Extending Commingling Analysis to Exponential Distributions and its Genetic Applications

A Dissertation Presented

by

Jung Yeon Lee

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

August 2008

Stony Brook University
The Graduate School

Jung Yeon Lee

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Stephen J. Finch-Dissertation Advisor
Professor
Applied Mathematics and Statistics

Nancy R. Mendell-Chairperson of Defense
Professor
Applied Mathematics and Statistics

Hongshik Ahn
Professor
Applied Mathematics and Statistics

Derek Gordon
Associate Professor
Genetics
Rutgers, the State University of New Jersey

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Extending Commingling Analysis to Exponential Distributions and its Genetic Applications

by

Jung Yeon Lee

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2008

Normal distribution commingling analysis tests whether data could better be transformed to a mixture of two normal components or transformed to a single component normal distribution and is an important tool of genetic analysis. This procedure is extended here to test whether data could better be transformed to a mixture of two exponentially distributed components or transformed to a single exponentially distributed component using a likelihood ratio test statistic (LRTS). I further extend the procedure to apply to censored data so that exponential commingling analysis can be applied to survival data as a first step in assessing whether there is potential genetic explanation for survival data. The null distribution of the LRTS for exponential commingling analysis is estimated by a simulation study. The null distribution of the LRTS has a low probability of zero values for small censoring rates. The distribution of the non-zero LRTS values appears to follow a gamma distribution with somewhat more than two degrees of freedom, roughly consistent with prior findings for commingling analysis. The power of this test is also simulated. Increasing sample size, larger differences between means, mixing proportion closer to 50%, and lower censoring rate are associated with greater power. The power of exponential commingling analysis for sample sizes typically used in genetic studies is large enough to be practically useful.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

My research starts with the question whether the data reflect a homogeneous process or whether there are indications of two components. This question has fundamental importance in genetic and pharmacological research. For example, the fact that the distribution of a quantitative phenotype is better characterized by a mixture of distributions than a single distribution is evidence for a single gene with major effect, as shown by Maclean, Elston, and Morton [1]. Viswanathan et al. [2] did a commingling analysis with a mixture model after Box-Cox transformation to test the hypothesis of a major genetic determinant of intraocular pressure (IOP) under the mixed genetic model and to elucidate the relationship between IOP and glaucoma. The mixture analysis on IOP was performed using a program SKUDRIVER. The best fitted model for complete population data set was a normal mixture distribution with three components. Bosse et al. [3] suggested the existence of a single gene with a major effect on the phenotype. A three component model provided a better fit to the data than a two component or a single component model. Ross et al. [4] used mixture analysis to determine the relationship between eye tracking disorder in schizophrenia which is a qualitative phenotype and the deficit syndrome. After transformation, the distribution of patients was best fit by a mixture of two normal distributions.

In addition to application of mixture analysis with normal distributions, there are potential applications of mixture analysis with exponential distributions. Mixture analysis with exponential distributions could be used with any ratio scale data since the data have only positive values, for example, microarray data or time to event data. It could also be applied to survival data with right censored data; for example, a study of age of onset might have to deal with censoring on the right. The ability to transform such data to exponentially distributed random variables would be a great advantage since one could then use the log-rank test, which performs well for exponentially distributed data.

Consequently, I develop procedure to determine whether there is an indication of a mixture mechanism in survival data after Box-Cox transformation. Specifically, I develop the likelihood ratio test statistic (LRTS) of the null hypothesis that the distribution is exponential against the alternative hypothesis that the distribution is a mixture of two exponentials after Box-Cox transformation. I also compute the maximum likelihood estimators (MLEs) of the parameters and use simulation techniques to estimate the null distribution and alternative distribution of the statistic.

My dissertation will address four questions.

1. How does one obtain MLEs for the exponential commingling analysis?
2. What is an effective method for maximization of the exponential commingling likelihood function?
3. Is there an invariant null distribution?
4. Does the test have reasonable power?

My dissertation consists of 5 chapters. Chapter 1 contains the statement of the research questions. In chapter 2, I review the literature that is relevant to my research questions: for example, commingling analysis (which is mixture analysis with normal distributions after Box-Cox transformation), the likelihood function approach to survival analysis, the expectation-maximization (EM) algorithm, and the likelihood ratio test in mixture problems. In chapter 3, I calculate the likelihood functions and first derivative functions to estimate the maximum likelihood estimators (MLEs). I also determine the number of random starting points (RSPs) to use in the EM computations. I also compute the censoring parameters that I will use in my simulation study. In chapter 4, the null distribution of the simulation study is approximated using simulation. The results of power study are also presented and compared to the power of the direct test of exponential mixtures. Chapter 5 has the answers to the four research questions presented in chapter 1 and discusses further research issues.

Chapter 2 Existing Methods

1. Box-Cox Transformation to Normality

1.1 Box-Cox Transformation to a Single Normal Distribution

The standard assumption in a linear model is that the observations $y_1, y_2, \cdots, y_n$ are sampled from independently normally distributed random variables with constant variance and with expectations specified by a model linear in a set of parameters θ. Box and Cox [5] considered the less restrictive assumption that such a normal, homoscedastic, linear model describes the data after some suitable transformation has been applied to $y_1, y_2, \cdots, y_n$. They work with a parametric family of transformations from y to $y^{(\lambda)}$, with the parameter λ defining the transformation. Two important examples are

$$y^{(\lambda)} = \begin{cases} \dfrac{y^{\lambda}-1}{\lambda} \ (\lambda \neq 0), \\ \log y \ (\lambda = 0), \end{cases} \tag{2.1.1}$$

and

$$y^{(\lambda)} = \begin{cases} \dfrac{(y+\lambda_2)^{\lambda_1}-1}{\lambda_1} \ (\lambda_1 \neq 0), \\ \log(y+\lambda_2) \ (\lambda_1 = 0). \end{cases} \tag{2.1.2}$$

The transformations (2.1.1) hold for $y > 0$, and (2.1.2) for $y > -\lambda_2$.

Their approach is to estimate λ and the other parameters of the model using maximum likelihood. For example, this analysis could show that $\sqrt{y}$ is the best scale for normality and homoscedasticity. The probability density for the original observations is obtained by multiplying the normal density by the Jacobian of the Box-Cox transformation. The likelihood in relation to the original observations y is thus

$$\frac{1}{(2\pi)^{\frac{1}{2}n\sigma^n}} \exp\left\{-\frac{\left(y^{(\lambda)} - a\theta\right)'\left(y^{(\lambda)} - a\theta\right)}{2\sigma^2}\right\} J(\lambda; y)$$

where

$$J(\lambda; y) = \prod_{i=1}^{n} \left| \frac{dy_i^{(\lambda)}}{dy_i} \right|$$

and

$$E\left(Y^{(\lambda)}\right) = a\theta .$$

This approach leads directly to point estimates of the parameters and to approximate tests and confidence intervals for λ based on the chi-squared distribution [6].

1.2 Box-Cox Transformation to Mixture of Two Normal Distributions

Maclean et al. [1] described the transformation of data using the Box-Cox transformation for commingling analysis. First, they found the Box-Cox transformation that maximized the probability that the data came from a mixture of two normally distributed components, with common (but unknown) variance. They also found the Box-Cox transformation that maximized the probability that the data came from a normally distributed random variable. They used the LRTS to determine whether the mixture model fit better than a single component normal distribution.

Ning and Finch [7] conducted a simulation study of the Maclean et al. procedure. They found that the algorithm for the calculation of the MLEs of the unknown parameters must use a large number of random starting points to protect against convergence to a local rather than global maximum. They found that the null distribution appeared to be the same for each of the Box-Cox transformations studied. Further, the distribution appeared to be a chi-square random variable for samples of 25 or more. For samples of size 500 the null distribution was roughly a chi-square distribution with 2.5 degrees of freedom. Ning and Finch [8] also studied the alternative distribution of the LRTS.

2. Survival Analysis

The problem of analyzing time to event data arises in a number of applied fields, such as medicine, biology, public health, epidemiology, engineering, economics, and demography. Survival analysis attempts to answer questions such as: what is the fraction of a population that will survive past a specified time? Of those that survive, at what rate will they die or fail? Can multiple causes of death or failure be taken into account? How do particular circumstances or characteristics decrease the odds of survival?

The object of the primary interest is the survival function, denoted S, which is defined as $S(t) = P(T > t)$ where t is a specified time value and T is a random variable denoting the time of death. That is, the survival function is the probability that the time of death is later than some specified time. The survival function must be non-increasing: $S(u) \le S(t)$ when $u > t$.

A common feature of these data is censoring. Censored data arises when an individual's death is known to occur only in a certain period of time. Possible censoring schemes are right censoring, where all that is known is that the individual is still alive at a given time, left censoring when all that is known is that the individual has experienced the event of interest prior to the start of the study, or interval censoring, where the only information is that the event occurs within some interval.

Survival models can be usefully viewed as ordinary regression models in which the response variable is time. However, computing the likelihood function is complicated by censoring. The likelihood function for a survival model, in the presence of censored data, is, by definition, the joint probability of the data given the parameters of the model. Then the likelihood function is the product of the likelihood of each observation. More specifically, the likelihoods for various types of censoring schemes may all be written by incorporating the following components [9].

$f(x)$; exact lifetimes

$S(C_r)$; right-censored observations

$1-S(C_l)$; left-censored observations

$[S(L)-S(R)]$; interval-censored observations

The likelihood function may be constructed by multiplying the component parts:

$$L=\prod_{i\in D} f(x_i)\prod_{i\in R} S(C_r)\prod_{i\in L}(1-S(C_l))\prod_{i\in I}(S(L_i)-S(R_i))$$

where D is the set of times of death, R the set of right-censored observations,

L the set of left-censored observations, and I the set of interval-censored observations.

3. EM algorithm

Dempster et al. [10] pointed out that the expectation-maximization (EM) method had been “proposed many times in special circumstances” by other authors. Their paper generalized the method and developed its theory. Since an EM iteration does not decrease the likelihood function, it is often used for finding maximum likelihood estimates of parameters in probabilistic models where the model depends on unobserved latent variables.

EM alternates between performing an E-step, which computes an expectation of the likelihood by including the latent variables as if they were observed, and an M-step, which computes maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E-step. The parameters found on the M-step are then used to begin another E-step, and the process is repeated.

The observable data y in this problem are “incomplete” in the sense that the component describing each observation is “missing.” Let z denote the missing component membership information. Together, z and y form the complete data. The complete data likelihood is the probability density function of the complete data with parameters given by the vector θ. Further, the conditional distribution of the missing data given the observed can be expressed as

$$f(z \mid y, \theta) = \frac{f(y, z \mid \theta)}{f(y \mid \theta)} = \frac{f(y \mid z, \theta) f(z \mid \theta)}{\int f\left(y \mid \hat{z}, \theta\right) f\left(\hat{z} \mid \theta\right) d\hat{z}}.$$

The EM algorithm iteratively improves an initial estimate θ_0 by constructing new estimates θ_1, θ_2 and so on. An individual re-estimation step that derives θ_{n+1} from θ_n has the following form:

$$\theta_{n+1} = \arg\max_{\theta} Q(\theta)$$

where

$$Q(\theta) = E_z\left[\log f(y, z \mid \theta) \mid y\right] = \int_{-\infty}^{\infty} f(z \mid y, \theta_n) \log f(y, z \mid \theta) dz.$$

4. Likelihood Ratio Test Statistic

One standard way of approaching the problem of testing for the number of components in a mixture model is to use the likelihood ratio test statistic (LRTS) which is defined by

$$-2\log\hat{L}_0-\left(-2\log\hat{L}_1\right),$$

where $\log\hat{L}_1$ is the log-likelihood function maximized under the alternative hypothesis and $\log\hat{L}_0$ is the log-likelihood function maximized under the null hypothesis.

Let g be the number of components in the model. Suppose we wish to test the null hypothesis $H_0 : g = g_0$ versus the alternative hypothesis $H_1 : g = g_1$ for some $g_1 > g_0$. Usually, $g_1 = g_0 + 1$ in practice, as many researchers consider a sequence of models such that $g_{i+1} = g_i + 1$. Let $\hat{\Theta}_i$ denote the maximum likelihood estimator (MLE) of Θ calculated for $H_i\,(i = 0,1)$. Then the data is not supportive of H_0 when the LRTS is sufficiently large. For mixture testing problems, there is the complication that the regularity conditions for the LRTS to have the asymptotic null distribution be a chi-squared distribution with degrees of freedom equal to the difference between the number of parameters under the null and alternative hypotheses do not hold [11].

To explain this, suppose that the component densities are completely specified. Then the parameter vector Θ consists of just the mixing proportions. Since $g_1 > g_0$, the null hypothesis is specified by the true value of Θ being on the boundary of the parameter space of the alternative (with one or more of the mixing proportions specified as zero). Further, if the component densities belong to the same parametric family $f(y_j;\theta)$ with θ unspecified, then H_0 will hold also if $\theta_k = \theta_l$ for some $k \neq l$. That is, H_0 corresponds to a nonidentifiable subset of the parameter space. Thus with the true value of the parameter vector under H_0 lying on the boundary of the parameter space and also in a nonidentifiable subset if the component densities depend on unknown parameters, the classic regularity condition in [12] about the asymptotic properties of the MLE are not valid under the null hypothesis H_0. In particular, the asymptotic distribution of the MLE in the nonidentifiable case under H_0 is unknown. The lack of identifiability leads to a degeneracy in the information matrix when considering the asymptotic null distribution of the (normalized) log likelihood formed under the alternative distribution H_1. As a consequence, when using classical Taylor series expansions for the LRTS, the remainder terms may not be bounded uniformly [13]. The null distribution of the LRTS for mixtures of gamma distributions without censoring diverges asymptotically at a rate of at least $\log\log n$ to infinity in probability [14].

Consequently, I obtain the null distribution of LRTS when the null hypothesis is a single exponential distribution and alternative hypothesis is mixture of two exponential distributions after Box-Cox transformation empirically. Additionally, I estimate the power of the test by simulation.

Chapter 3 Methods

1. The Transformation

I consider a Box-Cox like transformation to the data:

$$X^{(\lambda)} = X^{\lambda}, \lambda > 0 .$$

I restrict $\lambda > 0$ so that the transformation is always monotonically increasing.

2. Likelihood Functions

There are 13 likelihood functions that I must specify. The underlying distribution can be either a single component or mixture of exponential random variables, and there can be no censoring, right censoring, left censoring or interval censoring.

2.1 No Censoring

Case1: single exponential, no transformation

Let X be a random variable following an exponential distribution with mean μ. Then the likelihood function of a random sample of n observations is

$$L(x_i) = \prod_{i=1}^{n} \frac{1}{\mu} e^{-\frac{x_i}{\mu}}, \quad i = 1, \cdots, n,$$

with

$$\ln L(x_i) = -n \ln \mu - \frac{\sum_{i=1}^{n} x_i}{\mu}.$$

The MLE of μ [15] is

$$\hat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n}.$$

Case 2: mixture of two exponentials, no transformation

The probability density function of the mixture distribution with two exponential components, one with smaller mean μ_1 and the other with larger mean μ_2 ($\mu_1 < \mu_2$) is

$$f(x) = \pi \left(\frac{1}{\mu_1} \right) e^{-\frac{x}{\mu_1}} + (1 - \pi) \left(\frac{1}{\mu_2} \right) e^{-\frac{x}{\mu_2}}.$$

Here the probability of being in the component with smaller mean is π. The likelihood function is

$$L(\underline{\mu}, \pi) = \prod_{i=1}^{n} \left\{ \pi \left(\frac{1}{\mu_1} \right) e^{-\frac{x_i}{\mu_1}} + (1 - \pi) \left(\frac{1}{\mu_2} \right) e^{-\frac{x_i}{\mu_2}} \right\}.$$

Case 3: single exponential, transformation estimated

First, I find the transformation that makes the data most appear to follow a single component exponential distribution. Following Box and Cox [5], the Jacobian of the transformation for a single observation is

$$J(x) = \frac{dx^{\lambda}}{dx} = \lambda \cdot x^{\lambda-1},$$

and the absolute value of the Jacobian for n independent identically distributed (iid) observations can be written as

$$\prod_{i=1}^{n} \left| J(x_i) \right| = \prod_{i=1}^{n} \left| \frac{dx_i^{\lambda}}{dx_i} \right| = \left| \lambda^n \prod_{i=1}^{n} x_i^{\lambda-1} \right|.$$

Then the likelihood function of the transformed data is

$$L(\mu, \lambda) = \left(\frac{1}{\mu} \right)^n e^{-\frac{\sum_{i=1}^{n} x_i^{\lambda}}{\mu}} \cdot \prod_{i=1}^{n} \left| J(x_i) \right|.$$

After substituting $\hat{\mu}$ above in the log likelihood function, I find that

$$\ln L(\mu, \lambda) = -n \ln \frac{\sum_{i=1}^{n} x_i^{\lambda}}{n} - n + n \ln|\lambda| + |\lambda - 1| \sum_{i=1}^{n} \ln|x_i|.$$

Second, I find the MLE of λ by finding the root of the equation in which the partial derivative function with respect to λ is equal to zero. This partial derivative is:

$$\begin{aligned}\frac{\partial \ln L(\mu, \lambda)}{\partial \lambda} &= -n\sum_{i=1}^{n}\frac{x_i^{\lambda}}{n}\ln x_i + n\ln|\lambda| + |\lambda - 1|\sum_{i=1}^{n}\ln|x_i| \\ &= -\sum_{i=1}^{n} x_i^{\lambda}\ln x_i + \frac{n}{|\lambda|} + \sum_{i=1}^{n}\ln|x_i|\end{aligned}$$

Case 4: mixture of two exponentials, transformation estimated

Once again, the Jacobian for the single observation is:

$$J(x) = \frac{dx^{\lambda}}{dx} = \lambda \cdot x^{\lambda-1},$$

so that the likelihood is

$$L(\mu, \lambda) = \prod_{i=1}^{n}\left\{\sum_{j=1}^{2}\pi_j\left(\frac{1}{\mu_j}\right)e^{-\frac{x_i^{\lambda}}{\mu_j}}\right\}\cdot|J(x)|.$$

The log likelihood function is

$$\ln L(\underline{\mu}, \underline{\pi}, \lambda) = \sum_{i=1}^{n}\ln\left\{\sum_{j=1}^{2}\pi_j\left(\frac{1}{\mu_j}\right)e^{-\frac{x_i^{\lambda}}{\mu_j}}\right\} + n\ln|\lambda| + |\lambda - 1|\sum_{i=1}^{n}\ln|x_i|.$$

2.2 With Censoring

2.2.1 Right Censoring

In right censoring, the observed time is the minimum of the total time to event and the censoring time. A typical clinical trial starts with a fixed number of patients to which treatments are applied. Because of time or cost considerations, the investigator will terminate the study or report the results before all subjects realize their events. That is, some patients' times are right-censored.

Notation for right censoring:

t_i^* = the total time to event

u_i = the censored time

$t_i = \min(t_i^*, u_i),\ (1 \le i \le n) =$ the observed survival time

c_i = absence of censoring indicator of t_i

Case 5: single exponential, no transformation

Let the random variable $T \sim Exponentid(\mu)$. Then the likelihood function

$$L(\mu) = \prod_{i=1}^{n} \left(\frac{1}{\mu} e^{-\frac{t_i}{\mu}} \right)^{c_i} \left(e^{-\frac{t_i}{\mu}} \right)^{1-c_i},$$

and its log likelihood is

$$\ln L(\mu) = \sum_{i=1}^{n} \left\{ c_i \left(\ln \frac{1}{\mu} - \frac{t_i}{\mu} \right) - (1 - c_i) \frac{t_i}{\mu} \right\} \text{ [9].}$$

Since

$$\frac{\partial \ln L(\mu)}{\partial \mu} = \sum_{i=1}^{n}\left(-\frac{c_i}{\mu}+\frac{t_i}{\mu^2}\right),$$

the MLE of μ is

$$\hat{\mu} = \frac{\sum_{i=1}^{n} t_i}{\sum_{i=1}^{n} c_i} \quad .$$

Case 6: single exponential, transformation estimated

The likelihood function is

$$L(\mu, \lambda) = \prod_{i=1}^{n}\left(\frac{1}{\mu} e^{-\frac{t_i^\lambda}{\mu}} \cdot |J(t_i)|\right)^{c_i}\left(e^{-\frac{t_i^\lambda}{\mu}}\right)^{1-c_i} \quad ,$$

where

$$\prod_{i=1}^{n} J(t_i)^{c_i} = \lambda^{\sum_{i=1}^{n} c_i} \prod_{i=1}^{n} t_i^{c_i(\lambda-1)} \, .$$

Its log likelihood function is

$$\ln L(\mu, \lambda) = \sum_{i=1}^{n}\left\{c_i \cdot \ln\left(\frac{\sum_{i=1}^{n} c_i}{\sum_{i=1}^{n} t_i^\lambda}\right) - t_i^\lambda \cdot \frac{\sum_{i=1}^{n} c_i}{\sum_{i=1}^{n} t_i^\lambda} + c_i \ln|\lambda| + c_i|\lambda - 1|\ln|t_i|\right\}$$

by substituting

$$\hat{\mu} = \frac{\sum_{i=1}^{n} t_i^{\lambda}}{\sum_{i=1}^{n} c_i}. \qquad (2.2.1.1)$$

Case 7: mixture of two exponentials, transformation estimated

As in case 6,

$$\prod_{i=1}^{n} J(t_i)^{c_i} = \lambda^{\sum_{i=1}^{n} c_i} \prod_{i=1}^{n} t_i^{c_i(\lambda-1)}.$$

The likelihood function is then

$$L(\underline{\mu}, \underline{\pi}, \lambda) = \prod_{i=1}^{n} \left[\left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i^{\lambda}}{\mu_j}} \cdot \left| J(t_i) \right| \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \cdot e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\}^{1-c_i} \right].$$

The log likelihood function is

$$\ln L(\underline{\mu}, \underline{\pi}, \lambda) = \sum_{i=1}^{n} \left[c_i \ln \left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\} + (1-c_i) \ln \left\{ \sum_{j=1}^{2} \pi_j \cdot e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\} + c_i \ln|\lambda| + c_i |\lambda - 1| \ln|t_i| \right]$$

2.2.2 Left Censoring

In left censoring, the time recorded is the maximum of the total time to event and the censoring time. A lifetime associated with a specific individual in a study is considered to be left censored if it is less than a left censoring time. That is, the event of interest has already occurred for the individual before that person is observed in the study at a left censoring time. For example, in a study of early childhood learning, the dependent variable may be the age at which a child learns how to perform a task. A child who could perform the task at first observation would have a left censored time.

Notations for left censoring:

$t_i^* =$ the total time to event

$u_i =$ the censored time

$t_i = \max(t_i^*, u_i),\ (1 \le i \le n) =$ the observed survival time

$c_i =$ absence of censoring indicator of t_i

Case 8: single exponential, no transformation

Let the random variable $T \sim Exponentid(\mu)$. Then the likelihood function

$$L(\mu) = \prod_{i=1}^{n} \left(\frac{1}{\mu} e^{-\frac{t_i}{\mu}} \right)^{c_i} \left(1 - e^{-\frac{t_i}{\mu}} \right)^{1-c_i},$$

with

$$\ln L(\mu) = \sum_{i=1}^{n} \left\{ c_i \left(\ln \frac{1}{\mu} - \frac{t_i}{\mu} \right) + (1 - c_i) \ln \left(1 - e^{-\frac{t_i}{\mu}} \right) \right\}.$$

The MLE of $\hat{\mu}$ is a root of

$$\frac{\partial \ln L(\mu)}{\partial \mu} = \sum_{i=1}^{n} \left\{ -\frac{c_i}{\mu} + \frac{c_i \cdot t_i}{\mu^2} + (1 - c_i) \left(\frac{\frac{t_i}{\mu^2} \cdot e^{-\frac{t_i}{\mu}}}{1 - e^{-\frac{t_i}{\mu}}} \right) \right\} = 0.$$

Case 9: single exponential, transformation estimated

The likelihood function is

$$L(\mu, \lambda) = \prod_{i=1}^{n} \left(\frac{1}{\mu} e^{-\frac{t_i^{\lambda}}{\mu}} \cdot |J(t_i)| \right)^{c_i} \left(1 - e^{-\frac{t_i^{\lambda}}{\mu}} \right)^{1-c_i},$$

where

$$\prod_{i=1}^{n} J(t_i)^{c_i} = \lambda^{\sum_{i=1}^{n} c_i} \prod_{i=1}^{n} t_i^{c_i(\lambda-1)}.$$

The log likelihood function is

$$\ln L(\mu, \lambda) = \sum_{i=1}^{n} \left\{ c_i \cdot \left(\ln \frac{1}{\mu} - \frac{t_i^{\lambda}}{\mu} \right) + (1 - c_i) \ln \left(1 - e^{-\frac{t_i^{\lambda}}{\mu}} \right) + c_i \ln|\lambda| + c_i |\lambda - 1| \ln|t_i| \right\}.$$

Case 10: mixture of two exponentials, transformation estimated

Since, in this case,

$$\prod_{i=1}^{n} J(t_i)^{c_i} = \lambda^{\sum_{i=1}^{n} c_i} \prod_{i=1}^{n} t_i^{c_i(\lambda-1)},$$

the likelihood function is

$$L(\underline{\mu}, \underline{\pi}, \lambda) = \prod_{i=1}^{n} \left[\left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i^{\lambda}}{\mu_j}} \cdot |J(t_i)| \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \left(1 - e^{-\frac{t_i^{\lambda}}{\mu_j}} \right) \right\}^{1-c_i} \right].$$

The log likelihood function is

$$\ln L(\underline{\mu}, \underline{\pi}, \lambda) = \sum_{i=1}^{n} \left[\begin{array}{l} c_i \ln \left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\} \\ + (1-c_i) \ln \left\{ \sum_{j=1}^{2} \pi_j \left(1 - e^{-\frac{t_i^{\lambda}}{\mu_j}} \right) \right\} + c_i \ln|\lambda| + c_i |\lambda - 1| \ln|t_i| \end{array} \right]$$

2.2.3 Interval Censoring

In interval censoring, the observed value is the difference between the left end point and right end point. Interval censoring occurs when patients in a clinical trial have periodic follow-up. The patient's event time is only known to fall in the interval between the left follow-up point and the right follow-up point. Interval censored data commonly arise in studies where there is a non-lethal end point, such as the recurrence of a disease or condition. Animal tumorigenicity experiments may also have this characteristic.

Notation for interval censoring:

$l_i =$ left end point

$r_i =$ right end point

$t_i = (l_i, r_i] =$ the time interval observed for the i^{th} subject.

$c_i =$ absence of censoring indicator of t_i

Case 11: single exponential, no transformation

Let the random variable $T \sim Exponentiäl(\mu)$. Then the likelihood function

$$L(\mu) = \prod_{i=1}^{n} \left(\frac{1}{\mu} e^{-\frac{t_i}{\mu}} \right)^{c_i} \left(e^{-\frac{l_i}{\mu}} - e^{-\frac{r_i}{\mu}} \right)^{1-c_i},$$

with

$$\ln L(\mu) = \sum_{i=1}^{n} \left\{ c_i \left(\ln \frac{1}{\mu} - \frac{t_i}{\mu} \right) + (1 - c_i) \ln \left(e^{-\frac{l_i}{\mu}} - e^{-\frac{r_i}{\mu}} \right) \right\}.$$

Case 12: single exponential, transformation estimated

The likelihood function of the power transform when the data follow an interval censored single exponential distribution is

$$L(\mu,\,\lambda)=\prod_{i=1}^{n}\left(\frac{1}{\mu}e^{-\frac{t_i^{\lambda}}{\mu}}\cdot\left|J(t_i)\right|\right)^{c_i}\left(e^{-\frac{l_i^{\lambda}}{\mu}}-e^{-\frac{r_i^{\lambda}}{\mu}}\right)^{1-c_i},$$

where

$$J(t_i)^{c_i}=\lambda^{\sum_{i=1}^{n}c_i}\prod_{i=1}^{n}t_i^{c_i(\lambda-1)}.$$

Its log likelihood function is

$$\ln L(\mu,\,\lambda)=\sum_{i=1}^{n}\left\{c_i\cdot\left(\ln\frac{1}{\mu}-\frac{t_i^{\lambda}}{\mu}\right)+(1-c_i)\ln\left(e^{-\frac{l_i^{\lambda}}{\mu}}-e^{-\frac{r_i^{\lambda}}{\mu}}\right)+c_i\ln\left|\lambda\right|+c_i\left|\lambda-1\right|\ln\left|t_i\right|\right\}.$$

Case 13: mixture of two exponentials, transformation estimated

The likelihood function is

$$L(\underline{\mu},\,\underline{\pi},\,\lambda)=\prod_{i=1}^{n}\left[\left\{\sum_{j=1}^{2}\pi_j\left(\frac{1}{\mu_j}\right)e^{-\frac{t_i^{\lambda}}{\mu_j}}\cdot\left|J(t_i)\right|\right\}^{c_i}\left\{\sum_{j=1}^{2}\pi_j\left(e^{-\frac{l_i^{\lambda}}{\mu_j}}-e^{-\frac{r_i^{\lambda}}{\mu_j}}\right)\right\}^{1-c_i}\right].$$

Its log likelihood function is

$$\ln L(\underline{\mu}, \underline{\pi}, \lambda) = \sum_{i=1}^{n} \left[\begin{array}{l} c_i \ln\left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\} \\ + (1-c_i)\ln\left\{ \sum_{j=1}^{2} \pi_j \left(e^{-\frac{l_i^{\lambda}}{\mu_j}} - e^{-\frac{r_i^{\lambda}}{\mu_j}} \right) \right\} + c_i \ln|\lambda| + c_i|\lambda - 1| \ln|t_i| \end{array} \right].$$

3. Overview of the algorithm for maximizing the likelihood function

For the null likelihood, the first setting of the transformation parameter is 1. I then use the closed form estimate of the mean (equation 2.2.1.1) to get the first iteration MLE of the mean. I then use the Newton-Raphson algorithm to obtain the second iteration value of the transformation parameter. I then re-estimate the mean with the closed form estimate. I repeat this cycle until convergence occurs.

For the mixture likelihood, the first setting of the transformation parameter is 1. I use the EM algorithm below with the 50 RSPs as documented in section 4.1 to calculate means of the components and the mixing proportions. I then use the Newton-Raphson procedure to estimate a revised transformation parameter with the RSPs as described in section 5.1 Specifically, when the LRTS value with 20 RSPs is greater than or equal to 1.7, I use 20 RSPs for this iteration of the transformation parameter. When the LRTS value with 20 RSPs is greater than or equal to 1.1 and less than 1.7, I use 30 RSPs for this iteration of the transformation parameter. When the LRTS value with 20 RSPs is greater than or equal to 0.2 and less than 1.1, I use 40 RSPs for this iteration of the transformation parameter. Finally, when the LRTS value with 20 RSPs is less than 0.2, I use 100 RSPs for this iteration of the transformation parameter. With this next iterated value of the transformation parameter, I re-estimate the means of the components and the mixing proportions with the EM algorithm. I repeat this procedure until an iteration occurs in which the change in the transformation parameter is less than 0.05. The LRTS value is the value of the LRTS generated for the prior transformation value.

4. EM Algorithm

To specify the EM algorithm for this problem, I need to estimate the means and the mixing proportions for each component. There are seven cases to calculate the means and mixing proportions. Each case can be generalized to k components, where $k > 2$.

EM 1: mixture of two exponentials, no censoring, no transformation

The probability density function in case EM 1 is

$$f(x_i) = \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{x_i}{\mu_j}}, \quad i = 1, 2, \cdots, n, \ j = 1, 2 .$$

On E-step the EM algorithm calculates the probability that the i^{th} subject belongs to the j^{th} component;

$$\tau_{ji} = \frac{\pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{x_i}{\mu_j}}}{\sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{x_i}{\mu_j}}} .$$

On M-step, the EM algorithm finds the mixing proportion in the j^{th} component;

$$p_j = \frac{\sum_{i=1}^{n} \tau_{ji}}{n}$$

Also, the parameter μ_j which is the mean in the j^{th} component is found on M-step as a root of

$$\sum_{i=1}^{n} \tau_{ji} \frac{\partial \ln \left(\pi_i \frac{1}{\mu_j} e^{-\frac{x_i}{\mu_j}} \right)}{\partial \mu_j} = \sum_{i=1}^{n} \tau_{ji} \left(-\frac{1}{\mu_j} + \frac{x_i}{\mu_j^2} \right) = 0 .$$

Thus,

$$\mu_j = \frac{\sum_{i=1}^{n} \tau_{ji} \cdot x_i}{\sum_{i=1}^{n} \tau_{ji}} .$$

The parameters found on the M-step are then used to begin another E-step, and the process is repeated. The same process will be held in the other EM cases. In the following EM cases, I will write the equations only.

EM 2: mixture of two exponentials, right censoring, no transformation

The probability density is

$$f(t_i) = \left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i}{\mu_j}} \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \cdot e^{-\frac{t_i}{\mu_j}} \right\}^{1-c_i}, \quad i = 1, 2, \cdots, n, \ j = 1, 2 .$$

The probability that the i^{th} subject belongs to the j^{th} component is

$$\tau_{ji} = \frac{\left\{ \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i}{\mu_j}} \right\}^{c_i} \left\{ \pi_j \cdot e^{-\frac{t_i}{\mu_j}} \right\}^{1-c_i}}{\left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i}{\mu_j}} \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \cdot e^{-\frac{t_i}{\mu_j}} \right\}^{1-c_i}} .$$

The mixing proportion in the jth component is

$$p_j = \frac{\sum_{i=1}^{n} \tau_{ji}}{n} .$$

The parameter μ_j can be obtained as a root of

$$\sum_{i=1}^{n} \tau_{ji} \frac{\partial \ln \left\{ \left(\frac{\pi_j}{\mu_j} e^{-\frac{t_i}{\mu_j}} \right)^{c_i} \left(\pi_j \cdot e^{-\frac{t_i}{\mu_j}} \right)^{1-c_i} \right\}}{\partial \mu_j} = \sum_{i=1}^{n} \tau_{ji} \left(-\frac{c_i}{\mu_j} + \frac{t_i}{\mu_j^2} \right) = 0 .$$

Therefore,

$$\mu_j = \frac{\sum_{i=1}^{n} \tau_{ji} \cdot t_i}{\sum_{i=1}^{n} \tau_{ji} \cdot c_i}.$$

EM 3: mixture of two exponentials, right censoring, transformation estimated

The probability density function is

$$f(t_i^{\lambda}) = \left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i^{\lambda}}{\mu_j}} \cdot |J| \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \cdot e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\}^{1-c_i}$$

I use t^{λ} instead of t to calculate the parameters because the Jacobian term does not have the parameters. The mixing proportion in the j^{th} component is

$$p_j = \frac{\sum_{i=1}^{n} \tau_{ji}}{n}.$$

The probability that the i^{th} subject belongs to the j^{th} component is

$$\tau_{ji} = \frac{\left\{ \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\}^{c_i} \left\{ \pi_j \cdot e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\}^{1-c_i}}{\left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \cdot e^{-\frac{t_i^{\lambda}}{\mu_j}} \right\}^{1-c_i}}.$$

The parameter μ_j can be obtained by finding a root of

$$\sum_{i=1}^{n} \tau_{ji} \frac{\partial \ln \left\{ \left(\frac{\pi_j}{\mu_j} e^{-\frac{t_i^{\lambda}}{\mu_j}} \right)^{c_i} \left(\pi_j \cdot e^{-\frac{t_i^{\lambda}}{\mu_j}} \right)^{1-c_i} \right\}}{\partial \mu_j} = \sum_{i=1}^{n} \tau_{ji} \left(-\frac{c_i}{\mu_j} + \frac{t_i^{\lambda}}{\mu_j^2} \right) = 0.$$

Thus, the parameter μ_j can be obtained as

$$\mu_j = \frac{\sum_{i=1}^{n} \tau_{ji} \cdot t_i^{\lambda}}{\sum_{i=1}^{n} \tau_{ji} \cdot c_i}.$$

EM 4: mixture of two exponentials, left censoring, no transformation

The probability density function of this distribution is

$$f(t_i) = \left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i}{\mu_j}} \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \left(1 - e^{-\frac{t_i}{\mu_j}} \right) \right\}^{1-c_i}, \quad i = 1, 2, \cdots, n.$$

The probability that the i^{th} subject belongs to the j^{th} component is

$$\tau_{ji} = \frac{\left\{ \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i}{\mu_j}} \right\}^{c_i} \left\{ \pi_j \left(1 - e^{-\frac{t_i}{\mu_j}} \right) \right\}^{1-c_i}}{\left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i}{\mu_j}} \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \left(1 - e^{-\frac{t_i}{\mu_j}} \right) \right\}^{1-c_i}}.$$

The mixing proportion in the j^{th} component is

$$p_j = \frac{\sum_{i=1}^{n} \tau_{ji}}{n}.$$

The parameter μ_j is a root of

$$\sum_{i=1}^{n}\tau_{ji}\frac{\partial \ln\left\{\left(\frac{\pi_j}{\mu_j}e^{-\frac{t_i}{\mu_j}}\right)^{c_i}\left(\pi_j\left(1-e^{-\frac{t_i}{\mu_j}}\right)\right)^{1-c_i}\right\}}{\partial\mu_j}$$

$$=\sum_{i=1}^{n}\tau_{ji}\left\{-c_i\left(\frac{1}{\mu_j}+\frac{t_i}{\mu_j^2}\right)+(1-c_i)\left(-\frac{e^{-\frac{t_i}{\mu_j}}t_i}{\left(1-e^{-\frac{t_i}{\mu_j}}\right)\mu_j^2}\right)\right\}=0 \quad .$$

EM 5: mixture of two exponentials, left censoring, transformation estimated

The probability density function is

$$f(t_i^{\lambda})=\left\{\sum_{j=1}^{2}\pi_j\left(\frac{1}{\mu_j}\right)e^{-\frac{t_i^{\lambda}}{\mu_j}}\cdot|J|\right\}^{c_i}\left\{\sum_{j=1}^{2}\pi_j\left(1-e^{-\frac{t_i^{\lambda}}{\mu_j}}\right)\right\}^{1-c_i}$$

As before, I use t^{λ} instead of t to calculate the parameters because the Jacobian term does not have the parameters. The mixing proportion in the j^{th} component is

$$p_j=\frac{\sum_{i=1}^{n}\tau_{ji}}{n}$$

where, the probability that the i^{th} subject belongs to the j^{th} component is

$$\tau_{ji}=\frac{\left\{\pi_j\left(\frac{1}{\mu_j}\right)e^{-\frac{t_i^{\lambda}}{\mu_j}}\right\}^{c_i}\left\{\pi_j\left(1-e^{-\frac{t_i^{\lambda}}{\mu_j}}\right)\right\}^{1-c_i}}{\left\{\sum_{j=1}^{2}\pi_j\left(\frac{1}{\mu_j}\right)e^{-\frac{t_i^{\lambda}}{\mu_j}}\right\}^{c_i}\left\{\sum_{j=1}^{2}\pi_j\left(1-e^{-\frac{t_i^{\lambda}}{\mu_j}}\right)\right\}^{1-c_i}}.$$

The parameter μ_j can be obtained as a root of

$$\mu_j = \sum_{i=1}^{n} \tau_{ji} \left\{ c_i \left(-\frac{1}{\mu_j} + \frac{t_i^{\lambda}}{\mu_j^2} \right) - (1 - c_i) \frac{e^{-\frac{t_i^{\lambda}}{\mu_j}} \cdot t_i^{\lambda}}{\left(1 - e^{-\frac{t_i^{\lambda}}{\mu_j}} \right) \mu_j^{\lambda}} \right\} = 0 .$$

EM 6: mixture of two exponentials, interval censoring, no transformation

The probability density function of this distribution is

$$f(t_i) = \left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i}{\mu_j}} \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \left(e^{-\frac{l}{\mu_j}} - e^{-\frac{r}{\mu_j}} \right) \right\}^{1-c_i} , \quad i = 1, 2, \cdots, n .$$

The probability that the i^{th} subject belongs to the j^{th} component is

$$\tau_{ji} = \frac{\left\{ \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i}{\mu_j}} \right\}^{c_i} \left\{ \pi_j \left(e^{-\frac{l}{\mu_j}} - e^{-\frac{r}{\mu_j}} \right) \right\}^{1-c_i}}{\left\{ \sum_{j=1}^{2} \pi_j \left(\frac{1}{\mu_j} \right) e^{-\frac{t_i}{\mu_j}} \right\}^{c_i} \left\{ \sum_{j=1}^{2} \pi_j \left(e^{-\frac{l}{\mu_j}} - e^{-\frac{r}{\mu_j}} \right) \right\}^{1-c_i}} .$$

The mixing proportion in the j^{th} component is

$$p_j = \frac{\sum_{i=1}^{n} \tau_{ji}}{n} .$$

The parameter μ_j can be obtained as a root of

$$\sum_{i=1}^{n}\tau_{ji}\frac{\partial \ln\left[\left(\frac{\pi_j}{\mu_j}e^{-\frac{t_i}{\mu_j}}\right)^{c_i}\left\{\pi_j\left(e^{-\frac{l}{\mu_j}}-e^{-\frac{r}{\mu_j}}\right)\right\}^{1-c_i}\right]}{\partial\mu_j}$$

$$=\sum_{i=1}^{n}\tau_{ji}\left[c_i\left(-\frac{1}{\mu_j}+\frac{t_i}{\mu_j^2}\right)+(1-c_i)\left\{\frac{l_i\cdot e^{-\frac{l}{\mu_j}}-r_i\cdot e^{-\frac{r}{\mu_j}}}{\mu_j^2\left(e^{-\frac{l}{\mu_j}}-e^{-\frac{r}{\mu_j}}\right)}\right\}\right]=0$$

EM 7: mixture of two exponentials, interval censoring, transformation estimated

The probability density function is

$$f(t_i^{\lambda})=\left\{\sum_{j=1}^{2}\pi_j\left(\frac{1}{\mu_j}\right)e^{-\frac{t_i^{\lambda}}{\mu_j}}\cdot|J|\right\}^{c_i}\left\{\sum_{j=1}^{2}\pi_j\left(e^{-\frac{l^{\lambda}}{\mu_j}}-e^{-\frac{r^{\lambda}}{\mu_j}}\right)\right\}^{1-c_i}$$

Again, I use t^{λ} instead of t to calculate the parameters because the Jacobian term does not have the parameters. The mixing proportion in the j^{th} component is

$$p_j=\frac{\sum_{i=1}^{n}\tau_{ji}}{n}$$

where the probability that the i^{th} subject belongs to the j^{th} component is

$$\tau_{ji}=\frac{\left\{\pi_j\left(\frac{1}{\mu_j}\right)e^{-\frac{t_i^{\lambda}}{\mu_j}}\cdot J\right\}^{c_i}\left\{\pi_j\left(e^{-\frac{l^{\lambda}}{\mu_j}}-e^{-\frac{r^{\lambda}}{\mu_j}}\right)\right\}^{1-c_i}}{\left\{\sum_{j=1}^{2}\pi_j\left(\frac{1}{\mu_j}\right)e^{-\frac{t_i^{\lambda}}{\mu_j}}\cdot J\right\}^{c_i}\left\{\sum_{j=1}^{2}\pi_j\left(e^{-\frac{l^{\lambda}}{\mu_j}}-e^{-\frac{r^{\lambda}}{\mu_j}}\right)\right\}^{1-c_i}}$$

The parameter μ_j can be obtained as a root of

$$\sum_{i=1}^{n} \tau_{ji} \left[c_i \left(-\frac{1}{\mu_j} + \frac{t_i^{\lambda}}{\mu_j^2} \right) + (1 - c_i) \left\{ \frac{l^{\lambda} \cdot e^{-\frac{l^{\lambda}}{\mu_j}} - r^{\lambda} \cdot e^{-\frac{r^{\lambda}}{\mu_j}}}{\mu_j^2 \left(e^{-\frac{l^{\lambda}}{\mu_j}} - e^{-\frac{r^{\lambda}}{\mu_j}} \right)} \right\} \right] = 0 .$$

4.1 Determining the Number of Random Starting Points with the EM Algorithm

For given power transformation λ, the EM algorithm works very well with a good initial guess although it is slow and often converges to a local maximum. Consequently, I run the EM algorithm with a large number of random starting points (RSPs). Since the probability of finding the global maximum is increased by using more RSPs. I must specify the number of RSPs that is large enough so that the LRTS is the global maximum or close to it but small enough for a large simulation study.

I set the number of RSPs based on a pilot study. The sample size was 300 observations from an exponential distribution with mean 5 and expected censoring rate 30%. I processed 1000 samples (each of size 300) and used 25, 50 and 100 RSPs. I plotted the histogram of the likelihood function to determine how many solutions existed. With 100 RSPs, there were 4 to 6 solutions observed. For each sample of size 300, all solutions were observed with 50 RSPs. With 25 RSPs, however, only 2 or 3 of the solutions were observed. Consequently, I use 50 RSPs in my simulations. For each RSP, the convergence tolerance is the tolerance is 10^{-5} ; that is, the iteration process continues until the maximum number of iterations (1000) is reached or the difference in successive likelihood functions is less than the tolerance.

5. Newton-Raphson Method

I use the Newton-Raphson method to estimate the power transformation λ . This estimation can be obtained as a root of the partial derivative function of log-likelihood with respect to lambda. There are three cases, one for each type of censoring.

NRM 1: mixture of two exponentials, right censoring, transformation estimated

The partial derivative function with respect to lambda of the mixture of two exponential distributions with right censoring data is

$$\frac{\partial \ln L(\underline{\mu}, \underline{\pi}, \lambda)}{\partial\lambda} = \sum_{i=1}^{n}\left[\begin{array}{l} c_i\left\{\frac{\sum_{j=1}^{2}\pi_j \frac{1}{\mu_j} e^{-\frac{t_i^\lambda}{\mu_j}}\left(-\frac{t_i^\lambda}{\mu_j}\right)\cdot\ln(t_i)}{\sum_{j=1}^{2}\pi_j\frac{1}{\mu_j}\cdot e^{-\frac{t_i^\lambda}{\mu_j}}}\right\}+ \\ (1-c_i)\left\{\frac{\sum_{j=1}^{2} \pi_j\cdot e^{-\frac{t_i^\lambda}{\mu_j}}\left(-\frac{t_i^\lambda}{\mu_j}\right)\cdot\ln(t_i)}{\sum_{j=1}^{2}\pi_j\cdot e^{-\frac{t_i^\lambda}{\mu_j}}}\right\}+\frac{c_i}{|\lambda|}+c_i\cdot\ln|t_i|\end{array}\right].$$

The parameter λ can be estimated by setting this equation to zero.

NRM 2: mixture of two exponentials, left censoring, transformation estimated

The partial derivative function with respect to lambda of mixture of two exponential distributions with left censoring is

$$\frac{\partial \ln L(\underline{\mu}, \underline{\pi}, \lambda)}{\partial\lambda} = \sum_{i=1}^{n}\left[\begin{array}{l} c_i\left\{\frac{\sum_{j=1}^{2}\pi_j \frac{1}{\mu_j} e^{-\frac{t_i^\lambda}{\mu_j}}\left(-\frac{t_i^\lambda}{\mu_j}\right)\cdot\ln(t_i)}{\sum_{j=1}^{2}\pi_j\frac{1}{\mu_j} e^{-\frac{t_i^\lambda}{\mu_j}}}\right\}+ \\ (1-c_i)\left\{\frac{\sum_{j=1}^{2} \pi_j\left(e^{-\frac{t_i^\lambda}{\mu_j}}\right)\left(\frac{t_i^\lambda}{\mu_j}\right)\cdot\ln(t_i)}{\sum_{j=1}^{2}\pi_j\left(1-e^{-\frac{t_i^\lambda}{\mu_j}}\right)}\right\}+\frac{c_i}{|\lambda|}+c_i\cdot\ln|t_i|\end{array}\right].$$

NRM 3: mixture of two exponentials, interval censoring, transformation estimated

The partial derivative function with respect to lambda of mixture of two exponential distributions with interval censoring is

$$\frac{\partial \ln L(\underline{\mu},\, \underline{\pi},\, \lambda)}{\partial \lambda} = \sum_{i=1}^{n} \left[\begin{array}{l} c_i \left\{ \dfrac{\sum_{j=1}^{2} \pi_j \dfrac{1}{\mu_j} e^{-\frac{t_i^{\lambda}}{\mu_j}} \left(-\dfrac{t_i^{\lambda}}{\mu_j} \right) \cdot \ln(t_i)}{\sum_{j=1}^{2} \pi_j \dfrac{1}{\mu_j} e^{-\frac{t_i^{\lambda}}{\mu_j}}} \right\} + \\ (1-c_i) \left\{ \dfrac{\sum_{j=1}^{2} \pi_j \left(-\dfrac{l^{\lambda}}{\mu_j} e^{-\frac{l^{\lambda}}{\mu_j}} \ln(l) + \dfrac{r^{\lambda}}{\mu_j} e^{-\frac{r^{\lambda}}{\mu_j}} \ln(r) \right)}{\sum_{j=1}^{2} \pi_j \left(e^{-\frac{l^{\lambda}}{\mu_j}} - e^{-\frac{r^{\lambda}}{\mu_j}} \right)} \right\} + \dfrac{c_i}{|\lambda|} + c_i \cdot \ln|t_i| \end{array} \right].$$

5.1 Determining the Number of Random Starting Point with Newton-Raphson Method

I determine how many RSPs will be needed in the Newton-Raphson algorithm from a pilot study. I generate 1,000 samples of 300 observations from an exponential distribution with mean 5 and censoring rate 30%. The data were transformed to the power 1/3. After the algorithm is processed with 8 RSPs at each setting, I selected 25 samples using systematic sampling from samples which have positive LRTS. There were 22 samples with LRTS less than -.05. Each was considered in the pilot study. I added RSPs in groups of 8 and examined the increase in the LRTS with additional RSPs. I denote $L(I, K)$ is the LRTS value with I RSPs for the null distribution and K RSPs for the alternative distribution in the Table 1. Table 1 shows the results for the 25 samples with positive LRTS, and Table 2 shows the results for the samples with negative LRTS using 8 RSPs. I take the maximum value of the LRTS with 16 RSPs (the original 8 and the additional 8) and the maximum LRTS with 24 RSPs (the original 8, the additional 8, and the additional 8).

Table 1 Values of $L(I, K)$ for 25 samples with $L(8, 8) \geq 0$

Sub-Sample No.	Sample No.	$L(I, K)$		
		$L(8, 8)$	$L(16, 16)$	$L(24, 24)$
1	576	.000	.515	.515
2	2	.115	.134	.179
3	319	.241	.280	.328
4	398	.379	.382	.382
5	818	.490	.648	2.017
6	57	.601	.601	.601
7	864	.676	.685	.708
8	230	.802	.802	.802
9	84	.944	.956	.956
10	781	1.110	1.111	1.112
11	391	1.248	1.282	1.282
12	466	1.407	1.412	1.418
13	953	1.587	1.591	1.604
14	717	1.777	1.778	1.778
15	691	1.973	1.988	1.988
16	992	2.156	2.156	2.156
17	337	2.401	2.402	2.403
18	441	2.730	2.730	2.730
19	753	3.064	3.064	3.064
20	318	3.341	3.361	3.361
21	28	3.755	3.755	3.758
22	217	4.099	4.099	4.099
23	516	4.712	4.718	4.719
24	355	5.819	5.819	5.819
25	491	6.876	6.876	6.876

Table 2 Values of $L(I, K)$ for 22 samples with $L(8, 8) < 0$

Sub-Sample No.	Sample No.	$L(I, K)$		
		$L(8, 8)$	$L(16, 16)$	$L(24, 24)$
1	907	-.493	-.011	5.861
2	802	-.452	.212	.212
3	638	-.393	.624	.624
4	177	-.307	-.017	-.017
5	692	-.266	4.179	4.179
6	627	-.191	.948	.948
7	270	-.179	.096	.096
8	306	-.154	-.003	-.003
9	894	-.149	-.061	-.007
10	346	-.134	.030	.048
11	78	-.130	5.287	6.035
12	981	-.100	2.360	2.776
13	242	-.094	.597	.597
14	420	-.077	1.486	1.486
15	790	-.074	-.006	3.24
16	6	-.072	1.773	1.773
17	955	-.068	.784	.784
18	569	-.066	.577	.604
19	315	-.063	-.049	3.454
20	823	-.058	-.001	-.001
21	169	-.054	4.337	4.337
22	865	-.050	-.033	-.003

Each sample had an LRTS that was positive or close to zero with 24 RSPs. The smallest LRTS was -.017. There were a number of samples with a notable increase in LRTS from 16 to 24 RSPs. To examine the algorithm more carefully, 6 samples were chosen from the Table 1 and Table 2; these are starred in the table 3. One of these samples has a positive LRTS and the others have negative LRTS. I used an additional 24 RSPs to obtain the results shown in Table 3.

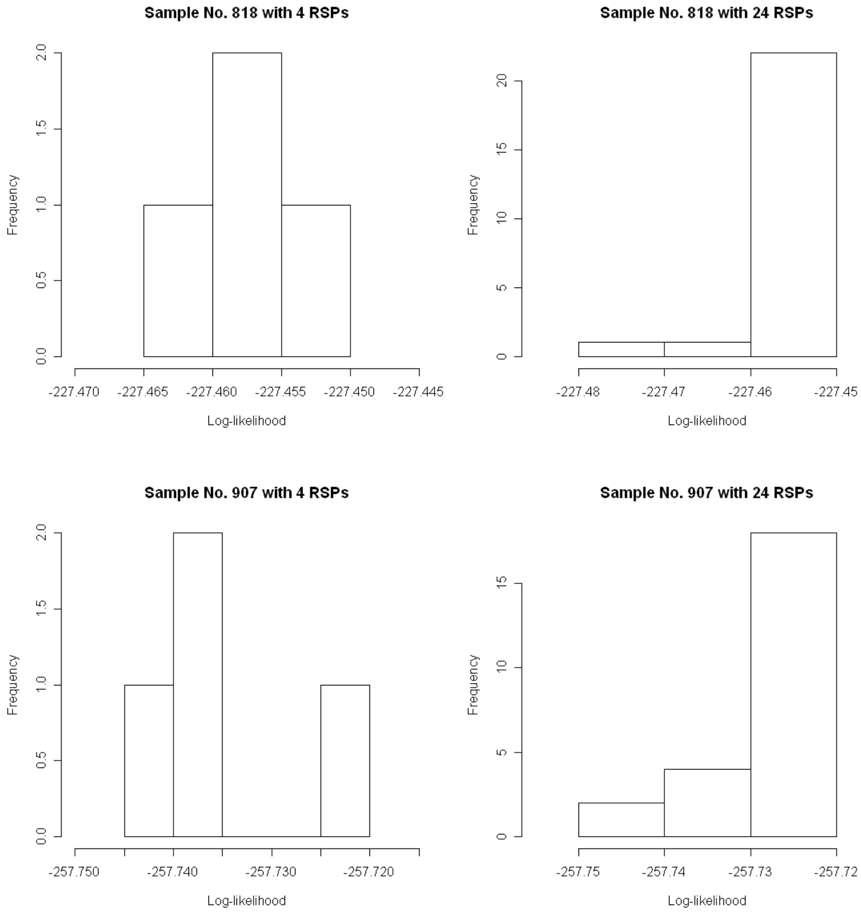
I conclude that I must use at least 32 RSPs, because all samples had positive LRTS using 32 RSPs. I also examined the histogram of likelihoods for the null distribution (L_0) (see Figure 1). Since the difference between the maximum null likelihood and minimum null likelihood is less than 0.01 (see the Figure 1), there is no apparent global convergence problem for null likelihood. Consequently, I reduce the number of RSPs for L_0 to 4. In Table 3, the value of LRTS (I, K) is the maximum L_1 value with K RSPs minus the maximum L_0 value with I RSPs.

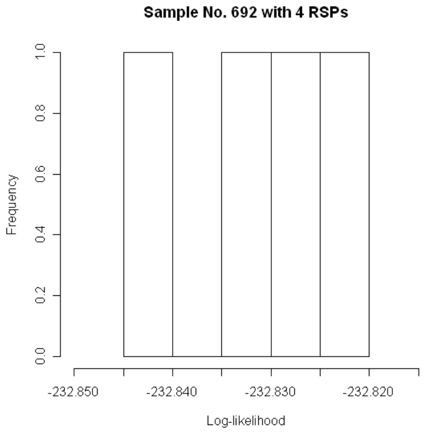
Table 3 Values of $L(I, K)$ for samples with large $L(I, K) - L(8, 8)$

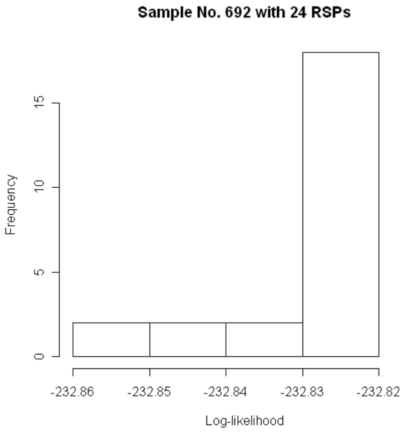
Sub-Sample No.		Sample No.	LRTS with Number of RSPs					
			L(8, 8)	L(16, 16)	L(24, 24)	L(32, 32)	L(4, 36)	L(4, 48)
Positive	5	818	.490	.648	2.017	2.017	2.023	2.039
Negative	1	907	-.493	-.011	5.861	5.962	5.962	5.962
	5	692	-.266	4.179	4.179	4.355	4.355	4.355
	11	78	-.130	5.287	6.035	6.035	6.028	6.028
	19	315	-.063	-.049	3.454	3.454	3.454	3.454
	21	169	-.054	4.337	4.337	4.337	4.337	4.375

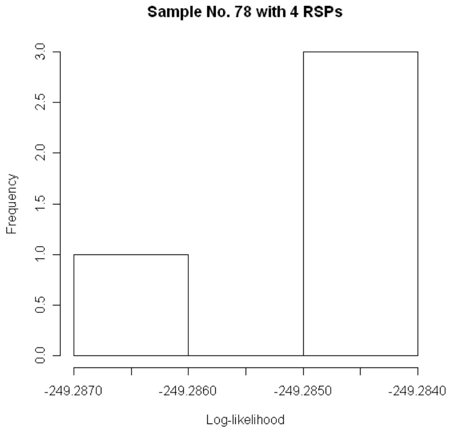
Figure 1 contains the histograms of L_0 with 4 or 24 RSPs. The sample number in the Figure 1 refers to the sample number in the Table 3. Difference of maximum L_0s with 4 and 24 RSPs is less than or equal to 0.176.

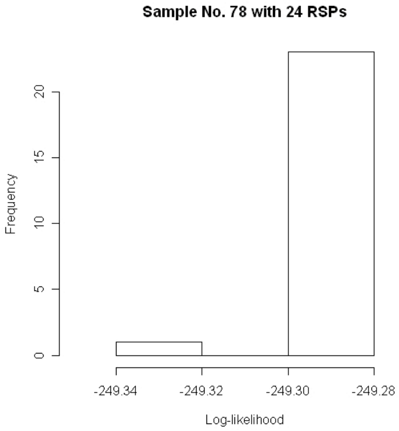
Figure 1 The histogram of L_0 for selected samples

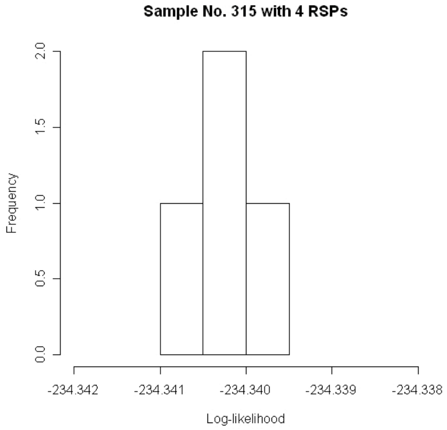


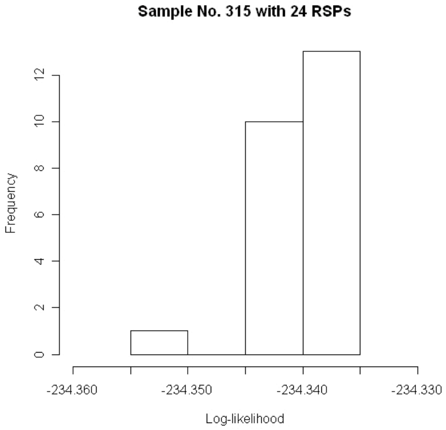

Sample No. 692 with 4 RSPs

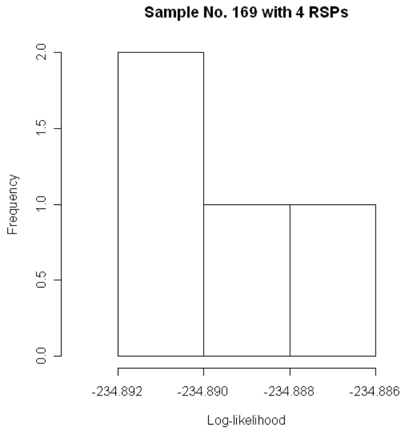

Sample No. 692 with 24 RSPs

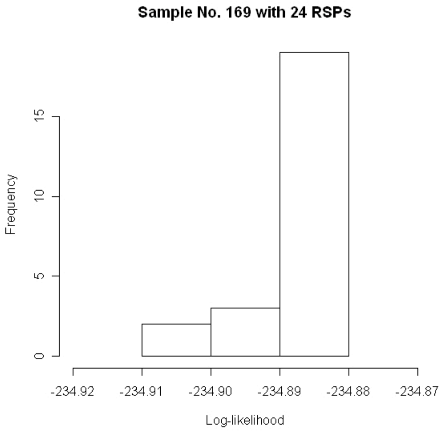

Sample No. 78 with 4 RSPs


Sample No. 78 with 24 RSPs


Sample No. 315 with 4 RSPs


Sample No. 315 with 24 RSPs





When the value of $L(8, 8)$ is large (>1.3), the increase in the value of $L(8, K)$ is small for $K \geq 8$. Consequently, I examined whether an algorithm in which I determined K from the value of $L(8, 8)$ could reduce my computational effort.

That is, I calculated $L(8, 8)$ for the 1,000 samples. I defined 25 strata based on the value of $L(8, 8)$. Stratum 1 has $L(8, 8) \leq 0.1$; stratum 2 has $0.1 < L(8, 8) \leq 0.2$;…; stratum 25 has $2.4 < L(8, 8) \leq 2.5$. I picked the three samples with smallest $L(8, 8)$ values in each stratum. I then calculated $L(4, 20)$, $L(4, 30)$, $L(4, 40)$, $L(4, 50)$, $L(4, 60)$ for each sample. The results are shown in Table 4.

Table 4 Values of $L(I, K)$ for 3 samples from 25 strata

Strata	Sample No.	$L(8, 8)$	$L(4, 20)$	$L(4, 30)$	$L(4, 40)$	$L(4, 50)$	$L(4, 60)$
1	576	0.000	0.086	0.086	0.271	0.271	0.769
	471	0.001	0.084	0.301	0.301	0.301	0.315
	834	0.002	0.002	0.703	1.827	1.827	1.827
2	351	0.101	1.240	1.457	1.563	1.563	1.563
	804	0.103	0.132	0.132	0.133	0.133	0.133
	813	0.107	0.143	0.143	0.143	0.143	0.143
3	415	0.207	0.459	0.459	1.076	1.076	1.081
	487	0.208	0.210	0.212	0.219	0.219	0.220
	718	0.220	0.220	0.233	0.285	0.285	0.286
4	166	0.300	0.584	0.632	0.638	0.638	0.653
	942	0.311	0.319	0.375	0.375	0.381	0.403
	969	0.311	3.788	4.109	4.109	4.109	4.112
5	286	0.405	0.413	0.413	0.413	0.413	0.413
	55	0.413	0.416	0.416	0.416	0.542	0.542
	549	0.415	2.629	2.935	3.078	3.078	3.078
6	253	0.503	0.508	0.508	0.508	0.509	0.509
	769	0.511	0.599	0.599	0.654	0.654	0.654
	651	0.515	0.543	0.543	0.543	0.543	0.543
7	830	0.600	0.600	0.602	0.602	0.617	0.620
	57	0.601	0.661	0.661	0.661	0.661	0.661
	453	0.601	1.043	1.043	1.043	1.129	1.129
8	22	0.701	0.704	1.510	1.510	1.510	1.510
	509	0.714	0.714	1.778	1.782	5.006	5.006
	450	0.714	0.884	0.932	0.932	0.932	0.932
9	230	0.802	0.802	0.802	0.802	0.802	0.805
	129	0.814	0.837	0.832	0.839	0.869	0.869
	426	0.821	0.825	0.825	0.841	0.841	0.841
10	869	0.900	4.027	4.205	4.205	4.857	4.857
	455	0.904	0.918	0.921	0.921	0.921	0.921
	707	0.906	0.906	0.912	0.917	0.917	0.917
11	489	1.004	3.007	3.007	3.007	3.007	3.007
	484	1.007	1.007	1.025	1.028	1.028	1.028

	112	1.020	1.243	1.279	1.284	1.294	1.340
12	781	1.110	1.114	1.118	1.435	1.435	1.435
	96	1.111	4.454	4.454	4.692	4.692	4.794
	898	1.116	1.122	1.148	1.153	1.153	1.153
13	290	1.204	1.214	2.267	2.267	2.267	2.293
	320	1.205	1.205	1.205	1.205	1.205	1.205
	429	1.209	1.210	1.228	1.228	1.228	1.228
14	348	1.302	1.302	1.302	1.302	1.313	1.313
	115	1.302	1.305	1.306	1.309	1.309	1.309
	560	1.303	1.321	1.321	1.321	1.321	1.322
15	311	1.403	1.407	1.407	1.407	1.407	1.408
	260	1.405	1.459	1.541	1.541	1.562	1.573
	466	1.407	1.412	1.413	1.413	1.413	1.416
16	735	1.507	1.538	1.538	1.545	1.545	1.545
	944	1.513	1.525	1.526	1.526	1.527	1.527
	875	1.520	1.524	1.524	1.729	1.729	1.729
17	99	1.600	1.614	1.642	1.642	1.642	1.642
	622	1.600	1.604	1.604	1.604	1.604	1.610
	428	1.612	1.616	1.616	1.623	1.623	1.623
18	160	1.702	1.713	1.713	1.713	1.713	1.713
	511	1.714	1.729	1.729	1.729	1.729	1.734
	180	1.714	2.087	2.087	2.087	2.087	2.087
19	727	1.806	1.806	1.807	1.809	1.809	1.811
	741	1.816	1.818	1.818	1.818	1.818	1.818
	335	1.830	1.830	1.830	1.831	1.831	1.831
20	966	1.904	1.904	1.925	1.925	1.945	1.945
	739	1.913	1.915	1.915	1.916	1.920	1.920
	700	1.914	1.914	1.914	1.914	1.914	1.914
21	523	2.011	2.011	2.016	2.016	2.016	2.018
	343	2.013	2.066	2.066	2.074	2.074	2.074
	855	2.014	2.016	2.017	2.018	2.018	2.018
22	116	2.107	2.483	2.483	2.483	2.483	2.483
	778	2.108	2.115	2.115	2.115	2.115	2.115
	733	2.111	2.116	2.120	2.121	2.121	2.121
23	593	2.201	2.201	2.216	2.216	2.216	2.216

	562	2.211	2.211	2.268	2.293	2.297	2.297
	637	2.213	2.213	2.214	2.214	2.214	2.214
24	150	2.302	2.302	2.310	2.310	2.310	2.310
	938	2.304	2.308	2.308	2.309	2.311	2.311
	657	2.305	2.380	2.380	2.388	2.389	2.391
25	337	2.401	2.406	2.406	2.406	2.406	2.406
	963	2.402	2.802	3.091	3.091	3.091	3.092
	422	2.407	2.449	2.449	2.458	2.458	2.461

In every sample, the value of $L(4, K)$ increases as K increases. When $L(8, 8) > 1.7$, $L(4, K)$ has a relatively small increase with K; specifically, the average of $L(4,70) - L(4,20) \leq 0.29$ for 15 samples considered. That is, $L(4, 20)$ is close to $\max_K L(4, K)$ when $L(4,20) > 1.7$. Samples with smaller $L(8, 8)$ need more RSPs for L_1. For example, sample 969 in the strata 4 had $L(8, 8) = 0.311$ and $L(4, 60) = 4.112$.

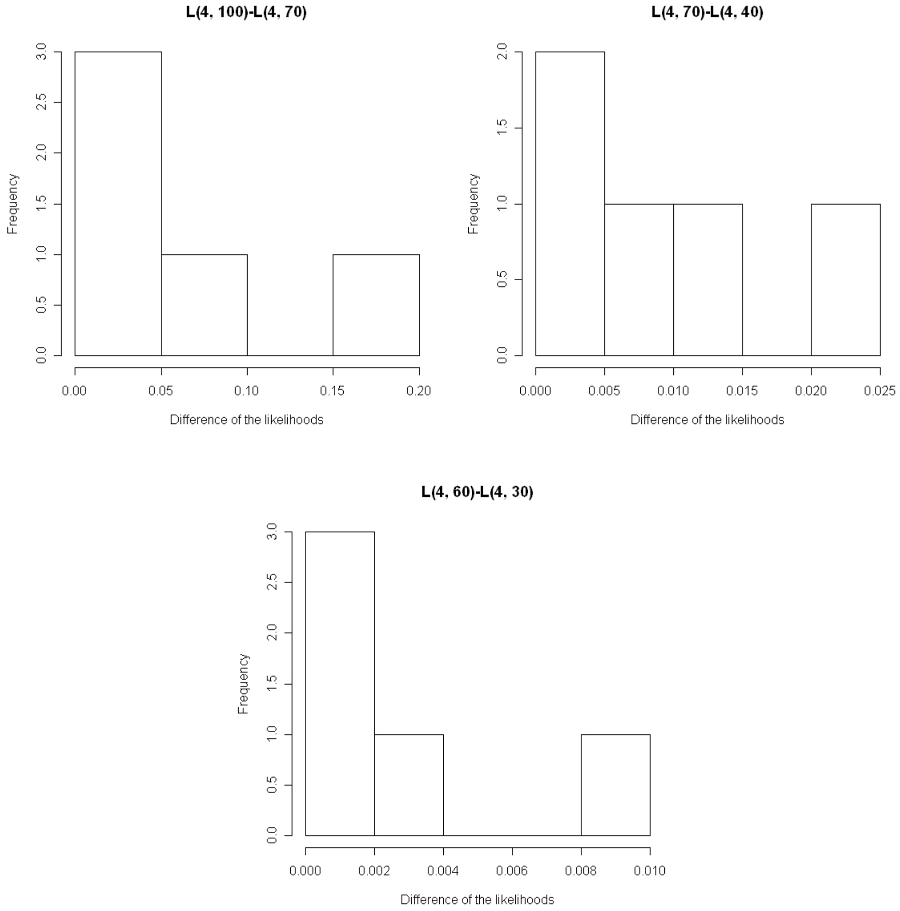
The LRTS computations in the simulations reported here always have 4 RSPs for L_0 and at least 20 RSPs for L_1. I then propose a conditional setting for K^*, the number of RSPs for L_1. When $L(4, 20)$ ≥ 1.7, I will use $L(4, 20)$ as the sample LRTS (that is, $K^* = 20$). When $1.1 \leq$ $L(4, 20)$ $<$ 1.7, I will use $L(4, 30)$. When $.2 \leq L(4, 20) < 1.1$, I will use $L(4, 40)$. When $L(4, 20) < .2$, I will use $L(4, 70)$.

I conducted a pilot study to confirm the properties of this algorithm. An additional one hundred samples of size 300 were generated from a single exponential distribution with mean 5 and censoring rate 30%. For each sample, I calculated $L(4, 20)$ and $L(4, K)$ for selected K. Only one sample had negative $L(4, 20)$; 6 samples had $0 \le L(4, 20) < 0.2$; 32 samples had $0.2 \le L(4, 20) < 1.1$; 8 samples had $1.1 \le L(4, 20) < 1.7$; and the remaining 53 had $1.7 \le L(4, 20)$. I select the first 5 samples from each interval for a total of 15 samples whose $L(4, K)$ values are shown in Table 5. I also report the difference $L(4, K^*+30) - L(4, K^*)$ and present the histograms of this value in Figure 2. For this sample, the choice of K^* is practical, as documented by mean difference $L(4, K^*+30) - L(4, K^*) \le 0.023$.

Table 5 Values of $L(4, K)$ for selected K; 15 samples

	Interval of $L(4, 20)$	Sample No.	$L(4, 20)$	$L(4, 70)$	$L(4, 100)$	$L(4, 100)$ - $L(4, 70)$	$L(4, 130)$	$L(4, 130)$ - $L(4, 100)$
1	<0.2, K*=100	57	-.006	1.385	1.385	0	1.385	0
2		99	.066	.080	.178	.098	.178	0
3		3	.091	.159	.191	.032	.191	0
4		60	.134	2.179	2.373	.194	2.373	0
5		71	.144	.302	.302	0	.302	0
				$L(4, 40)$	$L(4, 70)$	$L(4, 70)$ - $L(4, 40)$		
6	(0.2, 1.1), K*=40	46	.235	.392	.392	0		
7		1	.241	.248	.258	.01		
8		69	.256	.271	.286	.015		
9		23	.261	.261	.261	0		
10		72	.266	.859	.882	.023		
				$L(4, 30)$	$L(4, 60)$	$L(4, 60)$ - $L(4, 30)$		
11	(1.1, 1.7), K*=30	66	1.142	1.142	1.145	.003		
12		40	1.171	1.171	1.171	0		
13		94	1.228	1.231	1.241	.01		
14		83	1.269	1.269	1.277	.001		
15		56	1.270	1.270	1.270	0		

Figure 2 The histogram of likelihood of $L(4,\ K) - L(4,\ K^{*})$



6. Censoring Parameter Calculation

There are two censoring patterns in this simulation study: uniform and exponential censoring [16]. Additionally, there are three censoring rates: 10% and 40% in the simulation study of the null distribution and 10% and 30% in the power study. The use of 30% in the power study allows me to compare the power of the commingling test to the power of the mixture test [Ye et al.]. With 10% censoring rate, on average 10% of the data are censored during the course of the study (12 time periods with mean 5 periods) in each replication.

Let Y be the censoring time, X be the failure time, c be the end point of study and r be the censoring rate. Then, the censoring rate r is defined to be:

$$P(Y < X \mid \min(X,Y) < c) = r\,.$$

Since $Y < X$, this reduces to

$$P(Y < X \mid Y < c) = r\,.$$

From the definition of conditional probability,

$$\int_0^c\int_y^\infty f(x)g(y)\,dxdy = r\cdot P(\min(X,Y) < c)\,.$$

This equation can then be expressed as:

$$\int_0^c\int_y^\infty f(x)g(y)\,dxdy = r\cdot\{1 - P(\min(X,Y) > c)\}\,,$$

and

$$\int_0^c\int_y^\infty f(x)g(y)\,dxdy = r\cdot\{1 - P(X > c, Y > c)\}\,. \tag{3.5.1}$$

Equation (3.5.1) is the starting point in my calculation of censoring parameters.

CPC1: single exponential, uniform censoring

In this case X has an exponential distribution with mean μ, and Y has uniform distribution between 0 and l. That is,

$$f(x) = \frac{1}{\mu} e^{-\frac{x}{\mu}}$$

and

$$g(y) = \frac{1}{l}$$

where $l > c$. Equation (3.5.1) for this case is

$$\int_0^c \int_y^\infty \frac{1}{\mu} e^{-\frac{x}{\mu}} \cdot \frac{1}{l}\, dxdy = r \cdot \{1 - P(\min(X, Y) < c)\}.$$

The left hand side is

$$\int_0^c \int_y^\infty \frac{1}{\mu} e^{-\frac{x}{\mu}} \cdot \frac{1}{l}\, dxdy = \frac{\mu}{l}\left(1 - e^{-\frac{c}{\mu}}\right).$$

The right hand side is

$$r \cdot \left\{1 - P(\min(X, Y) < c)\right\} = r \cdot \left(1 - e^{-\frac{c}{\mu}} \cdot \frac{l - c}{l}\right).$$

Then,

$$l = \frac{-e^{-\frac{c}{\mu}}(\mu + c \cdot r) + \mu}{r \cdot \left(1 - e^{-\frac{c}{\mu}}\right)}. \qquad (3.5.2)$$

For example, to create a sample of size 100 from an exponential distribution with mean 5 with uniform censoring at the 10% rate, I generate one value from an exponential distribution with mean 5 and one value from U (0, 40.35). The value

40.35 is calculated from equation (3.5.2) with study duration 12 months. I then compare those two values and select the minimum as the reported value. If the uniform value is the one chosen, the observation is censored, and I set the censoring indicator on. I repeat this process independently 100 times.

CPC2: mixture of two exponentials, uniform censoring parameter

The random variable X is a mixture of two exponential with smaller mean μ_1 and mixing proportion π_1 and larger mean μ_2 and mixing proportion $\pi_2 = 1 - \pi_1$ so that the probability density function (pdf) of X is

$$f(x) = \pi_1 \frac{1}{\mu_1} e^{-\frac{x}{\mu_1}} + \pi_2 \frac{1}{\mu_2} e^{-\frac{x}{\mu_2}} .$$

Also pdf of Y is

$$g(y) = \frac{1}{l}$$

where $l > c$. Equation (3.5.1) is then

$$\int_0^c \int_y^\infty \left(\pi_1 \frac{1}{\mu_1} e^{-\frac{x}{\mu_1}} + \pi_2 \frac{1}{\mu_2} e^{-\frac{x}{\mu_2}} \right) \frac{1}{l} \, dxdy = r \cdot \{1 - P(X > c, Y > c)\}.$$

The left hand side is

$$\frac{1}{l} \int_0^c \pi_1 \cdot e^{-\frac{y}{\mu_1}} + \pi_2 \cdot e^{-\frac{y}{\mu_2}} \, dy = \frac{1}{l} \left(\pi_1 \mu_1 + \pi_2 \mu_2 - \pi_1 \mu_1 e^{-\frac{c}{\mu_1}} - \pi_2 \mu_2 e^{-\frac{c}{\mu_2}} \right).$$

The right hand side is

$$r\cdot\left\{1-\int_{c}^{\infty}\pi_1\frac{1}{\mu_1}e^{-\frac{x}{\mu_1}}+\pi_2\frac{1}{\mu_2}e^{-\frac{x}{\mu_2}}\,dx\int_{c}^{l}\frac{1}{l}\,dy\right\}=r\cdot\left\{1-\left(\pi_1\cdot e^{-\frac{c}{\mu_1}}+\pi_2\cdot e^{-\frac{c}{\mu_2}}\right)\cdot\frac{l-c}{l}\right\}.$$

Then

$$l=\frac{\pi_1\mu_1\left(1-e^{-\frac{c}{\mu_1}}\right)+\pi_2\mu_2\left(1-e^{-\frac{c}{\mu_2}}\right)-r\cdot c\left(\pi_1\cdot e^{-\frac{c}{\mu_1}}+\pi_2\cdot e^{-\frac{c}{\mu_2}}\right)}{r\cdot\left(1-\pi_1\cdot e^{-\frac{c}{\mu_1}}-\pi_2\cdot e^{-\frac{c}{\mu_2}}\right)}. \tag{3.5.3}$$

For example, to create a sample of size 100 from the mixture of two exponential distributions with the mean of difference is 1.0 (that is, the smaller mean is 0.5 and larger mean is 1.5) and equal proportion with uniform censoring pattern at the 10% censoring rate, I generate two values; one from an exponential distribution with mean 0.5 and the other one from an exponential distribution with mean 1.5. I choose one value with probability 1/2. I also generate one value from a *U* (0, 9.66). The value 9.66 is calculated from equation (3.5.3) with study duration 12 months. I then compare those two values and select the minimum as the reported value. If the uniform value is the one chosen, the observation is censored, and I set the censoring indicator on. I repeat this process independently 100 times.

CPC3: single exponential, exponential censoring

The random variable X has the exponential distribution with mean μ. The censoring time random variable Y also has an exponential distribution with mean μ_c. Then the pdf of X is

$$f(x) = \frac{1}{\mu} e^{-\frac{x}{\mu}},$$

and the pdf of Y is

$$g(y) = \frac{1}{\mu_c} e^{-\frac{y}{\mu_c}}.$$

Equation (3.5.1) for this case is

$$\int_0^c \int_y^\infty \frac{1}{\mu} e^{-\frac{x}{\mu}} \cdot \frac{1}{\mu_c} e^{-\frac{y}{\mu_c}} \, dx dy = r \cdot \{1 - P(X > c, Y > c)\}.$$

The left hand side is

$$\int_0^c e^{-\frac{y}{\mu}} \cdot \frac{1}{\mu_c} e^{-\frac{y}{\mu_c}} \, dy = \frac{\mu}{\mu + \mu_c} \left\{ 1 - e^{-\left(\frac{1}{\mu} + \frac{1}{\mu_c}\right)c} \right\}.$$

The right hand side is

$$r \cdot \left\{ 1 - \int_c^\infty \frac{1}{\mu} e^{-\frac{x}{\mu}} \, dx \int_c^\infty \frac{1}{\mu_c} e^{-\frac{y}{\mu_c}} \, dy \right\} = r \left\{ 1 - e^{-\left(\frac{1}{\mu} + \frac{1}{\mu_c}\right)c} \right\}.$$

Then,

$$\mu_c = \frac{(1 - r)}{r} \mu. \tag{3.5.4}$$

For example, to create a sample of size 100 from an exponential distribution with mean 5 and exponential censoring pattern with censoring rate 10%, I generate one value from an exponential distribution with mean 5, and generate one value from another independent exponential distribution with mean 45. The value 45 is calculated from equation (3.5.4) with study duration 12 months. I then compare those two values and select the minimum as the reported value. If the value from an exponential distribution with mean 45 is the one chosen, the observation is censored, and I set the censoring indicator on. I repeat this process independently 100 times.

CPC4: mixture of two exponentials, exponential censoring

The random variable X is a mixture of two exponential with smaller mean μ_1 and mixing proportion π_1 and larger mean μ_2 and mixing proportion $\pi_2 = 1 - \pi_1$ so that the pdf of X is

$$f(x) = \pi_1 \frac{1}{\mu_1} e^{-\frac{x}{\mu_1}} + \pi_2 \frac{1}{\mu_2} e^{-\frac{x}{\mu_2}} .$$

The censoring random variable Y has the exponential distribution with mean μ_c and pdf

$$g(y) = \frac{1}{\mu_c} e^{-\frac{y}{\mu_c}}.$$

Equation (3.5.1) for this case is

$$\int_0^c\int_y^\infty \left(\pi_1 \frac{1}{\mu_1} e^{-\frac{x}{\mu_1}} + \pi_2 \frac{1}{\mu_2} e^{-\frac{x}{\mu_2}}\right) \cdot \frac{1}{\mu_c} e^{-\frac{y}{\mu_c}}\, dxdy = r \cdot \left\{1 - P(X > c, Y > c)\right\}.$$

The left hand side is

$$\int_0^c \left(\pi_1 e^{-\frac{y}{\mu_1}} + \pi_2 e^{-\frac{y}{\mu_2}}\right) \frac{1}{\mu_c} e^{-\frac{y}{\mu_c}}\, dy$$

$$= \frac{1}{\mu_c}\left(\pi_1 \frac{\mu_1\mu_c}{\mu_1 + \mu_c} + \pi_2 \frac{\mu_2\mu_c}{\mu_2 + \mu_c} - \pi_1 \frac{\mu_1\mu_c}{\mu_1 + \mu_c} e^{-\left(\frac{1}{\mu_1}+\frac{1}{\mu_c}\right)c} - \pi_2 \frac{\mu_2\mu_c}{\mu_2 + \mu_c} e^{-\left(\frac{1}{\mu_2}+\frac{1}{\mu_c}\right)c}\right).$$

The right hand side is

$$r \cdot \left\{1 - \int_c^\infty \pi_1 \frac{1}{\mu_1} e^{-\frac{x}{\mu_1}} + \pi_2 \frac{1}{\mu_2} e^{-\frac{x}{\mu_2}}\, dx \int_c^\infty \frac{1}{\mu_c} e^{-\frac{y}{\mu_c}}\, dy\right\} = r \cdot \left\{1 - \left(\pi_1 e^{-\frac{c}{\mu_1}} + \pi_2 e^{-\frac{c}{\mu_2}}\right) e^{-\frac{c}{\mu_c}}\right\}.$$

Then, μ_c is a root of

$$\frac{1}{\mu_c}\left\{\pi_1 \frac{\mu_1\mu_c}{\mu_1 + \mu_c}\left(1 - e^{-\left(\frac{1}{\mu_1}+\frac{1}{\mu_c}\right)c}\right) + \pi_2 \frac{\mu_2\mu_c}{\mu_2 + \mu_c}\left(1 - e^{-\left(\frac{1}{\mu_2}+\frac{1}{\mu_c}\right)c}\right)\right\}$$

$$= r \cdot \left(1 - \pi_1 \cdot e^{-\left(\frac{1}{\mu_1}+\frac{1}{\mu_c}\right)c} - \pi_2 \cdot e^{-\left(\frac{1}{\mu_2}+\frac{1}{\mu_c}\right)c}\right). \tag{3.5.5}$$

For example, to create a sample of size 100 from the mixture of two exponential distributions with the mean of difference is 1.0 (that is, the smaller mean is 0.5 with proportion 50% and larger mean is 1.5) with uniform censoring pattern at censoring rate 10%, I generate two values; one from an exponential distribution with mean 0.5 and the other one from an exponential distribution with mean 1.5. I choose one value with probability 1/2. I also generate one value from an independent exponential distribution with mean 8.77. The value 8.77 is calculated from equation (3.5.4) with study duration 12 months. I then compare those two values and select the minimum as the reported value. If the value from an exponential distribution with mean 8.77 is the one chosen, the observation is censored, and I set the censoring indicator on. I repeat this process independently 100 times.

Chapter 4 Results

The null hypothesis is that the distribution of a sample of censored response times is a single exponential after Box-Cox transformation. The alternative hypothesis is that the distribution is a mixture of two exponential distributions after Box-Cox transformation.

The likelihood ratio test statistic (LRTS) is given by

$$d_n = -2\log \hat{L}_0 - \left(-2\log \hat{L}_1 \right),$$

where $\log \hat{L}_1$ is the log-likelihood function maximized under the alternative hypothesis and $\log \hat{L}_0$ is the log-likelihood function maximized under the null hypothesis.

1. Null Distribution of LRTS

I simulated the null distribution of d_n with: sample sizes $n = 300, 600, 900$; two censoring patterns (exponential and uniform), and two censoring rates: 10% and 40%. The family of distributions that I use to approximate the null distribution of d_n is

$$\pi_0 \chi_0^2 + (1 - \pi_0) \chi_{v_0}^2 ,$$

where π_0 is the fraction of zero LRTS values, and $\chi_{v_0}^2$ has probability density function (pdf)

$$f_{v_0}(x) = \frac{x^{\frac{v_0}{2}-1} e^{-\frac{x}{2}}}{\Gamma\left(\frac{v_0}{2}\right) 2^{\frac{v_0}{2}}} ,$$

where v_0 may be non-integer. That is, the distribution of non-zero LRTS values is

$$\Gamma\left(\frac{v_0}{2}, 2\right).$$

Tables 6 and 7 summarize the results of the simulation study of the null distribution of the LRTS. Operationally, any LRTS value that is less than or equal to 0.00001 is treated as a zero LRTS value. S denotes the sample size, R denotes the censoring rate, T denotes the transformation, L denotes the mean of LRTS values, V denotes the variance of LRTS values, F denotes the fraction of zero LRTS values, and NL denotes the mean of non-zero LRTS values in tables 6 and 7. The fraction of zero LRTS values is between 0 and 0.035 with average fraction of zero LRTS equal to 0.0072. The fraction of zero LRTS values is positively associated with the censoring rate (greater fraction of zeroes with increasing censoring rate, p<0.0001) but not associated with the sample size or censoring pattern ($R^2 = 0.449$). The mean of the non-zero LRTS values is between 1.88 and 2.53 with an average of 2.28. It decreases on average as n increases and the censoring rate increases. The fitted regression function

$$(3.76 \pm 0.51) - (0.72 \pm 0.28)\log(\log(n)) - (0.63 \pm 0.13)\,\text{censoring rate}$$

explains 46.7% of the variation in the mean of the non-zero LRTS ($p < 0.00003$). The mean of the non-zero LRTS values is not apparently associated with the censoring pattern ($p > 0.99$) or the power transformation ($p > 0.30$).

Table 6 Simulated Null Distribution with Exponential Censoring

S	R	T	L	V	F	N L	Percentile			
							75%	90%	95%	99%
300 (1000 rep.)	10%	¼	2.43	2.13	0	2.43	3.18	5.13	6.53	9.40
		½	2.38	2.07	0	2.38	3.34	4.96	6.27	9.93
		1	2.30	2.20	0.001	2.30	3.04	4.94	6.77	10.79
	40%	¼	2.39	2.24	0.002	2.39	3.15	5.08	6.62	10.15
		½	2.44	2.09	0.004	2.45	3.39	5.39	6.61	9.11
		1	2.23	2.06	0.001	2.24	2.97	4.95	6.43	9.77
600 (200 rep.)	10%	¼	2.37	2.08	0	2.37	3.08	4.40	7.00	10.63
		½	2.19	1.85	0	2.19	2.85	4.59	6.05	8.00
		1	2.36	2.26	0	2.36	2.86	4.89	6.49	10.78
	40%	¼	2.20	2.06	0.025	2.26	2.96	5.33	6.47	8.20
		½	2.14	1.78	0.01	2.16	3.41	4.81	5.70	6.69
		1	2.12	1.75	0	2.12	2.91	4.30	5.35	7.53
900 (200 rep.)	10%	¼	2.26	1.86	0.005	2.27	3.27	4.56	6.17	8.45
		½	2.50	2.14	0	2.50	3.31	5.12	6.44	9.80
		1	2.21	2.08	0.01	2.23	3.11	4.91	6.90	8.56
	40%	¼	2.33	2.08	0	2.33	3.14	5.34	6.51	8.90
		½	2.02	1.79	0.02	2.02	2.71	4.48	5.39	7.48
		1	1.97	2.10	0.02	2.01	2.63	4.13	5.06	9.72

Notes: S (sample size), R (censoring rate), T (transformation), L (mean of LRTS values), V (variance of LRTS values), F (fraction of zero LRTS values), NL (mean of non-zero LRTS values)

Table 7 Simulated Null Distribution with Uniform Censoring

S	R	T	L	V	F	NL	Percentile			
							75%	90%	95%	99%
300 (1000 rep.)	10 %	¼	2.44	2.06	0	2.44	3.38	5.11	6.51	9.55
		½	2.31	1.94	0	2.31	3.12	4.83	6.36	8.79
		1	2.38	2.06	0.001	2.39	3.26	4.97	6.47	9.20
	40 %	¼	2.29	2.79	0.006	2.31	3.01	4.84	6.44	9.47
		½	2.35	3.26	0.011	2.37	3.09	5.10	6.12	9.9
		1	2.18	2.08	0.008	2.20	3.01	4.66	5.99	9.59
600 (200 rep.)	10 %	¼	2.33	2.21	0	2.33	2.91	5.27	6.77	9.56
		½	2.52	2.42	0.005	2.53	3.44	5.59	7.01	11.96
		1	2.52	2.50	0	2.52	3.00	5.22	7.16	14.23
	40 %	¼	2.12	1.97	0.005	2.13	2.65	4.63	6.28	8.58
		½	1.92	1.80	0.025	1.97	2.64	3.98	6.15	8.01
		1	2.10	2.13	0.02	2.14	2.83	4.48	5.26	9.88
900 (200 rep.)	10 %	¼	2.51	2.27	0.005	2.52	3.20	5.45	7.61	11.08
		½	2.28	2.04	0	2.28	3.06	5.11	6.66	8.89
		1	2.34	2.26	0.005	2.35	3.15	5.36	7.24	9.56
	40 %	¼	1.88	1.84	0.02	1.88	2.60	3.92	5.80	8.02
		½	2.11	2.00	0.015	2.17	2.36	4.66	6.03	8.59
		1	2.09	2.17	0.035	2.16	3.14	4.65	5.70	8.22

Notes: S (sample size), R (censoring rate), T (transformation), L (mean of LRTS values), V (variance of LRTS values), F (fraction of zero LRTS values), NL (mean of non-zero LRTS values)

Table 8 contains the estimated 75th, 90th, 95th, and 99.9th percentiles for the sample sizes used in the simulation study. These percentiles explain 91.54% of the variance of the simulated percentiles in Table 6 and 7. Since the simulation of the null distribution uses sample size of 300, 600, 900 and censoring rate of 10%, 40%, I calculate the estimated percentiles for these settings. In the simulation study of the power, I use sample size 200 or 500 and censoring rates of 10% or 30% censoring rate so that I can compare the power of the LRTS with Box-Cox transformation to the power of the LRTS without Box-Cox transformation as simulated by Ye et al. [17] Consequently, I also add the estimation of the percentiles for these conditions in Table 8.

Table 8 Estimated Null Distribution Percentiles of LRT

Sample size	Censoring rate	75%	90%	95%	99%	99.9%
200	10%	1.70	3.22	4.43	7.74	11.65
	30%	1.61	3.09	4.28	7.17	11.45
300	10%	1.66	3.16	4.37	7.27	11.57
	40%	1.45	2.89	4.05	6.89	11.12
500	10%	1.61	3.10	4.30	7.19	11.47
	30%	1.52	2.97	4.15	7.01	11.26
600	10%	1.56	3.04	4.22	7.09	11.36
	40%	1.52	2.97	4.15	7.01	11.26
900	10%	1.60	3.08	4.27	7.16	11.44
	40%	1.42	2.84	4.00	6.82	11.05

2. Power Study of LRTS

I simulated the power of d_n using 500 replications for each of $3 \cdot 2^5$ settings:

two censoring patterns: exponential or uniform;

two sample sizes: 200 or 500;

two mixing proportions for the component with smaller mean: 0.50 or 0.85;

two differences between means: 1.0 or 1.5;

two censoring rates: 10% or 30%;

three transformation powers: 1 or 1/2 or 1/4.

Table 10 and 11 contain the results of the simulation study of the power of LRTS. S denotes the sample size, R denotes the censoring rate, T denotes the transformation, M denotes the mixing proportion, D denotes difference in means, L denotes the mean of LRTS values, F denotes the fraction of zero LRTS values, and NL denotes the mean of non-zero LRTS values in tables 10 and 11.

I estimate the power at the 0.01 level. The 50-50 mixture with difference of component means equals 1.5 (larger difference) and transformation 1/4 (smallest exponent) has power near 0.5 for sample size 200 and over 0.9 for sample size 500 for both censoring patterns. For skewed mixing proportions (with proportion in the smaller mean equal to 0.85) with greater difference of component means, the power increases with increasing sample size, as expected. For smaller difference of component means (difference equals 1.0) for both symmetric and skewed mixture, the power is low for both censoring patterns, both censoring rates, and both transformations.

To verify these impressions, I ran a multiple regression model with stepwise selection applied to the independent variables:

S: sample size (200 or 500),

P: censoring pattern (exponential or uniform),

M: mixing proportion (50-50 or 85-15),

R: censoring rate (10% or 30%),

D: difference between means (1.0 or 1.5),

T: transformation (1/4 or 1/2 or 1)

all two factor interactions,

and all sub-hierarchical interactions.

The dependent variable is the inverse normal cumulative density function (cdf) of the power of the LRT test using level of significance 1%. The fitted model is:

$$\begin{aligned}\Phi^{-1}\left(\hat{p}\right) &= -4.64 - 0.17\log S - 0.11P + 16.95R - 3.86D + 0.78M - 2.46(R\log S)\\ &\quad + 1.48(D\log S) - 0.10(M\log S) - 0.11(MR) - 0.03(MP) - 0.32(MD) - 3.89(RD)\end{aligned}$$

The R^2 for the fitted model is 0.96 and shows good fit. The transformation power T was not selected, and the logarithm of the sample size S and the censoring pattern P are not significant at the 0.05 level. The interaction between S and censoring rate R, the interaction between the censoring pattern P and the mixing proportion M and the difference between means are significant; that is, their p-values were between 0.0001 and 0.03. The mixing proportion M $\left(p < 10^{-13}\right)$, the $D \cdot \log S$ interaction $\left(p < 10^{-8}\right)$, censoring rate R $\left(p < 0.00001\right)$, and difference between means D $\left(p < 0.01\right)$ are significant.

I also used Ye et al. [17]'s estimated power as an independent variable in a bivariate regression analysis. The fitted model is:

$$\Phi^{-1}\left(\hat{p}\right) = \left(-1.43 \pm 0.05\right) + \left(0.46 \pm 0.02\right)\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right).$$

The R^2 for the fitted model of 0.8132 and does not fit as well as the more complete model. The implication is that the power of this commingling analysis is much lower than the power of the LRTS for a two component exponential mixture against an exponential distribution as shown by the negative intercept and the coefficient of $\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right)$ being 0.46, which is much less than one. Figure 3 shows the fitted power of the commingling analysis to the power of the LRTS for mixtures.

Additionally, there are significant adjustments due to sample size, censoring rate, censoring pattern, distance between means, skewness of the mixing proportions. To specify these adjustments, the independent variables $\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right)$, $\left\{\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right)\right\}^2$, and interactions were added to S, R, P, T, D, M. The fitted model is shown in Table 9.

Table 9 Regression model with the independent variables

$\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right), \left\{\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right)\right\}^2, S, R, P, T, D, M$, and interactions

Variable	Estimate	Std. Error	t value	$P(>\lvert t \rvert)$	
Intercept	-19.28	2.36	-8.20	10^{-11}	***
$\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right)$	-1.90	0.35	-5.37	10^{-6}	***
$\log S$	2.43	0.16	14.89	10^{-15}	***
R	18.84	4.62	4.08	0.00011	***
P	-0.36	0.10	-3.64	0.00049	***
T	-0.31	0.23	-1.38	0.17	
D	5.86	1.96	2.99	0.0038	**
M	1.09	0.34	3.20	0.00199	**
$\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right):R$	1.09	0.35	3.20	0.0028	**
$\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right):P$	0.04	0.02	1.79	0.078	.
$\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right):T$	-0.07	0.03	-2.07	0.042	*
$\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right):D$	0.95	0.35	2.70	0.008	**
$\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right):M$	0.07	0.03	2.31	0.02	*
$\log S:R$	-3.47	0.59	-5.91	10^{-7}	***
$\log S:M$	-0.17	0.02	-8.43	10^{-11}	***
$R:D$	-4.05	2.30	-1.76	0.082	.
$R:M$	0.60	0.16	3.66	0.00046	***
$P:T$	0.19	0.14	1.35	0.18	
$D:M$	-0.36	0.25	-1.42	0.16	

Significance codes: ‘***’ 0.001 ‘**’ 0.01 ‘*’0.05 ‘.’ 0.1 ‘ ’ 1

The R^2 for the fitted model is 0.97 and shows a good fit. The variable $\left\{\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right)\right\}^2$ was not selected and the transformation T is not significant at the 0.05 level, but its interaction with $\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right)$ is significant. The variables $\Phi^{-1}\left(\hat{p}_{\text{Ye et al.}}\right)$ $\left(p < 10^{-6}\right)$, the logarithm of sample size S $\left(p < 10^{-15}\right)$, the mixing proportion M $(p < 0.01)$, censoring rate R $(p < 0.001)$, censoring pattern P $(p < 0.001)$ and difference between means D $(p < 0.01)$ are significant; and the $R \cdot \log S$ interaction $\left(p < 10^{-7}\right)$ and the $M \cdot \log S$ interaction $\left(p < 10^{-11}\right)$ are also significant. Again, larger difference between means, more symmetric mixing proportion, and lower censoring rate are associated with greater power. Of course, increasing power is associated with increasing sample size.

Table 10 Power Results for 2^K Factorial Design for Exponential Censoring

S	R	T	M	D	L	F	NL	Power	Ye's Power
200 (500 rep.)	10 %	¼	0.50	1.0	3.94	0	3.94	0.282	NA
				1.5	11.41	0	11.41	0.53	NA
			0.85	1.0	3.22	0	3.22	0.034	NA
				1.5	4.35	0	4.35	0.084	NA
		½	0.50	1.0	3.98	0	3.98	0.054	NA
				1.5	10.30	0	10.30	0.428	NA
			0.85	1.0	2.88	0	2.88	0.012	NA
				1.5	4.83	0	4.83	0.104	NA
		1	0.50	1.0	3.84	0.002	3.85	0.046	0.51
				1.5	9.97	0.004	10.11	0.414	1
			0.85	1.0	2.93	0	2.93	0.02	0.06
				1.5	4.29	0.002	4.30	0.082	0.35
	30 %	¼	0.50	1.0	3.65	0	3.65	0.038	NA
				1.5	9.35	0.002	9.43	0.426	NA
			0.85	1.0	2.81	0	2.81	0.018	NA
				1.5	2.99	0	2.99	0.02	NA
		½	0.50	1.0	3.40	0	3.40	0.042	NA
				1.5	7.18	0.05	8060	0.302	NA
			0.85	1.0	2.75	0	2.75	0.016	NA
				1.5	3.13	0	3013	0.032	NA
		1	0.50	1.0	3.40	0	3.40	0.042	0.33
				1.5	7.16	0.05	8.62	0.324	1
			0.85	1.0	2.47	0.012	2.50	0.026	0.04
				1.5	3.17	0.002	3.17	0.034	0.15
500 (500 rep.)	10 %	¼	0.50	1.0	6.65	0	6.65	0.216	NA
				1.5	23.64	0	23.64	0.938	NA
			0.85	1.0	3.96	0.002	3.97	0.064	NA
				1.5	8.10	0	8.10	0.29	NA
		½	0.50	1.0	6.29	0	6.29	0.208	NA
				1.5	23.19	0	23.19	0.958	NA

			0.85	1.0	3.44	0.006	3.46	0.054	NA
				1.5	8.93	0	8.93	0.388	NA
		1	0.50	1.0	6.35	0	6.35	0.206	0.94
				1.5	23.01	0	23.01	0.932	1
			0.85	1.0	3.28	0.006	3.30	0.05	0.24
				1.5	7.69	0.006	7.73	0.292	0.72
	30 %	¼	0.50	1.0	5.27	0	5.27	0.116	NA
				1.5	19.46	0	19.46	0.9	NA
			0.85	1.0	2.61	0.004	2.62	0.016	NA
				1.5	4.40	0	4.40	0.086	NA
		½	0.50	1.0	4.97	0	4.97	0.122	NA
				1.5	18.11	0.024	19.41	0.83	NA
			0.85	1.0	2.58	0.008	2.60	0.012	NA
				1.5	4.12	0.004	4.13	0.076	NA
		1	0.50	1.0	4.40	0.002	4.41	0.098	0.76
				1.5	16.89	0.03	18.12	0.794	1
			0.85	1.0	2.37	0.014	0.40	0.016	0.1
				1.5	3.81	0.006	3.84	0.076	0.38

Notes: S (sample size), R (censoring rate), T (transformation), M (mixing proportion), D (difference in means), L (mean of LRTS values), F (fraction of zero LRTS values), NL (mean of non-zero LRTS values)

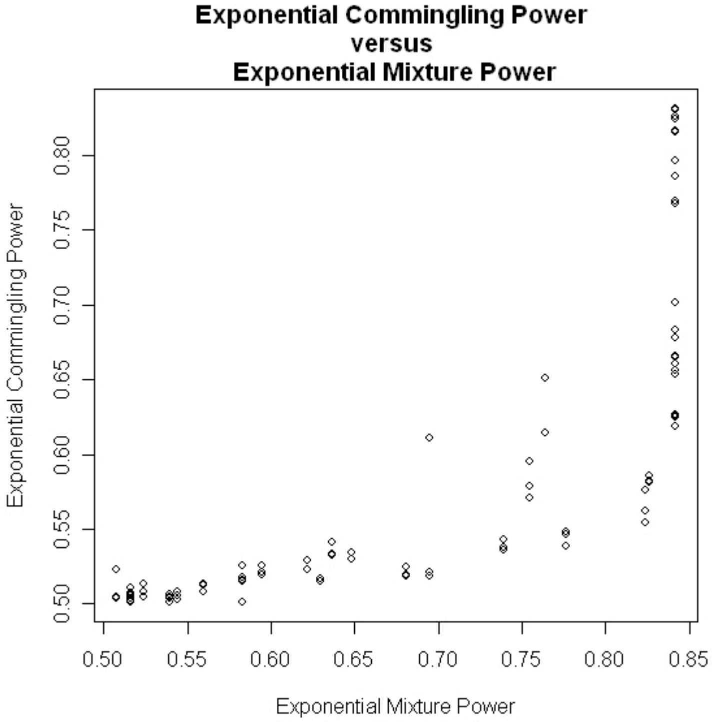
Table 11 Power Results for 2^K Factorial Design for Uniform Censoring

S	R	T	M	D	L	F	NL	Power	Ye's Power
200 (500 rep.)	10 %	¼	0.50	1.0	3.94	0	3.94	0.048	NA
				1.5	10.92	0	10.92	0.476	NA
			0.85	1.0	2.65	0.01	2.65	0.004	NA
				1.5	3.90	0	3.90	0.058	NA
		½	0.50	1.0	3.94	0.002	3.95	0.046	NA
				1.5	9.82	0.02	10.29	0.396	NA
			0.85	1.0	2.58	0.002	2.59	0.004	NA
				1.5	3.98	0.002	3.99	0.072	NA
		1	0.50	1.0	3.89	0.002	3.90	0.062	0.47
				1.5	10.25	0.01	10.49	0.462	1
			0.85	1.0	2.62	0.006	2.64	0.01	0.1
				1.5	4.16	0.004	4.18	0.074	0.31
	30 %	¼	0.50	1.0	3.56	0	3.56	0.004	NA
				1.5	9.33	0.002	9.39	0.402	NA
			0.85	1.0	2.36	0	2.36	0.01	NA
				1.5	2.49	0	2.49	0.01	NA
		½	0.50	1.0	4.62	0	4.62	0.064	NA
				1.5	6.15	0.1	8.37	0.32	NA
			0.85	1.0	4.60	0	4.60	0.058	NA
				1.5	2.54	0.006	2.64	0.014	NA
		1	0.50	1.0	3.76	0.006	3.89	0.044	0.21
				1.5	3.42	0.168	10.82	0.318	1
			0.85	1.0	2.61	0	2.61	0.012	0.02
				1.5	2.41	0.002	2.41	0.006	0.04
500 (500 rep.)	10 %	¼	0.50	1.0	6.16	0.002	6.17	0.192	NA
				1.5	23.90	0	23.90	0.938	NA
			0.85	1.0	3.21	0.15	3.25	0.04	NA
				1.5	6.53	0.002	6.54	0.198	NA
		½	0.50	1.0	5.76	0	5.76	0.156	NA
				1.5	23.36	0	23.36	0.956	NA
			0.85	1.0	3.21	0.006	2.23	0.038	NA

				1.5	4.51	0	4.51	0.242	NA
		1	0.50	1.0	5.69	0	5.69	0.136	0.93
				1.5	23.5	0	23.5	0.94	1
			0.85	1.0	2.92	0.026	3.00	0.04	0.21
				1.5	6.11	0.002	6.12	0.178	0.69
	30 %	¼	0.50	1.0	4.88	0.006	0.91	0.09	NA
				1.5	20.43	0	20.43	0.902	NA
			0.85	1.0	2.19	0.002	2.20	0.004	NA
				1.5	2.64	0.006	2.66	0.008	NA
		½	0.50	1.0	4.61	0.004	4.76	0.094	NA
				1.5	10.54	0.146	19.82	0.736	NA
			0.85	1.0	2.12	0.01	2.14	0.004	NA
				1.5	2.64	0.006	2.66	0.014	NA
		1	0.50	1.0	4.80	0	4.80	0.108	0.64
				1.5	8.24	0.142	18.46	0.732	1
			0.85	1.0	2.23	0.012	2.26	0.004	0.04
				1.5	2.63	0.016	2.66	0.02	0.11

Notes: S (sample size), R (censoring rate), T (transformation), M (mixing proportion), D (difference in means), L (mean of LRTS values), F (fraction of zero LRTS values), NL (mean of non-zero LRTS values)

Figure 3 The power of the commingling analysis versus the power of the LRTS for mixtures



Chapter 5 Discussion and Conclusions

The null distribution of exponential commingling LRTS is insensitive to censoring pattern and the transformation. The null distribution if this LRTS is well approximated by

$$\pi_0\chi_0^2 + (1-\pi_0)\chi_{\nu_0}^2 ,$$

where π_0 is the fraction of zero LRTS values. The fraction of zero LRTS values is positively associated with the censoring rate but not associated with the sample size or censoring pattern. The mean of the non-zero LRTS values, ν_0 , is approximated by $(3.76 \pm 0.51) - (0.72 \pm 0.28)\log(\log(n)) - (0.63 \pm 0.13)\,\text{censoring rate}$. The mean of the non-zero LRT values is not apparently associated with the censoring pattern or the power transformation. The null distribution of the exponential commingling LRTS, therefore, is apparently dependent only on the sample size and censoring rate.

Holding the difference in means, censoring rate, and censoring pattern constant, power is greater for 50-50 mixtures than skew mixtures. As expected, power increases as the difference of the component means increases. The power of the exponential commingling LRTS is less than the power of the exponential mixture LRTS.

I asked four questions in chapter 1:

1. How does one obtain MLEs for the exponential commingling analysis?

Software is in the appendix that calculates these MLEs. A large number of RSPs is needed to obtain the correct maximized likelihood function.

2. What is an effective method for maximization of the exponential commingling likelihood function?

The EM algorithm is practical for the estimation of the transformed mixture parameters, and the Newton-Raphson algorithm is effective for estimating the transformation parameter.

3. Is there an invariant null distribution?

Yes, the null distribution is apparently a function only of sample size and censoring rate.

4. Does the test have reasonable power?

While the power of the exponential commingling LRTS is less than the power of the exponential mixture test, sample sizes necessary to have high power are within the ranges of current genetic studies.

Normal distribution commingling analysis of cross-sectional data has long been an important tool for identifying variables that may have a genetic determinant. The application of exponential commingling analysis to the study of length of time to recovery or other survival analysis applications has wide-ranging implications in human genetics because of the importance of genetic heterogeneity. For many age-dependent conditions, including nicotine dependence, bipolar disorder and Alzheimer's disease, there is strong suspicion of genetic mixture mechanisms. This extension of the normal distribution commingling analysis is a first step in the effort to develop sufficiently flexible models to address these problems. In addition, the importance of heterogeneous responses in clinical trials has become ever more apparent. For example, recommendations concerning optimal dosing and drug treatment regimens could be influenced by the findings of analyses.

Future research

The purpose of commingling analysis is to make the inference of a mixture mechanism that might have a genetic origin more robust. The exponential commingling analysis developed here, for example, extends the application of exponential survival analysis procedures to data which is from a Weibull distribution. Specifically, data from a Weibull distribution can be transformed to the exponential form using the transformations of this research. While I have shown that the exponential commingling analysis has a well-defined null distribution for a range of transformations, I have not completed a simulation study of the rejection rate of exponential mixture analysis for samples drawn from a Weibull distribution. I conjecture that the rejection rate is not constant across Weibull distributions and propose to study this issue in future work.

Peng et al. [18] proposed a mixture model using the generalized *F* (GF) distribution family. The GF mixture model can relax the usual stronger distributional assumptions and may allow the analyst to uncover structure in the data that might otherwise be missed. The GF model has the advantage of being very flexible and including many commonly used distributions as special cases such as the lognormal, Weibull, gamma, and log-logistic distributions. I propose to extend the exponential commingling analysis to the GF model in future work.

My research questions for this study will be:

1. What are the rejection rates of the GF commingling mixture LRTS when the sample is from a single component distribution that is from the GF family?
2. What are the power properties of the GF commingling LRTS?

References

[1] C.J. Maclean, N.E. Morton, R.C. Elston, and S. Yee, Skewness in Commingled Distributions. Biometrics 32 (1976) 695-699.

[2] A.C. Viswanathan, R.A. Hitchings, A. Indar, P. Mitchel, P.R. Healey, P. McGuffin, and P.C. Sham, Commingling analysis of intraocular pressure and glaucoma in an older Australian population. Annals of Human Genetics 68 (2004) 489-497.

[3] Y. Bosse, L. Perusse, and M.C. Vohl, Genetics of LDL particle heterogeneity: from genetic epidemiology to DNA-based variations. Journal of Lipid Research 45 (2004) 1008-1026.

[4] D.E. Ross, G.K. Thaker, R.W. Buchanan, B. Kirkpatrick, A.C. Lahti, D. Medoff, J.J. Bartko, J. Goodman, and A. Tien, Eye tracking disorder in schizophrenia is characterized by specific ocular motor defects and is associated with the deficit syndrome. Biological Psychiatry 42 (1997) 781-796.

[5] G.E.P. Box, and D.R. Cox, An Analysis of Transformations. Journal of the Royal Statistical Society. Series B (Methodological) 26 (1964) 211-252.

[6] G. Casella, and R. Berger, Statistical Inference, Duxbury Advanced Series, 2001.

[7] Y.M. Ning, and S.J. Finch, The null distribution of the likelihood ratio test for a mixture of two normals after a restricted Box-Cox transformation. Communications in Statistics-Simulation and Computation 29 (2000) 449-461.

[8] Y.M. Ning, and S.J. Finch, The likelihood ratio test with the Box-Cox transformation for the normal mixture problem: Power and sample size study. Communications in Statistics-Simulation and Computation 33 (2004) 553-565.

[9] J. Klein, and M. Moeschberger, Survival Analysis Techniques for Censored and Truncated Data, 2002.

[10] A.P. Dempster, N.M. Laird, and D.B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, Blackwell Publishing for the Royal Statistical Society, 1977, pp. 1-38.

[11] S.G. Self, and K.-Y. Liang, Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests Under Nonstandard Conditions, Journal of the American Statistical Association, 1987, pp. 605-610.

[12] Cramer, Mathematical Methods of Statistics, Princeton University Press, 1946.

[13] G. McLachlan, and D. Peel, Finite Mixture Models, Wiley Series in Probability and Statistics, 2001.

[14] X. Liu, C. Pasarica, and Y.Z. Shao, Testing homogeneity in gamma mixture models. Scandinavian Journal of Statistics 30 (2003) 227-239.

[15] Hogg, McKean, and Craig, Intorduction to Mathematical Statistics, Pearson Education, 2005.

[16] Y. Peng, K.B.G. Dear, and K.C. Carriere, Testing for presence of cured patients: a simulation study. Statistics in Medicine 20 (2001) 1783-1976.

[17] Q. Ye, S.L. Cook, W. Kim, G.A. Chase, D. Gordon, and S.J. Finch, Null distribution and Power Function of the Likelihood Ratio Test of Survival Distribution as Mixture of Two Exponential Components against Single Exponential. Statistics in Medicine.

[18] Y. Peng, K.B.G. Dear, and J.W. Denham, A Generalized *F* Mixture Model for Cure Rate Estimation. Statistics in Medicine 17 (1998) 813-830.

Appendix

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>            // define time()
#include <math.h>

// global variables
// you can use them in all the functions in this file
// constant
const int n = 900; // no of observations
int sim = 100; // no of replicates
const int r = 50; //no of RSPs
const int ar = 30 ; //no of RSPs for alternative
const int nr = 4 ; //no of RSPs for null
//const float pp1 = 0.5; // the probability of the first component
//const float pp2 = 1 - pp1;
const int m1 = 5; // the mean of the 1st component
//const int m2 = 8; // the mean of the 2nd component
const double m3  = 7.5; //censoring rate (m3=1/c) 131.97
const double lam_inv = (double) (1.0/4.0); // true lambda
double c[n]; // vector of censoring indicator
double x[n]; // vector of observation
//double z[n];
//double w[n];
double ox[n]; //ordered observation
// updated
double mmm[2]; // the mean to be updated in EM algorithm
double ppp[2]; // the mixing proportion to be updated in EM algorithm
double m[2]; // the mean to be updated in EM algorithm in the next step
double p[2]; // the mixing proportion to be updated in EM algorithm in the next step
double loglik; // loglikelihood
double mloglik; // maximum loglikelihood

int do_readin = 0;//read and do not save the data when 1, generate and save the data when 0
```

```
}


char fn_in[] = "data900_10_4_u.txt"; // name of the data file to be read

/************************************************************************************/

void     char_str_to_num_ary( char *, int *, int * );
double random1(void);


// Uniform random number generation between 0 and 1
double random1()
{
    int x = rand() ;
    double y = ((double)x)/((double)RAND_MAX) ;
    return y ;
}
// Absolute value function
double myAbs(double x)
{
    if (x < 0) return -x;
    else return x;
}
// Summation fucntion
double sum_d( int nn, double *x )
{
    int i;
    double sum=0;
    for( i=0 ; i<nn ; i++ )
        sum += x[i];
    return  sum;
}
// Mean function
double mean_d(int nn, double *x)
{
    return  sum_d( nn, x ) / ((double) nn);
}
```

```
//y = random();

//void random(double* y, double* z)
//{
//      int x = rand() ;
//      *y = ((double)x)/((double)RAND_MAX) ;
//      *z=*y+1;
//}
//double y,z;
//random(&y,&z);
//Bernoulli distribution func.
int bernoulli( double p)
{
     int x;
     x = 0 ;
     double y = random1();
          if(y < p) {x  = 1;}
     return x ;
}


// Binomial distribution func.
int binomial(int nn, double p)
{
     int x;
     x = 0 ;
     for(int i=0; i < nn; i++){
          x += bernoulli(p);
     }
     return x ;
}

/***************************************************************************************/
  // Function of EM algorithm
 double emalgo(double ep1, double ep2, double em1, double em2, double *w ) // update mm and
```

```
        pp
{
    double t1[n], t2[n];
    double div, divr, div2, divr2;
    double arg1, arg2, divt1, divrt1, divt2, divrt2;
    double t1mean ;
    double mm[2];
    double pp[2];
    int j, n_iter;
    //double t1sum = 0.0;
        /*for(int k=0; k<n; k++){
                t1[k] = 0;
                t1[k] = random1();
                t1sum += t1[k];
                t2[k] = 1-t1[k];
        }*/

        // printf("m1 = %10.8f m2 = %10.8f \n", ma1, ma2);
        // printf("ip1 = %10.8f \n",op);
        m[0] = em1;
        m[1] = em2;
        p[0] = ep1;
        p[1] = ep2;
        mmm[0] = m[0];
        mmm[1] = m[1];
        ppp[0] = p[0];
        ppp[1] = p[1];
        mm[0] = 1.0;
        mm[1] = 10.0;
        pp[0] = 0.2;
        pp[1] = 0.8;
        n_iter = 0;

        //double powsum = pow(mm[0]-mmm[0],2) + pow(mm[1]-mmm[1],2) + pow(pp[1]-
       ppp[1],2) + pow(pp[0]-ppp[0],2);
       /* for ( int j=0; j<n; j++){
```

```
 printf("c[%d] = %10.8f   \n", j,c[j]);
 }*/
 while(  (pow(mm[0]-mmm[0],2)  +  pow(mm[1]-mmm[1],2)  +  pow(pp[1]-ppp[1],2)  +
pow(pp[0]-ppp[0],2) > pow(10,-5) ) & (n_iter <= 1000)){


            /* double p1 = t1sum/((double) n);
        double p2 = 1-p1;
        pp[0] = p1;
        pp[1] = p2;*/
        //printf("m0 = %10.8f m1 = %10.8f \n", m[0], m[1]);
     // printf("pp0 = %10.8f pp1 = %10.8f \n", pp[0], pp[1]);
      for (j=0;j<n;j++) {
            //t1[j] = 0;
               t1[j] = 1;
               if (m[0] < m[1]){
                     t1[j] = 0;
                     arg1 = (p[0] * (1/m[0]) * exp(-w[j]/m[0])) + (pp[1] * (1/m[1]) * exp(-
w[j]/m[1]) ) ;
                     arg2 = p[0] * (exp(-w[j]/m[0])) + p[1] * (exp(-w[j]/m[1]));
                     divt1 =  p[0] * (1/m[0]) * exp(-w[j]/m[0]) ;
                     divrt1 =  ( p[0] * (1/m[0]) * exp(-w[j]/m[0])) + ( p[1] * (1/m[1]) *
exp(-w[j]/m[1]) );
                     divt2 = p[0]*(exp(-w[j]/m[0]));
                     divrt2 = (p[0]*(exp(-w[j]/m[0]))+p[1]*(exp(-w[j]/m[1])));
               }
               if ( arg1 > 0.00001 &  arg2 > 0.00001){
                     t1[j] =  c[j]*(divt1/divrt1) + (1-c[j])*(divt2/divrt2);
                     t2[j] = (1 - t1[j]);
                  //printf("t1 changed..\n");
         }
         }

      n_iter = n_iter+1;
      mmm[0] = mm[0];
      mmm[1] = mm[1];
      div = 0.0;
```

```
divr = 0.0;

for ( j=0;j<n;j++){
      div += (t1[j] * w[j]);
      divr += (t1[j] * c[j]);
      //printf("t1[%d] = %10.8f \n", j,t1[j]);
      //printf("t1[%d] = %10.8f w[%d] = %10.8f \n", j,t1[j], j,w[j]);
}
mm[0] = div/divr;
if (mm[0] >= 500){mm[0] = 500;}
if (isnan(mm[0] == 1)){mm[0] = 500;}

div2 = 0.0;
divr2 = 0.0;
for ( j=0;j<n;j++){
      div2 += t2[j] * w[j];
      divr2 += t2[j] * c[j];
      //printf("t2[%d] = %10.8f w[%d] = %10.8f \n", j,t2[j], j,w[j]);
//printf("div2 = %10.8f divr2 = %10.8f \n", div2,divr2);
}
mm[1] = div2/divr2;
if (mm[1] >= 500){mm[1] = 500;}
if (isnan(mm[1] == 1)){mm[1] = 500;}

m[0] = mm[0];
m[1] = mm[1];

ppp[0] = pp[0];
ppp[1] = pp[1];
//printf("m0 = %10.8f m1 = %10.8f \n", m[0],m[1]);
t1mean = mean_d(n, t1);
if ( t1mean >= 0 )
      pp[0] = t1mean;
else pp[0] = 0;
if ( t1mean > 1 )
      pp[0] = 1;
```

```
            else pp[0] = t1mean;
            pp[1] = 1-pp[0];
            p[0] = pp[0];
            p[1] = pp[1];
            //printf("p0 = %10.8f p1 = %10.8f \n", p[0],p[1]);

    } //end of while

} //End of emalgo()

/*******************************************************************************/

// Function g is the derivative of Log-likelihood
double g(double lam, double *z)
{
    double g;
    double divg1, divrg1, divg2, divrg2;
    double r1, r2, r3, r4;
    double term1, term2, term3, term4;
    int i;
    r1 = 0.0;
    r2 = 0.0;
    r3 = 0.0;
    r4 = 0.0;
    g = 0.0;
        for ( i=0; i<n; i++ ){
            //printf("pp[0]=        %8.6f\t        mm[0]=        %8.6f\t        z[i]=        %8.6f\t
        lam= %8.6f\t\n",pp[0],mm[0],x[i],lam);
            r1 = p[0]/m[0];
            //printf("=========pow result= %8.6f\n",-pow(z[i],lam));
            r2 =  (-pow(z[i],lam))/m[0] ;
            r3 = p[1]/m[1];
            r4 = (-pow(z[i],lam))/m[1];
            //printf("r1= %8.6f\t r2= %8.6f\t r3= %8.6f\t r4= %8.6f\t\n",r1,r2,r3,r4);
            divg1 =  r1*exp(r2)*r2*log(z[i]) +  r3*exp(r4)*r4*log(z[i]);
            divrg1 = r1 * exp(r2) + r3*exp(r4);
```

```
              divg2 = p[0]*exp(-pow(z[i],lam))*r2*log(z[i]) + m[1]*exp(-pow(z[i],lam))*r4*log(z[i]);
              divrg2 = p[0]*exp(r2) + p[1]*exp(r4);
              //printf("mm0g = %10.8f mm1g = %10.8f \n", mm[0],mm[1]);
              //printf("divg1=      %8.6f\t      divrg1=      %8.6f\t      divg2=      %8.6f\t
         divrg2= %8.6f\t\n",divg1,divrg1,divg2,divrg2);
              term1 = c[i]*(divg1/divrg1);
              term2 = (1-c[i])*(divg2/divrg2);
              term3 = (c[i]/myAbs(lam));
              term4 = c[i] * log(z[i]);
              //printf("===========divg2/divrg2=%8.6f\n",divg2/divrg2);
              //printf("===========x[i]=%8.6f\t      abs(x[i])=%8.6f\t      log(x[i])=%8.6f\n",x[i],
         myAbs(x[i]), log(x[i]));
              //printf("c[i]==   %8.6f\t   term1=   %8.6f\t   term2=   %8.6f\t   term3=   %8.6f\t
         term4= %8.6f\t\n",c[i], term1,term2,term3,term4);
              g +=  term1 + term2 + term3 + term4;
         }
      return g;
} //End of derivative Function

/*************************************************************************************/
// Maximum loglikelihood calculation under alternative
// before using maxlam pp and mm should be updated using em
double maxlam(double lambda, double *z, double *w)
{
      double op, op2, oxsum, oxsumb, ma1, ma2, lo1, lo2;
      int i, k, n1, n2;
//Section A generates the RSPs for the component means and mixing proportions with
         transformation //parameter lambda fixed for this iteration.
            for(k=0;k<r;k++){
              op = 0;
              n1 = 0;
                  op = random1();
              n1 = binomial(n, op);
              while( n1 == 0 ){n1 = binomial(n,op);}
              while( n1 == n ){n1 = binomial(n,op);}
              //printf("n1 = %d  \n",n1);
```

```
        //printf("n = %d   \n",n);
        n2 = n - n1;
        //printf("n2 = %d   \n",n2);
        oxsum = 0;
        for (i=0;i<n1;i++){
            oxsum += ox[i];
        }
        oxsumb = 0;
        for (i=n1;i<n;i++){
            oxsumb += ox[i];
        }
        ma1 = oxsum/((double) n1);
        ma2 = oxsumb/((double) n2);
        op2 = 1-op;
        //double w[n];
        //printf("ip1 = %10.8f \n",op);
        //printf("n1 = %d n2 = %d \n",n1,n2);
        //printf("oxsum = %10.8f oxsumb = %10.8f \n",oxsum, oxsumb);
        //printf("ma1 = %10.8f ma2 = %10.8f \n",ma1, ma2);

        for (i = 0; i<n ;i++){
            w[i]=0;
            w[i] = pow(z[i], lambda);
            //printf("w[%d] = %10.8f z[%d] = %10.8f \n", i,w[i], i,z[i]);
        }
    //Section B applies the EM algorithm detailed in section 3.3 of dissertation
        emalgo(op, op2, ma1, ma2, w);
        //printf("ip1 = %10.8f \n",op);
        //printf("p0 = %10.8f p1 = %10.8f \n",p[0],p[1]);
        mloglik = -pow(10,10);
        loglik = 0;
        lo1 = 0;
        lo2 = 0;
        for (i=0;i<n;i++){
            lo1 = (log((p[0]/m[0])*exp(-w[i]/m[0])+(p[1]/m[1])*exp(-
    w[i]/m[1]))+log(myAbs(lambda))+(lambda-1)*log(z[i]));
```

```
                        lo2 = (log(p[0]*(exp(-w[i]/m[0]))+p[1]*(exp(-w[i]/m[1]))));
                        //printf("mm0m = %10.8f mm1m = %10.8f \n", mm[0],mm[1]);
                        loglik += c[i]*lo1 + (1-c[i])*lo2;
                        //printf("loglik = %10.8f \n", loglik);
                }

                if (isnan(loglik)==1){loglik = (-pow(10,10));}
                //if (k==0) {mloglik = loglik;}

                if (mloglik < loglik) { mloglik = loglik;}
                //printf(" mloglik = %10.8f \n", mloglik);

        } //End of RSPs
//Subroutine then goes to main program
return mloglik;
}

/********************************************************************************/
// Maximum loglikelihood calculation under null
double naxlam(double lambda, double *z, double *w, double *pmu)
{
    double nloglik, loglik0;
    double loc; //summation of censoring indicator
    double low; //summation of transformed data
    double mu;
    int i, j;

                nloglik = -pow(10,10); //calculating log likelihood of null distn
                for ( i=0;i<n;i++){
                        w[i] = 0.0;
                        w[i] = pow(z[i], lambda);
                }
                loc = 0.0;
                low = 0.0;
                mu = 0.0;
                for ( j=0;j<n;j++){
```

```
                    loc += c[j];
                    low += w[j];
                }
                mu = low/loc;
                loglik0 = 0.0;
                for (i=0;i<n;i++){
                    loglik0 += c[i]*(log(loc/low) + log(myAbs(lambda)) + (lambda-1)*log(z[i])) -
            w[i]*(loc/low) ;
                    //printf("loglik0 = %10.8f \n", loglik0);
                }

                if (isnan(loglik0)==1){loglik0 = (-pow(10,10));}

                if (nloglik < loglik0) { nloglik = loglik0;}
                //printf(" nloglik = %10.8f   mu = %10.8f \n",  nloglik, mu);
                *pmu = mu;
        return nloglik;
}//End of naxlam function

/**************************************************************************************/
//ng Function is derivative of Log-likelihood under null
double ng(double lam, double *z)
{
        double nf;
        double term1;
        double term2;
        int i, j;
            term1 = 0.0;
            term2 = 0.0;
            for ( j=0; j<n; j++ ){
                term1 += pow(z[j],lam)*log(z[j]);
                term2 += pow(z[j],lam);
                //printf("c[i]==    %8.6f\t    term1=    %8.6f\t    term2=    %8.6f\t    term3=    %8.6f\t
            term4= %8.6f\t\n",c[i], term1,term2,term3,term4);
            }
            nf = 0.0;
```

```
            for (i=0; i<n; i++){
                 nf +=   c[i]*(-(term1/term2) +   1/myAbs(lam) + log(z[i]));
            }
       return nf;
} //End of ng Function
/*********************************************************************************/
//Main program
       int main()
       {

       double u1[n];
       //double u2[n];
       double u3[n];
       //double u4[n];
       double x1[n];
       //double x2[n];
       //double x3[n];
       double x4[n];
       double z[n];
       double w[n];
       double lrts;
       double del, lam, lam_new, mloglik0, mloglik1;
       double maxloglik, maxlambda, maxp1, maxp2, maxm1, maxm2, temp;
       double mmaxlambda, mmaxp1, mmaxp2, mmaxm1, mmaxm2;
       double mmaxloglik;
       double mu, mmu, nnaxloglik, nnaxlambda;
       double r1, r2;
       double ndel, nlam, nlam_new, nloglik0, nloglik1;
       double naxloglik, naxlambda, ntemp;
       double nr1, nr2;
       int i, j, jj, k, kk, ii, ijk, ijkl;
       char fn_out[256];
       FILE *fp_out, *fp_in;
       char tmp[1024], *tmp2;
       int n_resim, resim[1024];
```

```
if( do_readin ){
    fp_in = fopen( fn_in, "r");
    fgets( tmp, 1024, fp_in );
    char_str_to_num_ary( tmp, &n_resim, resim );
    sim = n_resim;
    for( i=0  ; i<n_resim ; i++)
        printf("resim[%d]=%d\n", i, resim[i]);
    fclose( fp_in );
}
else{
    sprintf(fn_out, "data%d.txt", n);
    fp_out = fopen(fn_out, "w");//writing data file open
}


srand(time(0));
//srand(0); //with same seed
for(ii=0; ii<sim; ii++){ //Start of Replicates
    fprintf(stderr, "computing %d th out of %d\n", ii+1, sim);
    if(do_readin){
        fp_in = fopen( fn_in, "r");
        fgets( tmp, 1024, fp_in );
        fgets( tmp, 1024, fp_in );
        for( j=0 ; j<(n+2)*(resim[ii]-1) ; j++ )
            fgets( tmp, 1024, fp_in );
            fgets( tmp, 1024, fp_in );
            for( j=0 ; j<n ; j++ ){
                fgets( tmp, 1024, fp_in );
                strtok(tmp, " \t\n");
                x[j]=atof( tmp );
                tmp2 = tmp + strlen(tmp)+1;
                c[j]=atof( tmp2 );
            }
    }
    else{
        for(j=0;j<n;j++){
```

```
    u1[j] = 0;
    // u2[j] = 0;
    u3[j] = 0;
    // u4[j] = 0;
    u1[j] =random1();
    //u2[j] =random1();
    u3[j] =random1() * m3;
    //u4[j] =random1();
    //printf("u %8.6f\n",u[j] );
}
for(j=0;j<n;j++){
    x1[j] = 0;
    //printf("log %10.8f\n", log(1-u[j]));
    //printf("m1 %8.6f\n", -m1);
    x1[j] = -m1*log(1-u1[j]);
    //printf("x1 %22.20f\n",x1[j] );
}
/*for(j=0;j<n;j++){
    x2[j] = 0;
    x2[j] = -m2*log(1-u2[j]);
    //printf("x2 %8.6f\n",x2[j] );
}
   for(j=0;j<n;j++){
       x3[j] = 0;
       if (u4[j] <= pp1){
            x3[j]=x1[j];
       }else {
            x3[j]=x2[j];
       }
       // printf("x3 %8.6f\n",x3[j] );
} */
   for(j=0;j<n;j++){
       x4[j] = 0;
   //x4[j] =-m3*log(1-u3[j]); //exponential censoring
   x4[j] = u3[j]; //uniform censoring
```

```
            //printf("x4  %8.6f\n",x4[j]  );
        }

        for(j=0;j<n;j++){  //data  generate
            x[j]  =  0;
            //  printf("x3  =   %8.6f   ,   x4  =  %8.6f  \n",  x3[j],x4[j]);
            if  (x1[j]  <  x4[j])  {
                x[j]=x1[j];
                c[j]=1;
                //printf("================0  case\n");
            }
            else  {
                x[j]=x4[j];
                c[j]=0;
            }
            //printf("x  =  %8.6f,  c=  %8.6f\n",x[j],c[j]);
            //printf("before  x[j][0]  ==  %8.6f\n",x[j][0]  );
        }  //End  of  data  generate

        //if(  !do_readin){
        fprintf(fp_out,  "xx%d,  cc%d  \n",  ii+1,  ii+1);
        for(j  =0;  j<n;  j++){
            fprintf(fp_out,  "%10.8f  \t  %10.8f  \n"  ,x[j],  c[j]);
        }
        fprintf(fp_out,"\n");//End  of  data  saving
    }//End  of  else  for  do_readin  in  the  rep


    for(j=1;j<n;j++){  //sorting  the  data
        ox[j]  =  0;
        ox[j]  =  x[j];
        for  (i=0;i<j;i++){
            if  (ox[i]  >  x[j])  {
                for  (k=j;k>i;k--){
                    ox[k]  =  ox[k-1];
                }  //move  for
```

```
                ox[i] = x[j];
            break;
            }//if - sliding
        }//finding the right place in ox
    } // End of sorting the data

    //for(j=0;j<n;j++) {
    //printf("%12.10f\n ",ox[j]);
    //}

    /*for(j=0;j<n;j++) {
    printf("x[%d][0] x[%d][1]= %12.10f || %12.10f\n ", j,j,x[j][0],x[j][1]);
    }
    */
    for(j=0;j<n;j++){
        z[j] = 0;
        //printf("before x[j] == %8.6f\n",x[j] );
        z[j] = pow(x[j],lam_inv);
        // printf("z[j] = %20.18f\n",z[j]);
    } // data inversion



    mmaxloglik = -pow(10,10);
    for ( jj=0;jj<ar;jj++){
        del = 0.01;
        lam = 10.0;
        lam_new = 1; // Initial setting for lambda to 1
        mloglik0 = 0.0;
        mloglik1 = 0.0;
        maxloglik = -pow(10,10);
        r1 = 0.0;
        r2 = 0.0;
        ijk = 0;
//This section calls the subroutine that estimates the value of lambda for the next iteration.
        while((ijk < 1000) & ( myAbs(lam_new - lam) > 0.05) ){
            lam = lam_new;
```

```
        for(j=0;j<n;j++){
            w[j] = 0;
            //printf("before x[j] == %8.6f\n",x[j] );
            w[j] = pow(z[j],lam);
            // printf("z[j] = %20.18f\n",z[j]);
        }
        //printf("===============lam= %8.6f\n",lam);
        mloglik0 = maxlam(lam, z, w);
        mloglik1 = maxlam(lam+del, z, w);
        r1 = g(lam, z);
        r2 = g(lam + del, z);
        temp = r1 / ( r2 - r1) ;
        lam_new = lam - temp * del;
        //printf("===============lam_new= %8.6f\n",lam_new);
        if (isnan(lam_new)==1){lam_new = lam;}
        if( mloglik1 >  maxloglik ){
            maxloglik = mloglik1;
            maxlambda = lam + del;
            maxp1 = p[0];
            maxp2 = p[1];
            maxm1 = m[0];
            maxm2 = m[1];
        }
        //printf("r1= %8.6f\t r2= %8.6f\t\n", r1, r2);
        //printf("lam_new= %8.6f\t lam= %8.6f\t mloglik0= %8.6f\t
mloglik1= %8.6f\t\n",lam_new, lam, mloglik0,mloglik1);
        //printf("ijk= %d\t lam= %8.6f\t p[0]= %8.6f\t p[1]= %8.6f\t m[0]= %8.6f\t
    m[1]= %8.6f\t mloglik0= %8.6f mloglik1= %8.6f\t\n", ijk, lam, p[0],p[1],m[0],m[1],mloglik0,
    mloglik1);
//This section of code calls the subroutine that used the EM algorithms to generate the next
    iteration of
//estimates of the mixing proportion and the means of the components with the revised estimate of
    the
//transformation parameter.

        ijk = ijk+1;
```

```
            } //End of While for g

            if( maxloglik >  mmaxloglik){
                mmaxloglik = maxloglik;
                mmaxlambda = maxlambda;
                mmaxp1 = maxp1;
                mmaxp2 = maxp2;
                mmaxm1 = maxm1;
                mmaxm2 = maxm2;
            }
            //printf ("rsp = %d\t maxloglik = %f maxlambda = %f maxp1 = %f maxp2 = %f
        maxm1 = %f maxm2 = %f\n", jj+1, maxloglik, maxlambda, maxp1, maxp2, maxm1,
        maxm2);
         }//End of jj for alternative
         //printf ("mmaxloglik = %f mmaxlambda = %f mmaxp1 = %f mmaxp2 = %f mmaxm1
        = %f mmaxm2 = %f\n", mmaxloglik, mmaxlambda, mmaxp1, mmaxp2, mmaxm1,
        mmaxm2);

         nnaxloglik = -pow(10,10);

         for ( jj=0;jj<nr;jj++){
            ndel = 0.01;
            nlam = 10.0;
            nlam_new = 1;
            nloglik0 = 0.0;
            nloglik1 = 0.0;
            naxloglik = -pow(10,10);
            nr1 = 0.0;
            nr2 = 0.0;

            ijkl = 0;
//The iterative procedure ends when the change in the estimated transformation parameter is
//less than or equal to 0.05. In that event the estimates from the current lambda are used.
            while((ijkl < 1000) & (myAbs(nlam_new - nlam) > 0.05) ){
                nlam = nlam_new;
```

```
        for(j=0;j<n;j++){
            w[j] = 0;
            //printf("before x[j] == %8.6f\n",x[j] );
            w[j] = pow(z[j],nlam);
            // printf("z[j] = %20.18f\n",z[j]);
        }
        //printf("===============lam= %8.6f\n",lam);
        nloglik0 = naxlam(nlam, z, w, &mu);
        nloglik1 = naxlam(nlam+ndel, z, w, &mu);
        nr1 = ng(nlam, z);
        nr2 = ng(nlam + ndel, z);
        ntemp = nr1 / ( nr2 - nr1) ;
        nlam_new = nlam - ntemp * ndel;
        //printf("===============lam_new= %8.6f\n",lam_new);
        if (isnan(nlam_new)==1){nlam_new = nlam;}
        if( nloglik1 >  naxloglik ){
            naxloglik = nloglik1;
            naxlambda = nlam + del;
            mu;
        }

        //printf("r1= %8.6f\t r2= %8.6f\t\n", r1, r2);
        //printf("lam_new= %8.6f\t lam= %8.6f\t mloglik0= %8.6f\t mloglik1= %8.6f\t\n",lam_new, lam, mloglik0,mloglik1);
        //printf("ijkl= %d\t nlam= %8.6f\t nloglik0= %8.6f\t nloglik1= %8.6f\t \n", ijkl, nlam, nloglik0, nloglik1);
        ijkl = ijkl+1;
    } //End of While for ng
    if( naxloglik >  nnaxloglik){
        nnaxloglik = naxloglik;
        nnaxlambda = naxlambda;
        mmu = mu;
    }
    //printf ("rsp = %d\t naxlambda = %f mu = %f naxloglik = %f \n", jj+1, naxlambda, mu, naxloglik);
 }//End of jj for null
```

```
        //printf ("nnaxlambda = %f mmu = %f nnaxloglik = %f \n", nnaxlambda, mmu,
       nnaxloglik);
        lrts = 2*(mmaxloglik - nnaxloglik);
        //printf("=====================sim = %d\t", ii+1);
        printf("sim = %d\t mmaxlambda = %8.6f mmaxp1= %8.6f mmaxp2= %8.6f
       mmaxm1= %8.6f mmaxm2= %8.6f mmaxloglik=%8.6f nnaxlambda = %8.6f mmu = %8.6f
       nnaxloglik=%8.6f lrts=%8.6f \n", ii+1, mmaxlambda, mmaxp1, mmaxp2, mmaxm1,
       mmaxm2, mmaxloglik, nnaxlambda, mmu, nnaxloglik, lrts);
    } //End of rep
} //End of Main Function




//Function of reading the data to reuse
void      char_str_to_num_ary( char *string, int *p_n_int, int *int_ary )
{
        int       n_string;
        char      *str_bck, *tmp, *delimit=" ,\n\t";
        int       i_str;

        n_string = strlen(string);
        str_bck = (char*)calloc( n_string+2, sizeof(char) );
        strcpy( str_bck, string );
        tmp = str_bck;

        i_str=0;
        while( strlen( tmp ) )
        {
                strtok( tmp, delimit );
                int_ary[i_str++] = atoi( tmp );
                tmp = tmp + strlen(tmp) + 1;
        }
        *p_n_int=i_str;
        free( str_bck );
}
```