig. 8.3: After input file upload

8.2.2 *Tab 1: Select, Analyze, And Filter Data*

This tab allows the user to observe the input data and analyze the reliability growth through trend tests. This tab provides graphical and tabular views of both the data and these trend tests.

8.2.2.1 Open Data File

Upon successful data upload as shown in Figure 8.3, a plot of cumulative failures will be displayed by default. However, the data can be viewed in different formats such as cumulative failures, times between failures, and failure intensity by selecting options from the "*Choose a view of the failure data*" pull-down menu.

Upon successful data upload as shown in Figure 8.3, a plot of cumulative failures will be displayed by default. However, the data can be viewed in different formats such as cumulative failures, times between failures, and failure intensity by selecting options from the "*Choose a view of the failure data*" pull-down menu.

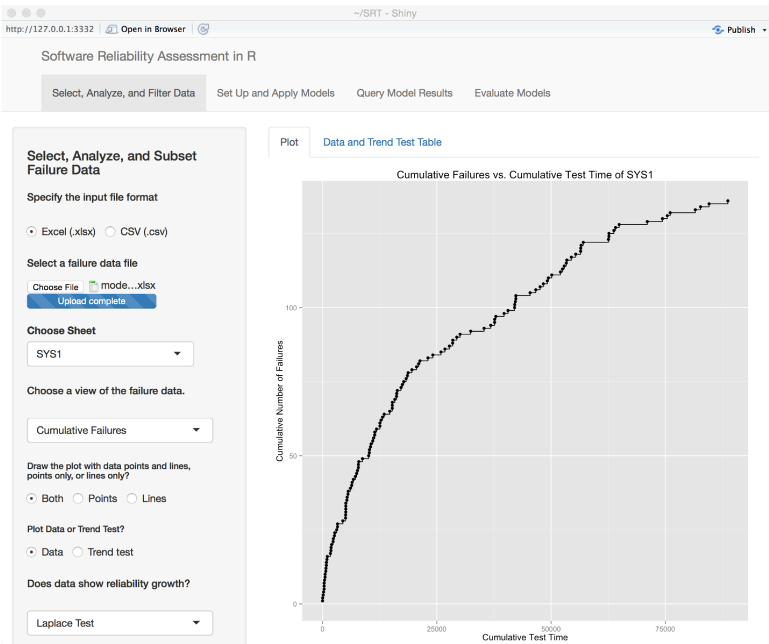

Fig. 8.4: Cumulative failure data plot for the SYS1 data set

Figure 8.4 shows the cumulative failures, plotting the total number of failures as a function of testing time. In addition to the graphical view, clicking on the "*Data and Trend Test Table*" tab above the plot in Figure 8.4 displays the input data in tabular form.

Figure 8.5 shows the corresponding tabular view of the SYS1 data, including the failure number, times between failures, and cumulative failure times.


Plot | Data and Trend Test Table

Show 10 entries Search:

	Failure.Number	Times.Between.Failures	Failure.Time
	All	All	All
1	1	3	3
2	2	30	33
3	3	113	146
4	4	81	227
5	5	115	342
6	6	9	351
7	7	2	353
8	8	91	444
9	9	112	556
10	10	15	571

Showing 1 to 10 of 136 entries Previous 1 2 3 4 5 … 14 Next

Fig. 8.5: Tabular view of the SYS1 data set

Figure 8.6 shows the times between failures plot of the SYS1 data. Figure 8.6 indicates that the times between failures at the beginning of testing is smaller because more faults were identified earlier, while there are larger times between failures as testing progresses, suggesting reliability growth.

Figure 8.7 shows the failure intensity plot of the SYS1 data. Figure 8.7 shows the number of failures observed per unit of testing time, which is higher at the beginning of testing and decreasing as testing progresses. These plots provide alternative views of the input data so that the user can analyze the data from multiple perspectives.

Options allow the user to decide whether the plot display both data points and lines, lines only, or data points only. To do this, use the radio buttons labeled "*Draw the plot with data points and lines, points only, or lines only?*". The default is both data points and lines.

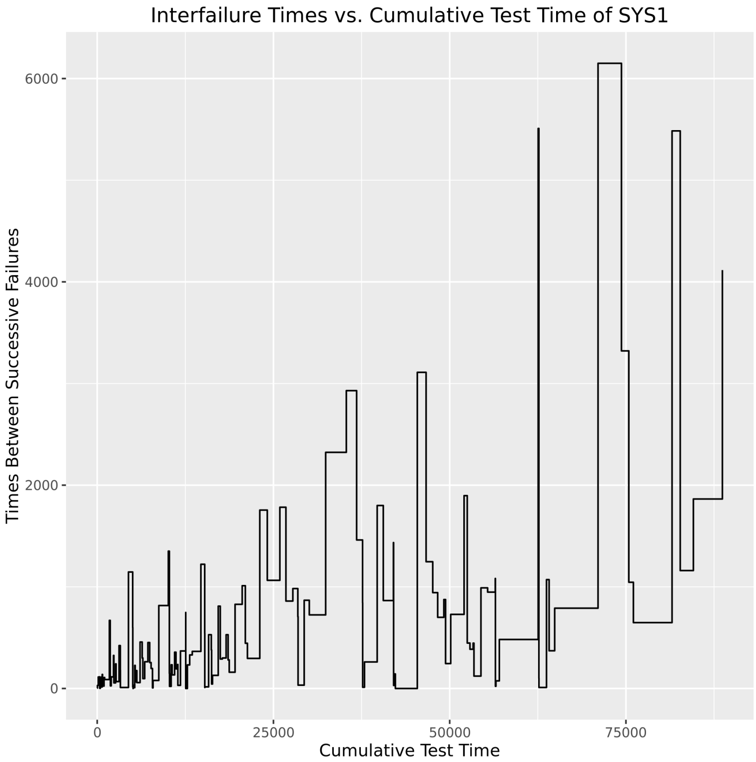

Fig. 8.6: Times between failure plot

8.2.2.2 Analyzing Failure Data for Trends

Before applying one or more models to the failure data, it is important to check if the failure data exhibits reliability growth. Since software reliability models assume that the reliability of the software system increases during testing as new defects are identified and removed, reliability growth in the data is necessary for models to characterize the data well and enable accurate predictions. Toward this end, SFRAT implements two trend tests to determine whether the failure data exhibits reliability growth, including the Laplace trend test and running arithmetic average.

Figure 8.8 and 8.9 shows the trend test radio button selected along with a plot of the Laplace Trend Test for the SYS1 data set.

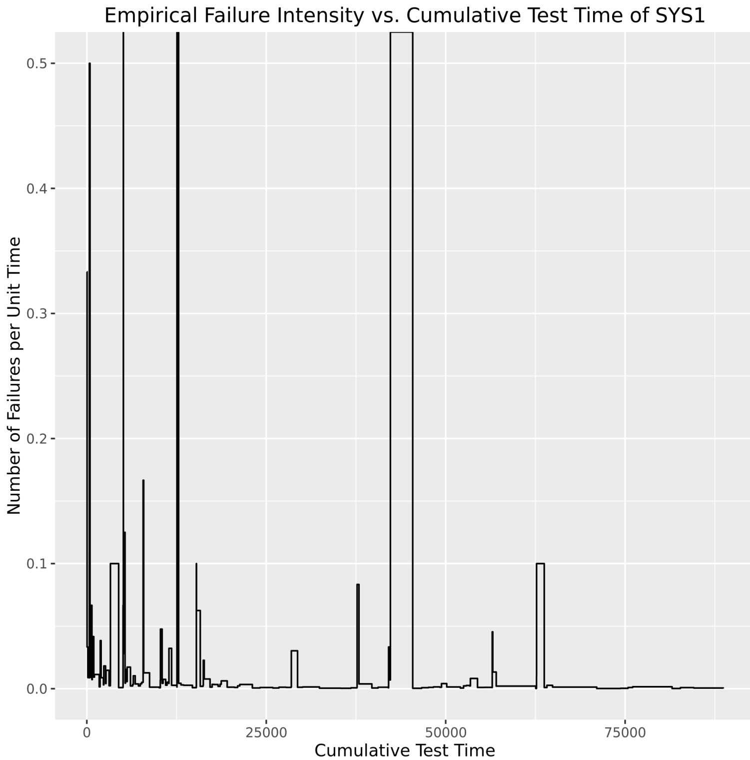

Fig. 8.7: Failure intensity plot

Figure 8.9 shows a zoomed in view of the left side of Figure 8.8 to clarify how to switch from the data to trend test by clicking on the radio button labeled "*Trend test*" under the "*Plot Data or Trend Test?*". One of the two trend test can be selected from the pull-down menu under the "*Does data show reliability growth*" label. The user can also specify the desired level of significance by entering a confidence value between 0 and 1 in the input area under the "*Specify the confidence level for the Laplace Test*" label in Figure 8.8.

The Right side of Figure 8.8 shows the default trend test plot for the SYS1 data, which is the Laplace trend test at the 90% confidence level indicated by the red horizontal line. In the Laplace trend test, the value of the test statistic is computed for each prefix of the failure data. A decreasing trend in the value of the test statistic indicates that the data exhibits reliability growth. If an intermediate value of the test statistic is

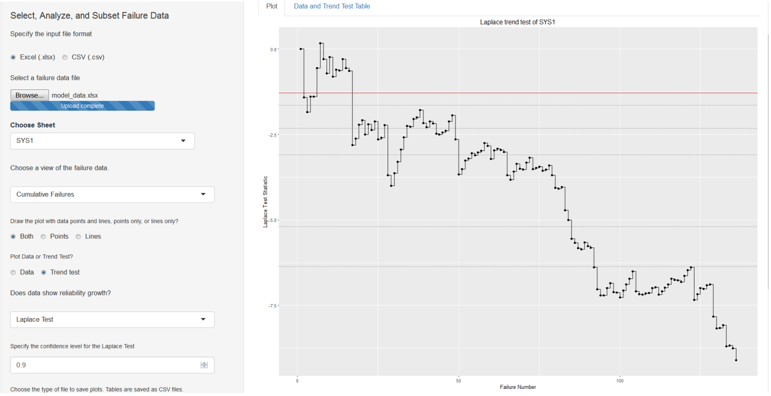

Fig. 8.8: Trend tests options along with Laplace trend test plot

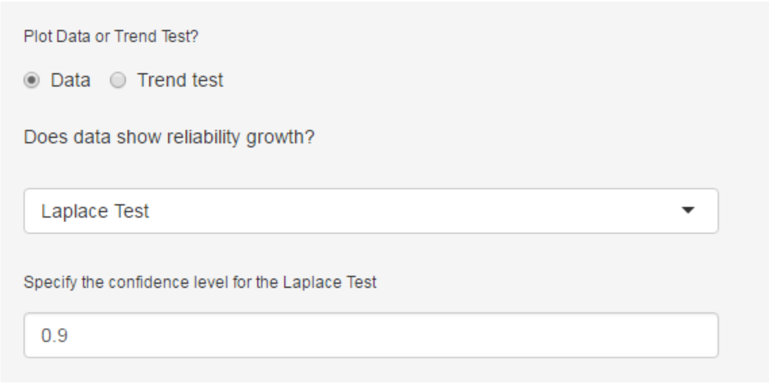

Fig. 8.9: Trend test options

less than the specified confidence level, this means that the data exhibits reliability growth. For the SYS1 data, the plot is decreasing indicating reliability growth at the 90% confidence level at all points beyond the 17th failure. Thus, it is appropriate to apply reliability growth models after the data has begun to exhibit reliability growth with a desired level of confidence.

Figure 8.10 shows the corresponding tabular form of the Laplace test results.

Figure 8.10 shows the Laplace test statistic value with the corresponding failure data. The table only lists times between failures because this trend test uses inter-failure data to compute the test statistics.

Plot | Data and Trend Test Table

Show 10 entries Search:

	Failure.Number	Times.Between.Failures	Laplace.Test.Statistic
	All	All	All
1	1	3	0.0000000
2	2	30	-1.4171325
3	3	113	-1.8455060
4	4	81	-1.3964758
5	5	115	-1.3927309
6	6	9	-0.5583275
7	7	2	0.1722696
8	8	91	-0.2919402
9	9	112	-0.7159030
10	10	15	-0.2315468

Showing 1 to 10 of 136 entries Previous 1 2 3 4 5 … 14 Next

Fig. 8.10: Tabular view of Laplace test statistical values for the SYS1 data

If, on the other hand, a data set does not exhibit reliability growth, it is recommended that additional testing be performed to ensure that reliability growth is achieved. If the user applies the model to data that does not exhibit reliability growth and subsequently makes predictions based on the model results, there is a risk that these predictions will be inaccurate and either over or under estimate the time to remove remaining faults and the corresponding time to achieve a desired reliability. Therefore, it is important to determine that a failure data set exhibits reliability growth prior to applying models and making predictions.

Figure 8.11 shows the Laplace trend test of the J4 data set which does not exhibit reliability growth.

In fact, the increasing trend suggest that the system exhibits reliability deterioration. In such cases, additional testing may be required. Investigations may also be conducted to determine the cause of the apparent reliability decrease and to make appropriate changes to the development process and schedule. Programs that fail to achieve reliability growth are at greater risk of schedule and cost overruns as well as cancellation. Thus, assuming tests are representative of required functionality, trend tests provide government organization with high level oversight regarding program progress. They also provide contractors with a quantitative mechanism to demonstrate that they are meeting reliability targets on or

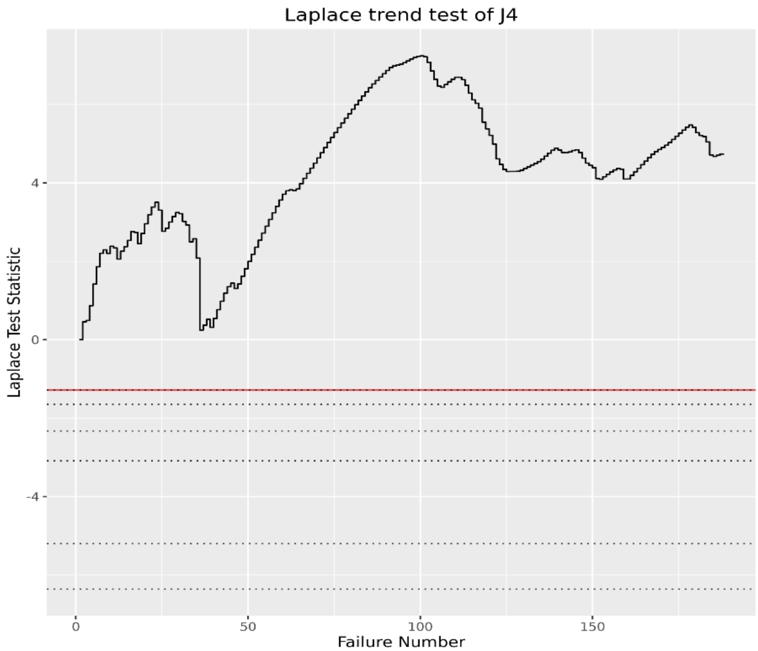

Fig. 8.11: Laplace Trend Test Plot for J4 data

ahead of schedule, creating the potential for positive incentives that reward attention to system reliability.

The second trend test is the running arithmetic average, which plots the average of the times between failures. A plot exhibiting a positive slope indicates that the times between failures are increasing, suggesting reliability growth and that it would be suitable to apply software reliability models to the data. In contrast, a plot exhibiting a negative slope would indicate the times between failures are decreasing, which would suggest that the software reliability is decreasing. In such situations, it would not be appropriate to apply software reliability models due to risk of poor model fits and predictions as noted above. Note that there are no confidence bounds associated with this trend test as there are for the Laplace test. It only provides point estimates of the rate at which the average time between failures is increasing.

Figure 8.12 shows a plot of the running arithmetic average test for the SYS1 data set.

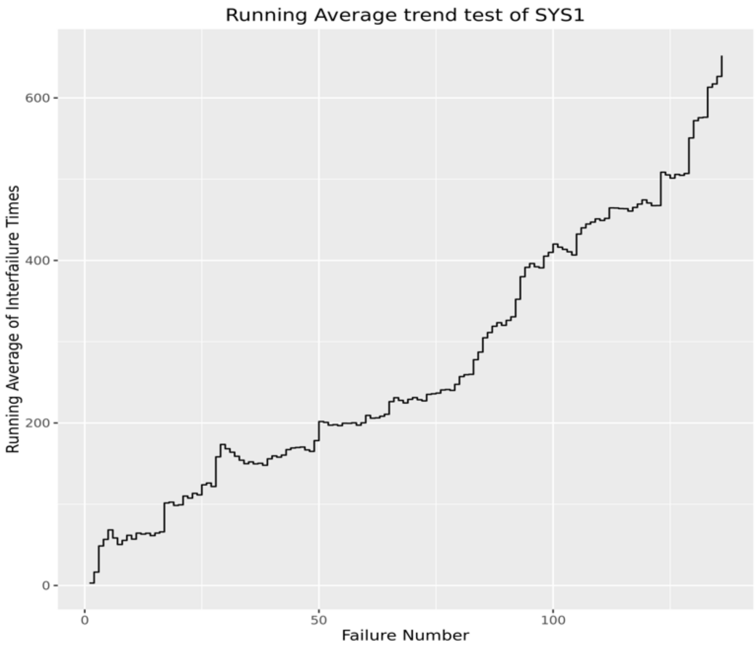

Fig. 8.12: Running arithmetic average test plot

Figure 8.12 exhibits an increasing average, indicating that the average time between failures is increasing, which corresponds to reliability growth.

To return to the failure data view shown in Figure 8.4, click the "*Data*" radio button under the "*Plot Data or Trend Test*" label shown in Figure 8.8.

8.2.2.3 Subset Failure Data

A user can select a subset of the data to work with instead of the entire data set. For example, it might be the case that only a portion of the testing represented by the data is appropriate for reliability testing or the user might want to see which model would have predicted best to decide which model to predict with now.

Figure 8.13 shows a double-ended sliding bar that can be used to subset the data.

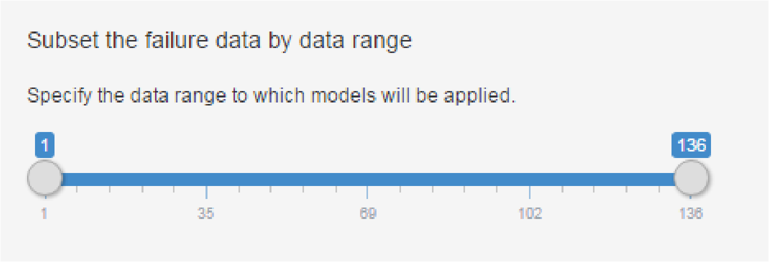

Fig. 8.13: Subset failure data

Figure 8.13 shows the data subset option for the SYS1 data, which has 136 failures. To select a starting point for the subset, click and drag to the right the round control on the left of the slider. To select the end point for the subset, click and drag to the right round control to the left.

Figure 8.14 shows an example of a subset selection.

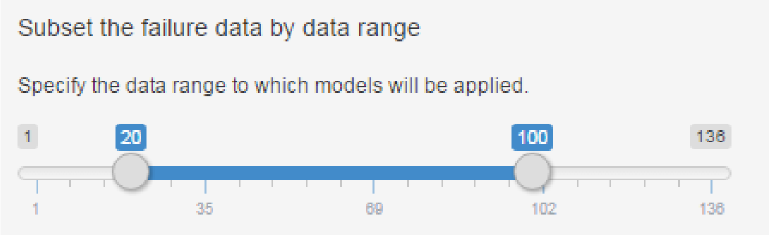

Fig. 8.14: Subset selection: Failures 20-100

Figure 8.14 shows the data subset restricted to the range $20-100$. Once this has been performed, the plot of the data will be updated to only show the subset selected.

A data subset must include 5 or more failures. If fewer than five failures are selected, SFRAT will display messages that too few failures were selected and automatically adjust the endpoints of the slider so that the subset includes five failures.

8.2.2.4 Saving Data and Trend Test Tables

To save a plot or data table, click the "*Save Display*" button near the bottom of the controls on the left side of the window. The data or trend test plot will be saved in either JPEG, PDF, PNG, or TIFF format.

Figure 8.15 shows the types of images that a plot or table can be saved as.

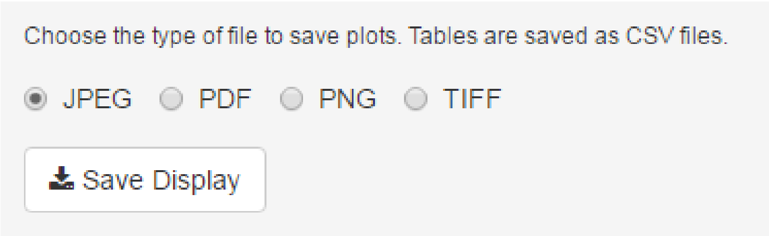

Fig. 8.15: Save data plots or tables

Figure 8.15 shows the available options for saving the plots, which can be inserted into a document or report. Tabular displays of failure data or trend test results can also be saved as a comma separated value (CSV) file by clicking on any of the save options while viewing the table.

8.2.3 Tab 2: Set Up And Apply Models

After analyzing the data and trends, the next step is to apply software reliability models to the data and view the results by selecting the second tab shown in Figure 8.1 - "*Set Up and Apply Models.*" There are two groups of controls in the sidebar on the left side of the window. The first group configures and selects models, while the second group controls the way model results are displayed. There is also a "*Save*" button at the bottom of the controls, which works in a manner similar to the "*Save Display*" button described in Section 2.2.4.

8.2.3.1 Configuring and Applying Models

The controls allow the user to specify the number of failures to predict and then select one or more models of the five available models, including the Jelinski-Moranda and Geometric failure rate models, and Goel-Okumoto, Weibull, and delayed S-shaped failure counting software reliability growth models.

Figure 8.16 shows the controls available under configure and apply models.

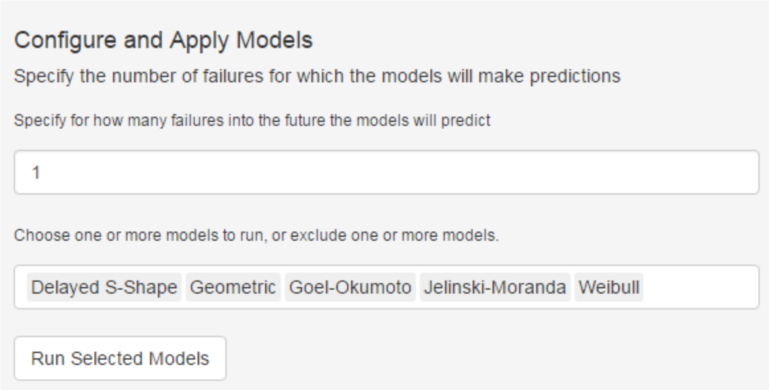

Fig. 8.16: Select. Configure, and Apply Models

Figure 8.16 shows the option that allows the user to specify the number of failures to predict with the models selected, which can be performed by entering a whole number greater than 0 into the area labeled "*Specify for how many failures into the future the models will predict.*" In Figure 8.16, the example includes prediction of 1 failure, which is a default value, into the future by applying all the five models from the pull-down menu "*Choose one or more models to run, or exclude one or more models*" and clicking "*Run Selected Models*" for the SYS1 data, which was chosen on Tab1.

8.2.3.2 Select Model Results for Display and Control the Display Type

Once one or more models have been run, plots of the model results and the way they are displayed can be specified using the controls shown in Figure 8.17.

Figure 8.17 shows a pull-down menu labeled "*Choose one or more sets of model results to display*" with all the model names that are applied. Select one or more sets of model results to be plotted. Using the "*Choose a plot*

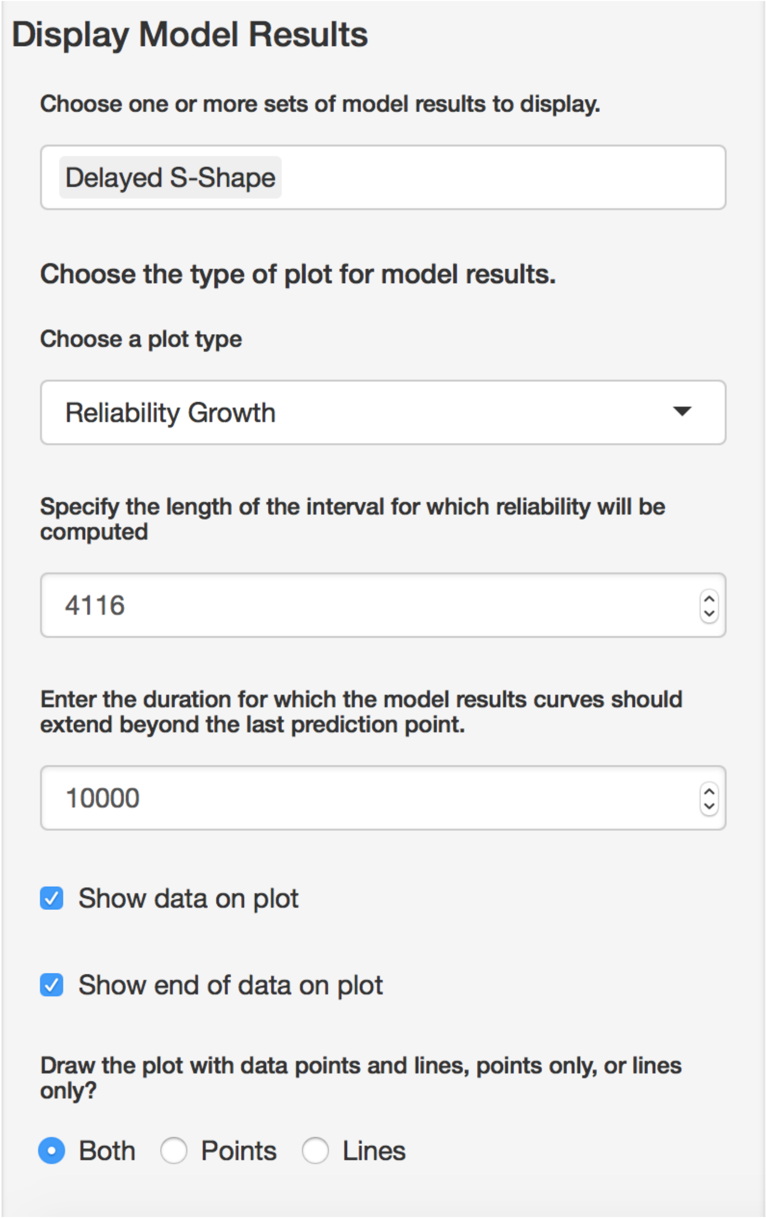

Fig. 8.17: Control model results display

type" menu shown in Figure 8.17, the model results can be viewed as cumulative failures vs. elapsed time (the default view), failure intensity, times between failures, or reliability growth. If view reliability growth is chosen, an input area labeled "*Specify the length of the interval for which reliability will be computed*" appears in the display controls. The input is simply the amount of time for which the user would like to estimate the probability that the system will run without experiencing a single failure. The default value of this input is the last non-zero value of time between successive failures in the data set being analyzed, which is in the case of the SYS1 data set is 4116. This can be changed to any value greater than 0. When viewing the result curves, the input control labeled "*For how much time should the model results curve extend beyond the last prediction point?*" is used to specify how many time units beyond the end of the last predicted failure the curves will be drawn. The predictions can be shown as data points, curves drawn according to the model result equations, or both. The default value is $10,000$ time units.

The controls at the bottom of Figure 8.17 determine the appearance of the model results plot. Checking the "*Show data on plot*" box displays the failure data on the same plot as the model results. This allows qualitative visual comparison of the model results with the original failure data. Checking the "*Show end of data on plot*" box, displays a vertical black line passing through the final data point in the set of failure data being analyzed. This enables the user to easily distinguish the model predictions (right of the line) and estimates (left of the line) regarding the previous failure history. Finally, the radio buttons at the bottom of the controls plot the model results as both data points (each point represents a failure) connected by a curve drawn according to the model equations, data points only, or curves only. The default is to draw both data points and curves.

Figure 8.18 shows an example of all the model fits, the SYS1 data set, and each models predictions of the times to observe the next 10 failures into the future.

Figure 8.18 shows the fit of all five models to the data as well as the SYS1 data, which is displayed as a red staircase plot in the figure. The curve extends $20,000$ time units beyond the end of testing. This visual display indicates that the Geometric and Weibull model most closely match the SYS1 data. To zoom into an area of a plot, use the "*paint and double-click*" mechanism. First, click and drag over the region of the plot to zoom into to define ("*paint*") a rectangle, then double-click inside that rectangle.

Figure 8.19 shows an enlarged area of Figure 8.18.

Fig. 8.18: Default model results display showing all models

To return to the original plot, simply double click anywhere on the plot and the display will zoom back out to the original view.

Figure 8.20 shows the corresponding tabular form of the model results.

As shown in Figure 8.20, the drop-down menu at the top of the table labeled "*Choose one or more sets of model results to display*" is used to view the desired model results. For each model selected, the model parameters, the cumulative time, the models estimates, and predictions of the cumulative number of failures at that time, the estimated and predicted times between failures, the estimated and predicted failure intensity, and the estimated and predicted reliability are shown. It is possible to scroll up and down in the table as well as left and right.

After gaining experience using SFRAT and working with several failure data sets, users may encounter situations in which one or more models are unable to produce results for some data sets. One of the types of data sets that can cause this to happen is a data set that does not exhibit reliability growth according to the trend tests. For example, refer to the discussion of the J4 data set in Section 2.2.2). In this case, only the names of the models that successfully ran to completion will appear in the pull-down menu labeled "*Choose one or more sets of model results to display*" in Figures 8.17, 8.18, and 8.19.

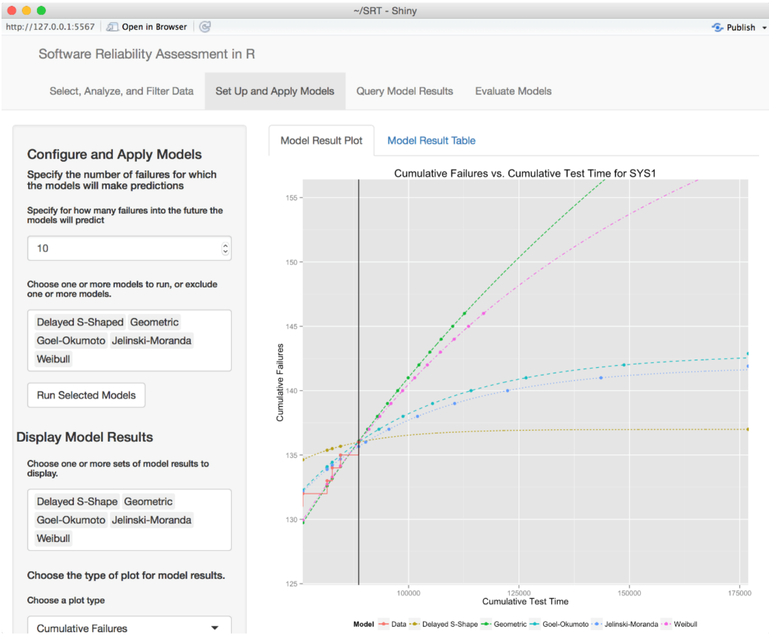

Fig. 8.19: Zooming into a model result plot

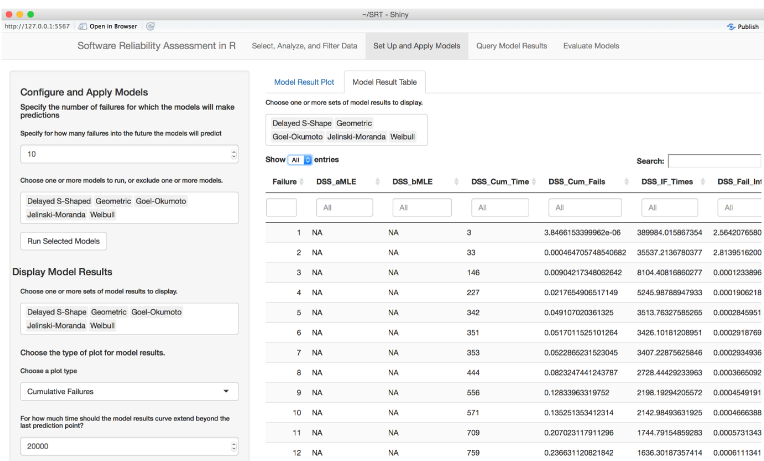

Fig. 8.20: Tabular display of model results

Figure 8.21 shows an example of results for J4 data, where all 5 models were selected to run, but the Jelinski-Moranda model does not appear in the list of models that can be displayed because its parameter estimates failed to converge to values that would enable prediction.

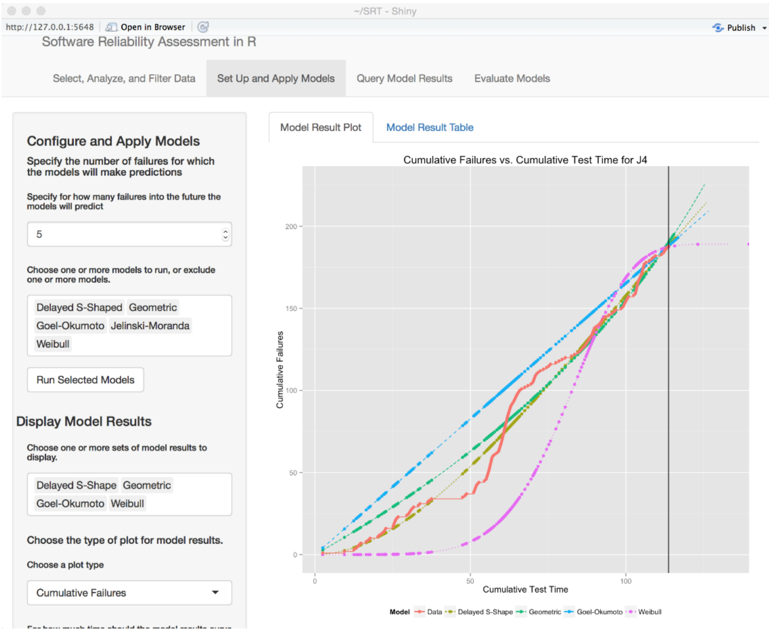

Fig. 8.21: Data set for which one model did not complete successfully

Figure 8.21 shows a plot of the J4 data set and the cumulative failures for each of the model except the Jelinski-Moranda because the model did not converge, so it is excluded from the results list. It can also be seen that the other model fits to not closely match the data set in red, providing a second warning that the models do not characterize the data well and that the user should be careful as making predictions about future failures is likely to be inaccurate.

To save the model results plot or table, click the "*Save*" button at the bottom of the controls on the left side of the window. When viewing the model results plot, clicking the "*Save*" button will save that plot in the format specified with the radio buttons shown in Figure 8.15. When viewing a tabular display of model results, clicking the "*Save*" button will save as a comma separated value (CSV).

8.2.4 Tab 3: Query Model Results

The plots and tabular displays of model results can provide a good overview of what the different models predict for the future failure behavior of the system being analyzed. However, there are some details that are difficult to see in the plots and tables, so SFRAT allows detailed queries of the models to answer the following questions:

- How many more failures will be observed in a given amount of time?
- How long will it take to observe a given number of failures?
- How much more testing time will be needed to obtain a given reliability for a specified operating time?

Figure 8.22 shows the controls on the "*Query Model Results*" tab to answer these questions.
In Figure 8.22, the pull-down menu at the top of the controls allows the user to choose one or more sets of model results to query. For this example, all 5 models were run and all 5 completed successfully on the SYS1 data. If a model does not complete successfully, the name of that model will not be displayed in the pull-down menu. The second control allows the user to determine how much more time will be needed to observe a given number of failures. For this example, the time to observe the next 5 failures for each of the models sought. The third control allows the user to determine how many more failures we will be observed in a given amount of future testing time. Since the models are a continuous approximation to a discrete counting process, the number of failures predicted in response to this query will not necessarily be a whole number. Finally, the last two controls allow the user to determine how much more time will be needed to obtain a specified reliability. This requires entering values in the input area labeled "*Specify the desired reliability*" and the operational time (the input labeled "*Specify the length of the interval for which reliability will be computed*".

Figure 8.23 shows responses to the queries made in Figure 8.22.
Figure 8.23 shows the response to the query in the 6-column table. The first column identifies the row of the table. The second column identifies the model whose results are shown in the next 4 columns. Note that there is a blank row between models. The third column tells you how much more time is needed to achieve a given reliability for a specified operating time. For this example, the default value (0.9) for the reliability was used. The operating time is 4116 time units, which is the length of the last non-zero value of times between failures in the data set being analyzed. If a models results indicate that the reliability objective has already been achieved, column 3 displays the message "*R = 0.9 achieved.*"

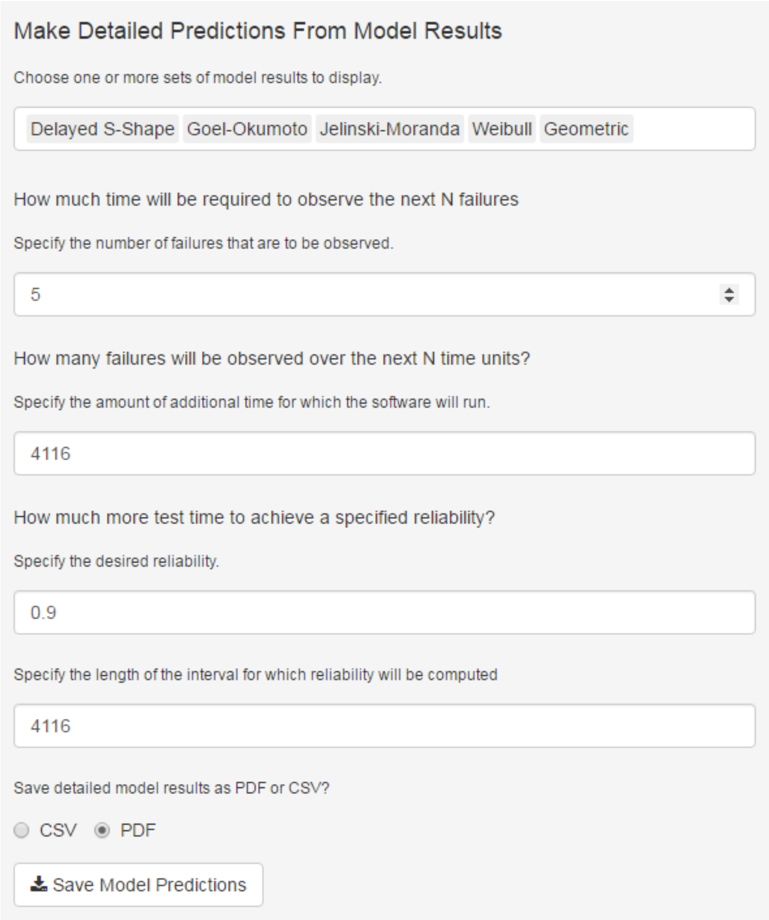

Fig. 8.22: Detailed model queries

Column 4 reports how many more failures are expected in specified time period. Finally, columns 5 and 6 indicate how much more testing time will be needed to observe the next N failures, where *N* ranges from 1 to the number of failures specified in the "*Specify the number of failures that are to be observed*" input area. In this example, notice that the Delayed S-Shape model predicts that fewer than 5 failures remain to be discovered. Thus, only the required number of rows are displayed. If a model indicates that the amount of time needed to discover a failure is

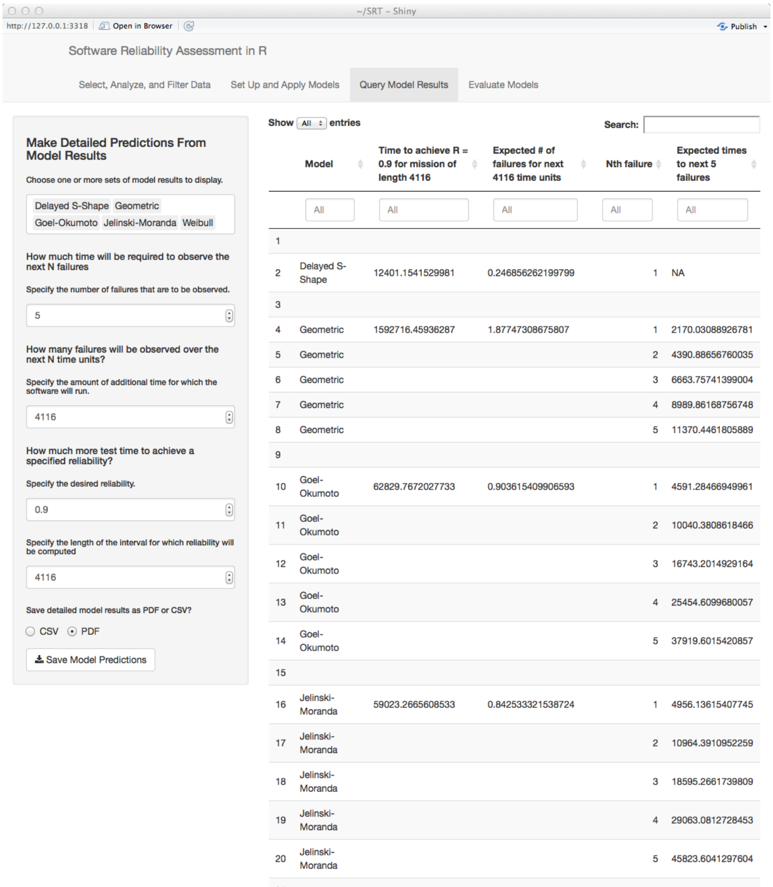

Fig. 8.23: Detailed model query response

infinite, the value "*NA*" is displayed in column 6, as is shown for the Delayed S-Shape model in this example.

Save the results of the query as either a CSV or PDF file by clicking the "*Save Model Predictions*" button. Clicking this button will bring up a dialog box in which to specify the name of the file to save and the folder in which to save it. The type of file the query results are saved to is controlled by the radio buttons just above the "*Save Model Predictions*" button. The default file type is PDF. Results can also be saved as a CSV file.

8.2.5 Tab 4: Evaluate Models

After applying one or more models to a set of failure data, the user can assess which model or models produce better predictions. This version of SFRAT includes two methods to evaluate the performance of the models to identify those that will provide better predictions the Akaike Information Criterion (AIC) [Akaike, 1974] and Predictive Sum of Squares Error (PSSE) [Pham].

Figure 8.24 shows the tabular format of the goodness-of-fit measures.

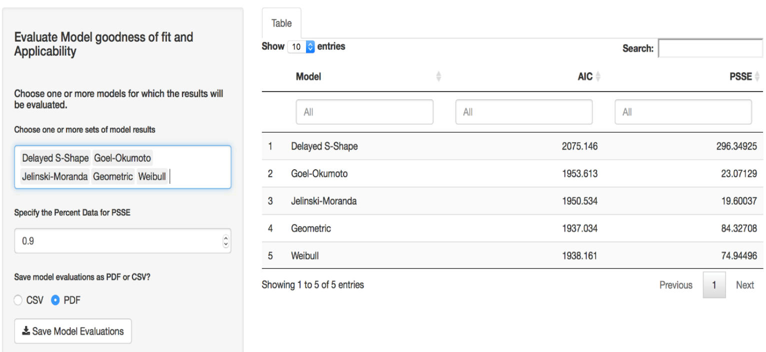

Fig. 8.24: Evaluating Model Performance

The controls on the left side of Figure 8.24 allow the user to select the models they want to compare, configure the PSSE analysis, and save the model evaluations as a CSV or PDF file. The pull-down menu labeled "*Choose one or more models for which the results will be evaluated*" identifies those models that completed successfully. You can choose one or more models to compare from this menu. To perform the PSSE evaluation, choose a subset of the data to apply the selected model and use the remainder of the data for the evaluation. The square of the differences between the models predictions for the remainder of the data are then computed and summed for each of the models chosen. A lower PSSE value indicates less distance between the models predictions and the actual data, which suggests a better-performing model.

The AIC is an information theoretic measure of the relative likelihood that the data is well characterized by a model. A lower AIC value indicates that a model is more likely to characterize the data well. The AIC identifies the model that minimizes the information lost. A model that minimizes the information loss is more likely to perform better and therefore may be a more appropriate model. The AIC indicates how much more likely it is that model 2 minimizes the information loss than model 1. Note that AIC does not indicate how well the model results fit the data in the geometric sense. It is important to note that if none of the model results fit the data well, the AIC will not provide any warning to the user.

As with detailed model queries, save the results of model evaluations as either a CSV or PDF file by clicking the "*Save Model Evaluations*" button. Clicking this button will bring up a dialog box in which to specify the name of the file to save and the folder in which to save it. The type of file to which the evaluation results are saved is controlled by the radio buttons just above the "*Save Model Evaluations*" button. The default file type is PDF; The user can also save the model evaluations as a CSV file.

References

H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automat. Contr.*, AC-19(6):716–723, dec 1974.

H Ascher and Harry Feingold. *Repairable systems reliability: Modeling, inference, misconceptions and their causes*. Marcel Dekker, 1984.

Elsayed A Elsayed. *Reliability engineering*, volume 88. John Wiley & Sons, 2012.

W. Farr and O. Smith. Statistical modeling and estimation of reliability functions for software (SMERFS) users guide. Technical Report NAVSWC TR-84-373, Revision 2, Naval Surface Warfare Center, Dahlgren, VA, 1984.

A. Goel. "Software reliability models: Assumptions, limitations, and applicability". *IEEE Trans. Software Eng.*, 11(12):1411–1423, dec 1985.

A. Goel and K. Okumoto. Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures. *IEEE Trans. Rel.*, R-28(3):206–211, aug 1979.

Zygmunt Jelinski and P Moranda. Software reliability research. In *Statistical computer performance evaluation*, pages 465–484. Elsevier, 1972.

David Kleinbaum, Lawrence Kupper, Azhar Nizam, and K Muller. *Applied Regression Analysis and Other Multivariable Methods, 4th ed.* Australia: Thomson Learning Inc. Brooks/Cole, 2008.

L. Leemis. *Reliability: Probabilistic Models and Statistical Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

M. Lyu, editor. *Handbook of Software Reliability Engineering*. McGraw-Hill, New York, NY, 1996.

Michael R Lyu and Allen Nikora. Casre: a computer-aided software reliability estimation tool. In *Computer-Aided Software Engineering, 1992. Proceedings., Fifth International Workshop on*, pages 264–275. IEEE, 1992.

Xie Min. *Software reliability modelling*, volume 1. World Scientific, 1991.

Hiroyuki Okamura and Tadashi Dohi. SRATS: Software reliability assessment tool on spreadsheet (experience report). In *Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium on*, pages 100–107. IEEE, 2013.

H. Pham. *Software Reliability*.

K Trivedi. *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*. Wiley Interscience Publication, 2001.

Shigeru Yamada and Shunji Osaki. Reliability growth models for hardware and software systems based on nonhomogeneous Poisson processes: a survey. *Microelectronics Reliability*, 23(1):91–112, 1983.

Shigeru Yamada, Hiroshi Ohtera, and Hiroyuki Narihisa. Software reliability growth models with testing-effort. *IEEE Transactions on Reliability*, 35(1):19–23, 1986.

Appendix A
Mathematical review

This chapter provides a review of basic mathematical identities used in subsequent sections such as differentiation, exponents, and logarithms.

1. Rules of differentiation:

$$\frac{d}{dx}c = 0 \tag{A.1}$$

   where c is a constant.

$$\frac{d}{dx}cx = cdx \tag{A.2}$$

$$\frac{d}{dx}x^n = nx^{(n-1)}dx \tag{A.3}$$

$$\frac{d}{dx}a^x = a^x \ln a dx \tag{A.4}$$

$$\frac{d}{dx}lnx = \frac{1}{x}dx \tag{A.5}$$

2. Properties of exponents:

$$a^x a^y = a^{(x+y)} \tag{A.6}$$

   Equation (A.6) generalizes to

$$\prod_{i=1}^{n} a^{x_i} = a^{\Sigma_{i=1}^{n}(x_i)} \tag{A.7}$$

$$\frac{a^x}{a^y} = a^{(x-y)} \tag{A.8}$$

$$(a^x)^y = a^{xy} \tag{A.9}$$

3. Properties of logarithms:

$$\log_b(xy) = \log_b x + log_b y \tag{A.10}$$

Similar to Equation (A.7), Equation (A.10) can be generalized to

$$\log_b \prod_{i=1}^{n} x_i = \sum_{i=1}^{n} \log_b x_i \tag{A.11}$$

$$\log_b(\frac{x}{y}) = \log_b(x) - \log_b(y) \tag{A.12}$$

$$\log_b(x^y) = y \log_b x \tag{A.13}$$

Appendix B
Installation and Starting SFRAT

An automated installation script has been prepared and is available from the GitHub repository at:

https://github.com/LanceFiondella/srt.core/blob/master/install_script.R.

The manual installation procedure is:

- Make sure that your platform is either 64-bit Windows 7 or later, Mac OS X 10.9 or later, or a version of Linux capable of running R Studio.
- Perl 5 version 16 or later.
- Install R and RStudio on your machine. You can download both RStudio and R at rstudio.com. For Windows, Mac OS X, and Linux, you will need version R version 3.0 or later and RStudio 0.99.482 or later.
  - R (https://cran.r-project.org/) needs to be installed before RStudio can be installed. Once RStudio has been installed, the following packages need to be installed.
    - **shiny** - is a web application framework for R.
    - **gdata** - is a package that provides various R programming tools for data manipulation.
    - **ggplot** - is a graphical package, which offers a powerful graphics language for creating elegant and complex plots.
    - **DT** - is a package that provides an R interface to the JavaScript library DataTables.
    - **rootSolve** - is a package to find the roots of n nonlinear (or linear) equations.
    - **knitr** - is a package that provides a general-purpose tool for dynamic report generation in R using Literate Programming techniques.

  - **rmarkdown** - is a package that includes high level functions for converting R Markdown documents into a variety of formats including HTML, MS Word PDF, and Beamer.

– You can install the packages using the Install packages menu item in RStudios Tools menu. This will bring up the dialog box in Figure B.1:

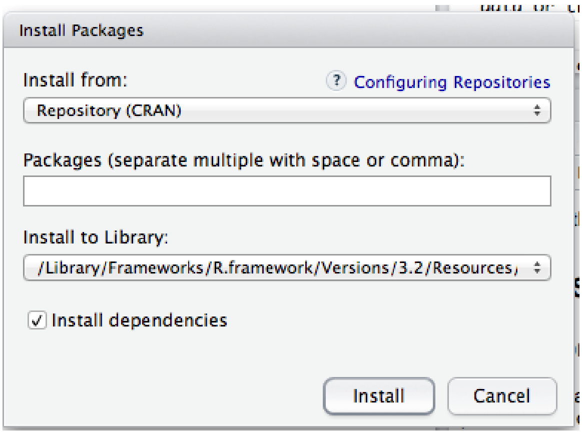

Fig. B.1: Install Packages Dialog Box

Specify the package(s) you want to install in the first line of the dialog box. It is not necessary to change the installation location, so just leave the second line alone. Make sure that the "*Install dependencies*" box is checked if is not, the packages you install may not work the way they should.

Once you have installed RStudio, configure your system so that R scripts (files with an ".R" extension) are opened with RStudio. This will make it easy to start SFRAT.

- Make a directory called "SFRAT" on a portion of your disk to which you have write access.
- Download the SFRAT files from https://github.com/LanceFiondella/srt.core and copy the SFRAT files to the folder you have created in the previous step.
- Open RStudio and find the folder in which you have installed SFRAT and double-click on the icon representing either "server.R" or "ui.R".
- To start SFRAT, simply click on the Run App button at the top right corner of your workspace as shown in Figure B.2 below.

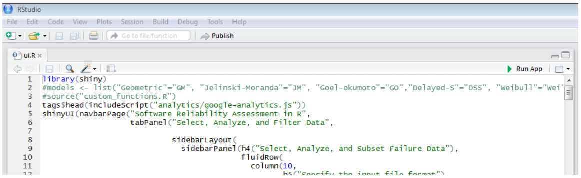

Fig. B.2: SFRAT application launch using RStudio

- The tool is also accessible through a web instance at http://sasdlc.org/.

Acronyms

AIC Akaike Information Criterion
CASRE Computer-Aided Software Reliability Estimation tool
CSV Comma Separated Value
FN Failure Number
FT Failure Times
IF Inter Failure
NHPP Non-homogeneous Poisson process
PSSE Predictive Sum of Squares Estimation
SASDLC System and Software Development Life Cycle
SFRAT Software Failure and Reliability Assessment Tool
SMERFS Statistical Modeling and Estimation of Reliability Functions for Software
SRATS Software Reliability Assessment Tool on Spreadsheet
SRGM Software Reliability Growth Model