A Power Study Assuming a Single Exponential Distribution with Long Term Survivors and a Mixture of Two Exponential Distributions

A Dissertation Presented

by

Samuel Cook

to

The Graduate School

in Partial fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

(Statistics)

Stony Brook University

August 2009

Stony Brook University
The Graduate School

Samuel Cook

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

Dissertation Advisor Professor Stephen J. Finch
Department of Applied Mathematics and Statistics

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

Professor Hongshik Ahn
Department of Applied Mathematics and Statistics

Professor Derek Gordon
Rutgers University, Department of Genetics

This dissertation is accepted by the Graduate School.

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

A Power Study Assuming a Single Exponential Distribution with Long Term Survivors and a Mixture of Two Exponential Distributions

by

Samuel Cook

Doctor of Philosophy

in

Applied Mathematics and Statistics

(Statistics)

Stony Brook University

2009

Mixture models for fitting long-term survivors (LTS) have an extensive research history and mixture models assuming a mixture of two component exponentials with finite means has, more recently, been researched. These tests of mixture mechanisms in survival data have a fundamental importance in bio-statistical research. While the tests are well documented, an analysis of power of these tests has seen little attention. A simulation program to replicate these situations is developed, as is software for the computation of these tests. The null distribution of a single exponential with LTS is shown to be $\pi_0\chi_0^2 + (1-\pi_0)\chi_1^2$, where π_0 represents the proportion of zero likelihood ratio test statistics and is confirmed to converge to $\frac{1}{2}\chi_0^2 + \frac{1}{2}\chi_1^2$ as n goes to infinity for a finite study. The null distribution for a mixture of two components appears to be $\pi_1\chi_0^2 + (1-\pi_1)\chi_{\nu_1}^2$. When studying the power of a two component exponential mixture, mixing proportion of the components, and the difference in component means were the primary variables considered in the alternative hypothesis. A 50-50 mixture with greatest difference of component means (difference equals 1.5) has power near 1 for both censoring patterns and censoring rates, even for sample size. For skewed mixing proportions (that is, m =0.85) with greatest difference of component means, the power increases with increasing sample size, as expected. For smaller difference of component means (difference equals 0.5) for both symmetric (m =0.50) and skewed mixtures, the power is low for both censoring patterns and both censoring rates. In the mixture model fitting LTS the length of the study and the proportion of LTS were the primary variables considered. The longer the study and the larger the proportion of LTS provided the highest power, with power near 1 for large samples and study lengths of at least 5 times the distribution mean. A model to estimate the power for both models is developed to help estimate the model's effectiveness based on the properties of ones sample.

Table of Contents

Chapter 1. Introduction……………………………………………………………1

Chapter 2. Methods

Section 2.1. Definitions ……………………….………………….……..…6

2.11 Long Term Survivors ….……………………………….…...6

2.12 Censoring ………..………………………...…………..6

2.13 Censoring Distributions ………………...…..…………..7

2.14. Likelihood Function of Single Exponential (L_0) …..……...7

2.15 Likelihood Function of Single Exponential With LTS(L_S)...8

2.151 Finite Censoring without LTS…………………..…8

2.152 Finite Censoring with LTS…….……………………10

2.153 Finite Censoring with Study Duration……………12

2.16 Likelihood Function of Mixture of Two Exponentials(L_M).12

2.17 L_S vs L_0 13

2.171 Settings 13

2.172 Number of Random Starting Points 14

2.173 Power Study ………………………………………14

2.18 L_M vs L_0 15

2.181 Settings 16

2.182 Number of Random Starting Points 16

2.183 Checking Simulation Output vs Past Results …….19

2.184 Power Study ………………………………………19

Section 2.2 Nelder-Mead Algorithm ……………………………………21

Chapter 3. Results

Section 3.1. Programming ……………………………………………....23

3.11 Time to Event Data ……………………………………….. 23

3.12 MLE Calculations ………………………………………… 24

Section 3.2 Single Exponential with LTS ………………………………24

3.21 Null Distribution of LRTS for L_S vs L_0 ……………………24

3.22 Global and Local Maximization checks …………………. .25

3.23 Estimated Null Perctiles …………………………………...28

3.24 Power ……………………………………………………….37

3.25 Model Limits ……………………………………………….44

Section 3.3 Mixture of Two Exponentials ……………………………..56

3.31 Null Distribution of LRTS for L_M vs L_0 ………………..…56

3.32 Power ……………………………………………………….60

Chapter 5. Conclusions…………………………………………………………..69

References…………………………………………..…………………73

Appendix ……………………………………………………………...…76

Acknowledgments

The thesis presented is dedicated to the memory of my father John E. Cook, who was not able to see me complete it. My path towards a doctoral degree was greatly influenced by him. I love him and miss him greatly.

The thesis presented could not have been completed without the love and support from my mother Susan J. Littrell, and my sister Rachel L. Cook. I would also like to acknowledge my extended family and particularly thank Jim Cook, who has in many ways become a father figure in my life.

Of course, this thesis would not have been possible with out the support of the love of my life Lauren Adamek and the many friends who have pushed me towards my goal. I would like to particularly thank Brian Fix who was instrumental in guiding me through the programming of much of the code used in this thesis, without him I would have not finished in the time that I did.

I would like to thank my committee, who was supportive of me and very helpful. I would particularly like to thank Dr. Nancy Mendell, who went above and beyond her duties in assisting my advisor Dr. Stephen Finch on the advising of this thesis.

I would like to acknowledge two former teachers in my life. I thank the late Dr. Miguel Paredes, who was instrumental in my acceptance to Stony Brook University. I would also like to acknowledge Dr. Harrison “Chuck” Straley who was a great inspiration to me as a man who loves to teach. I can only hope to be half the teacher in my life that he has been in his.

Finally, I would have never been able to do this without the unconditional support of my advisor Dr. Stephen Finch. His patience was instrumental in my completion of this thesis. His guidance is greatly appreciated and will never be forgotten. Thank you Dr. Finch.

Chapter 1. Introduction

In survival analysis, the variable of primary interest is the time between a specified originating event and the occurrence of the event of interest. For example, the time from treatment to death or the time from treatment to remission are variables of importance.

Mixture models for fitting long-term survivors (LTS) have an long research history (Farewell 1982), and mixture models assuming a mixture of two component exponentials with finite means has, more recently, been researched (Ye 2006). These tests of mixture mechanisms in survival data have a fundamental importance in bio-statistical research. While the tests are well documented, an analysis of power of these tests has received little attention.

There is a wide range of applications where mixture distributions are important. This is due to their flexibility in mirroring complex situations. Various mixture survival models have been proposed over the last century (Böhning and Seidel, 2003).

A common mixture survival model is the cure rate model. This is a survival distribution that is the mixture of two components, in which one component follows a population who have expected responses and the other component has a survival distribution of patients given a treatment for a disease. It has been presented in medical and biomedical applications, clinical applications, and epidemiological applications. The modeling of a cancer patient who had received a specific treatment was developed by Boag (1949). He estimated the fraction of patients “cured” by a cancer therapy using a lognormal model with maximum likelihood. Berkson and Gage (1952) considered a model in which the survival time of a patient with cancer was modeled as the mixture of the survival distribution of the general population and a survival distribution determined by the cancer. The survival distribution determined by the cancer was modeled as the survival distribution of the general population reduced by a negative exponential decay factor linear in time. Haybittle (1965) developed an “extrapolated actuarial” two-parameter model that also incorporated survival functions for treated patients that are functional modifications of the “normal population”.

This dissertation follows the LTS model given by Farewell (1982). Farewell defines LTS as a noticeable proportion of subjects who, by the end of the study, do not see their event of interest. Farewell (1982) assumed a fraction of LTS in a survival study. In a later work, Farewell (1986) examined the use of mixture models for LTS. Maller and Zhou (1992) developed an independent non-parametric censoring model for estimating the proportion of LTS in a censored sample. In a subsequent paper Zhou and Maller (1995) discussed the test for identifying the presence of LTS in the population and goodness of fit tests for the parametric description of the data. They developed its asymptotic theory and used the likelihood ratio test statistic (LRTS) to test whether a non-zero proportion of

LTS is indeed present in the population. Their procedure used the exponential distribution for non-LTS subjects. Copas and Heydari (1997) developed an exponential mixture model for recidivism of criminals that explicitly allows for delay. They estimated the risk of return to the judicial system, where the survival time including both the time from release to the first re-offence and the time from this re-offence to conviction. Peng et al. (1998) proposed a mixture model by using the generalized F distribution family. Tsodikov (2001) provided a parametric cure model and the corresponding algorithm to estimate the cure rate. He estimated the distribution function $F(t)$ non-parametrically as if the cure rates were known. Tsodikov (2002) developed a series of semi-parametric survival models and algorithms to deal with the combining of long-term and short-term covariate effects in cancer survival analysis. Tsodikov et al. (2003) considered the utility of the bounded cumulative hazard model as an alternative to the two-component mixture model in the cure rate estimation. Corbiere et al. (2009), suggest a penalized likelihood approach, which allows for flexible modeling of the hazard function for susceptible individuals when studying individuals, who may experience the event of interest, and non-susceptible individuals that will never experience it.

The power of these models has seen little attention. Furukawa et al. (2009), studied the power of risk assessment applying to age-time trends and susceptible subgroups. Broet et al. (2003) studied the power of the inclusion of long term survivors in a two sample tests in randomized designs.

Broet et al. (2001) proposed statistics for testing a two-sample comparison of survival times with long-term survivors. Lam et al. (2005) provided a model for the recurrence of breast cancer in long term survivors with a focus on censored data.

I will study the LTS model (L_S) considered by Zhou and Maller (1995) and the mixture model (L_M) considered by Ye (2006) in tests against a single exponential test distribution (L_0).

This dissertation addresses the following questions:

1. How does a finite study duration affect the null distribution of the test for L_S against L_0?
2. Is it possible to estimate the power for the test H0: L_0 vs H1:L_S or the test H0: L_0 vs H1:L_M?
3. What are the range of parameters (e.g. sample size, fraction of LTS and censoring rate) in which the power of the LRTS exceeds 50%?

Chapter 2 of this dissertation presents the methods including the numerical algorithms for each model. The Nelder-Mead (NM) algorithm (Nelder and Mead 1965) is used to estimate the maximimum likelihood estimators with different settings of starting values. For the L_M model, different numbers of starting values are considered and compared to maximize the probability of finding the maximum. Numerical algorithms are programmed in C++ for Linux. They also can be run in any Windows operating system with use of a Linux emulator. This software is available on request from me and will soon be available for use on my website.

Chapter 3 of this dissertation gives the simulation results for the MLE's, the null distribution of the LRTS, and the approximate alternative distribution of the LRTS. It gives the observed simulated power and the fitted estimated power models. For the L_S estimated power model, the limitations are studied in depth.

Chapter 4 of this dissertation contains the conclusion and a discussion of issues raised in the simulation study as well as the future direction of this work.

Chapter 2: Methods

2.1 - Definitions:

The probability density function (PDF) of the exponential distribution (exp(λ)) is $f(t) = \lambda e^{-\lambda t}, t > 0$ with a mean of $\frac{1}{\lambda}$. The uniform distribution (U(a,b)) has a PDF of $f(t) = \frac{1}{b-a}, b > a, a < t < b$. Its expected value is $\frac{a+b}{2}$.

2.11 - Long Term Survivors:

The survival function of a random variable whose cumulative distribution function is *F(x)* is defined to be $S(x) = P(X > x) = 1 - F(x)$. The hazard function is the instantaneous death rate and is defined to be $h(x) = \lim_{\Delta x \to 0} \frac{P(x \leq X \leq x + \Delta x \mid X \geq x)}{\Delta x} = \frac{f(x)}{s(x)}$. (Klein & Moeschberger 2003).

2.12 - Censoring

I denote the true observed time of the i^{th} observation as t_i^* and its censoring time as u_i. The observed time $t_i = \min(t_i^*, u_i), (1 \leq i \leq n)$. The ordered survival times are denoted $t_{(i)}$ ($t_{(1)} \leq t_{(2)} \leq ... \leq t_{(n)}$). For each observation a censoring indicator is recorded such that $c_i = 1$ indicates an absence of censoring (i.e., $t_i = t_i^*$) and $c_i = 0$ otherwise (i.e., $t_i = u_i$). If there are multiple

$t_{(i)}$ with equal responses, then they will be ordered by first listing those that have not been censored.

2.13 - Censoring Distributions

When determining censoring times, I will follow Peng et al. (2001) and use a uniform censoring pattern, as well as an exponential censoring pattern. The exponential censoring pattern has a mean of $\frac{1}{\beta}$, and the uniform is $U[0, b]$. The parameters β and b are calculated so that the expected proportion of censoring will be a specified value such as 10%, 30% or 45%. I assume the censoring distribution is independent of the survival distribution.

2.14 - Likelihood Function: Single exponential (L_0):

A single component exponential distribution with mean $\frac{1}{\lambda}$, $f(t) = \lambda e^{-\lambda t}$, has a survival function of $S(t) = e^{-\lambda t}$. With censoring, the likelihood function for a random sample of n censored observations is

$L_0(t_1, t_2, ..., t_n, \lambda) = \prod_{i=1}^{n} [(\lambda e^{-\lambda t_i})^{c_i} (e^{-\lambda t_i})^{1-c_i}]$. The log-likelihood function for the model is $l = \log(L_0) = \sum_{i=1}^{n} \{c_i[\log(\lambda) - \lambda t_i]\} + (1 - c_i)[\log(e^{-\lambda t_i}))]\}$. The maximum likelihood estimate of λ is $\hat{\lambda} = \frac{\sum_{i=1}^{n} c_i}{\sum_{i=1}^{n} t_i}$. (Maller and Zhou 1996).

2.15 - Single exponential with survivors (L_S):

Long Term Survivors (LTS) include in practice those subjects who, by the end of the study, did not observe the event of interest. Mathematically LTS is defined as $LTS = 1 - \lim_{t \to \infty} F(t)$ (Farewell 1982).

The survival function of a single exponential with fraction LTS is $S(t) = 1 - (LTS) + (LTS)e^{-\lambda t}$ The likelihood function is

$$L_S(t_1, t_2, ..., t_n, \lambda, p_L) = \prod_{i=1}^{n} ((LTS)\lambda e^{-\lambda t_i})^{c_i} (1 - (LTS) + (LTS)e^{-\lambda t_i})^{1-c_i} .$$

Its log-likelihood function is

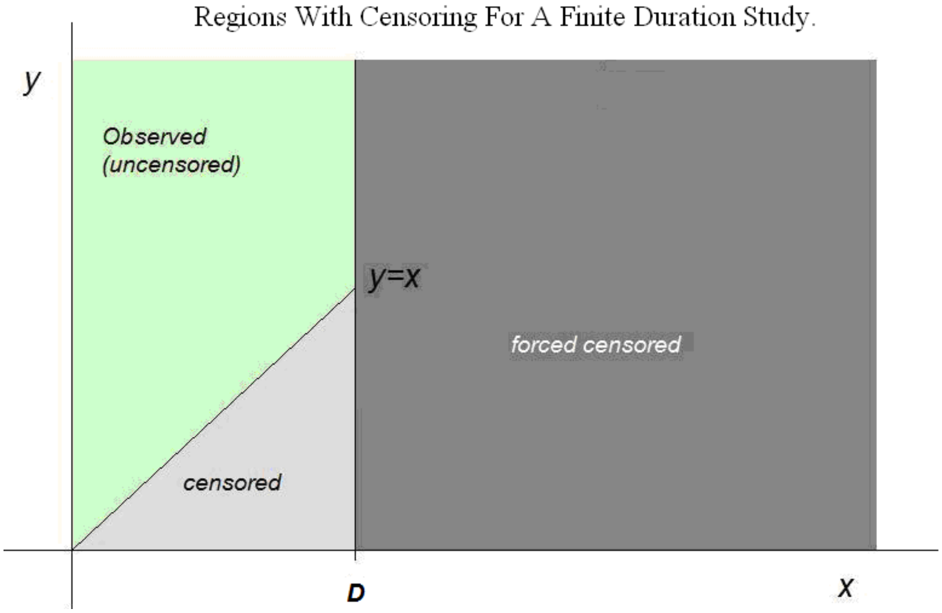
$$\log(L_S) = \sum_{i=1}^{n} \{c_i[\log(LTS) + \log(\lambda) - \lambda t_i]\} + (1 - c_i)[\log(1 - (LTS) + (LTS)e^{-\lambda t_i}))]\} .$$

(Zhou and Maller 1995). I will refer to this model as L_S.

2.151 - Finite Censoring without LTS:

I consider censoring with a finite study duration of time D; that is, at time D there will be active participants who will not be followed further. I will consider any subject still active at the end of the study to be censored at D. For example, suppose $X \sim$ EXP(1), and the data is censored according to an independent exponential censoring pattern $Y \sim$ EXP(β). The proportion of uncensored observations is then $P((X < Y) \cap (X < c))$. This will dictate a larger censoring pattern mean parameter than would hold for an indefinitely long study, so that we can still ensure the expected proportion of censored responses we specify. This is illustrated in Figure 2.1.

Figure 2.1:



Then the proportion of uncensored observations for a study of duration *D* is given by:

$$P\big((X<Y)\cap(X<c)\big)=\int_0^c\int_x^\infty e^{-x}\frac{1}{\beta}e^{-y/\beta}dydx=\int_0^c e^{-(x+x/\beta)}dx$$

$$=\left(\frac{-\beta}{\beta+1}e^{-(x+x/\beta)}\Big|_0^c\right)=\frac{\beta}{\beta+1}-\frac{\beta}{\beta+1}e^{-(c+c/\beta)}$$

It may not be possible to have an expected censoring rate *r* with study duration *D*. Table 2.1 gives *P(X > D)* for specified *D*.

Table 2.1 $P(X > D)$ For Specified *D*.

Study Duration	$D = \infty$	$D = 5$	$D = 4$	$D = 3$	$D = 2.5$	$D = 2$	$D = 1.5$
Proportion Yet to be observed at *D*	0%	0.67%	1.83%	4.98%	8.21%	13.53%	22.3%

When considering a finite study duration, *D* observations will be censored for two reasons: still active at *D* and censored by *Y* before *D*. For example, in a study of duration 4, where 10% of all observations are censored, the 10% that are censored will be divided into an expected 1.83% with X > 4, and an expected 8.17% that will be censored by *Y* before 4. This reduction in the proportion of observations, that take the censoring distribution time from 10% to 8.17%, will dictate a censoring pattern mean that is different from the mean that satisfies $P(X > Y) = 0.10$, which describes a study of infinite duration.

2.152 - Finite Censoring with LTS:

In considering a finite study duration, the inclusion of LTS brings a third censoring issue. The censored data is still a combination of events censored at end of study *D* and events censored by *Y* before $X < D$. However, the proportion of events censored at *D* will increase due to the proportion of LTS. That is, a LTS will always be active at the end of study *D* and is hence censored. The proportion of non LTS is $P(X_i > D \mid E(X_i) = \mu)$, where μ is the mean of non-LTS survival times. It must be set to *p - LTS*. The expected proportion censored *r*, is:

$r = (LTS) + [1-(LTS)] \cdot [1 - P((X<Y) \cap (X<c))]$. Table 2.2 reports the expected proportion censored at *D* plus the expected *LTS*.

Table 2.2:

Proportion of Observations Censored at D For Selected Fractions *LTS* with Expected Survival Mean = 1.

Study Duration (*D*)	*LTS*												
	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.15	0.2
1.00	0.3679	0.374	0.381	0.387	0.393	0.399	0.406	0.412	0.418	0.425	0.431	0.463	0.494
1.25	0.2865	0.294	0.301	0.308	0.315	0.322	0.329	0.336	0.344	0.351	0.358	0.394	0.429
1.50	0.2231	0.231	0.239	0.246	0.254	0.262	0.270	0.278	0.285	0.293	0.301	0.340	0.379
1.75	0.1738	0.182	0.190	0.199	0.207	0.215	0.223	0.232	0.240	0.248	0.256	0.298	0.339
2.00	0.1353	0.144	0.153	0.161	0.170	0.179	0.187	0.196	0.205	0.213	0.222	0.265	0.308
2.25	0.1054	0.114	0.123	0.132	0.141	0.150	0.159	0.168	0.177	0.186	0.195	0.240	0.284
2.50	0.0821	0.091	0.100	0.110	0.119	0.128	0.137	0.146	0.156	0.165	0.174	0.220	0.266
2.75	0.0639	0.073	0.083	0.092	0.101	0.111	0.120	0.129	0.139	0.148	0.158	0.204	0.251
3.00	0.0498	0.059	0.069	0.078	0.088	0.097	0.107	0.116	0.126	0.135	0.145	0.192	0.240
3.25	0.0388	0.048	0.058	0.068	0.077	0.087	0.096	0.106	0.116	0.125	0.135	0.183	0.231
3.50	0.0302	0.040	0.050	0.059	0.069	0.079	0.088	0.098	0.108	0.117	0.127	0.176	0.224
3.75	0.0235	0.033	0.043	0.053	0.063	0.072	0.082	0.092	0.102	0.111	0.121	0.170	0.219
4.00	0.0183	0.028	0.038	0.048	0.058	0.067	0.077	0.087	0.097	0.107	0.116	0.166	0.215
4.25	0.0143	0.024	0.034	0.044	0.054	0.064	0.073	0.083	0.093	0.103	0.113	0.162	0.211
4.50	0.0111	0.021	0.031	0.041	0.051	0.061	0.070	0.080	0.090	0.100	0.110	0.159	0.209
4.75	0.0087	0.019	0.028	0.038	0.048	0.058	0.068	0.078	0.088	0.098	0.108	0.157	0.207
5.00	0.0067	0.017	0.027	0.037	0.046	0.056	0.066	0.076	0.086	0.096	0.106	0.156	0.205
5.25	0.0052	0.015	0.025	0.035	0.045	0.055	0.065	0.075	0.085	0.095	0.105	0.154	0.204
5.50	0.0041	0.014	0.024	0.034	0.044	0.054	0.064	0.074	0.084	0.094	0.104	0.153	0.203
5.75	0.0032	0.013	0.023	0.033	0.043	0.053	0.063	0.073	0.083	0.093	0.103	0.153	0.203
6.00	0.0025	0.012	0.022	0.032	0.042	0.052	0.062	0.072	0.082	0.092	0.102	0.152	0.202
6.25	0.0019	0.012	0.022	0.032	0.042	0.052	0.062	0.072	0.082	0.092	0.102	0.152	0.202
6.50	0.0015	0.011	0.021	0.031	0.041	0.051	0.061	0.071	0.081	0.091	0.101	0.151	0.201
6.75	0.0012	0.011	0.021	0.031	0.041	0.051	0.061	0.071	0.081	0.091	0.101	0.151	0.201
7.00	0.0009	0.011	0.021	0.031	0.041	0.051	0.061	0.071	0.081	0.091	0.101	0.151	0.201
7.25	0.0007	0.011	0.021	0.031	0.041	0.051	0.061	0.071	0.081	0.091	0.101	0.151	0.201
7.50	0.0006	0.011	0.021	0.031	0.041	0.051	0.061	0.071	0.081	0.091	0.100	0.150	0.200
7.75	0.0004	0.010	0.020	0.030	0.040	0.050	0.060	0.070	0.080	0.090	0.100	0.150	0.200
8.00	0.0003	0.010	0.020	0.030	0.040	0.050	0.060	0.070	0.080	0.090	0.100	0.150	0.200
∞	0.0000	0.010	0.020	0.030	0.040	0.050	0.060	0.070	0.080	0.090	0.100	0.150	0.200

2.153 - Finite Censoring with Study Duration:

I considered exponential censoring and uniform censoring. Any simulation setting that use exponential censoring is possible because the exponential distribution is unbounded. Uniform censoring can change the desired study duration because the censoring distribution is bounded. That is, if one were to use uniform censoring to simulate a 30% censoring rate, the censoring pattern would be $x \sim U(0,3.2)$. Therefore if one wanted to simulate a study of duration 5 times the expected event time with 30% censoring, the uniform censoring pattern has the undesirable property that censoring will not occur between 3.2 and 5.0.

In settings that use a uniform distribution, only those with an upper bound greater than the duration of the study are used.

2.16 – Mixture of Two Exponentials (L_M):

Under the alternative hypothesis, the survival function is

$S(t) = me^{-\lambda_1 t} + (1-m)e^{-\lambda_2 t}$, $(t \geq 0, 0 \leq m \leq 1, 0 \leq \lambda_2 \leq \lambda_1)$

where m is the proportion from the exponential component with smaller mean, $\frac{1}{\lambda_1}$. The log-likelihood function $l(\lambda_1, \lambda_2, m)$ is

$$l(\lambda_1,\lambda_2,m) = \log(L_n) = \sum_{i=1}^{n}\{c_i[\log(m\lambda_1 e^{-\lambda_1 t_i} + (1-m)\lambda_2 e^{-\lambda_2 t_i}] + (1-c_i)[\log(me^{-\lambda_1 t_i} + (1-m)e^{-\lambda_2 t_i})]\}.$$

(Ye 2006). I will refer to this model as L_M.

2.17 – L_S vs L_0:

I developed an algorithm to calculate the MLE of L_S in the programming language C++ with the NM alorithim. The programming methodology is discussed in section 2.3. The programming code for L_S can be found in Appendix A1.

2.171 – Settings:

The null distribution is a single exponential with a finite study duration and censoring. The sample sizes in the simulation are 50, 100, 200, 500, 1000, 2000 and 10,000. I used an exponential(β) and uniform(0, b) censoring pattern with distribution means set such that simulations had 10%, 30% and 50% expected censoring rate. I also considered a finite study duration of 3 times the mean of the event distribution (short study) and a study duration of 5 times the event mean (long study). These simulations are summarized by mean LRTS, variance of the LRTS, the fraction of zero LRTS values observed (where nearly zero is defined as an LRTS less than 0.0001), the mean of the non-zero LRTS values and LRTS values at selected percentiles. There were 42 different simulations under exponential censoring as shown in Table 3.3. The results are presented in section 3.23. There were 21 uniform simulations as shown in Table 3.4. The results are presented in section 3.23.

2.172 – Choosing number of Random Starting Points:

In section 2.182 the number of random starting points for the mixture model I studied is discussed in detail. For the L_s model 9 random starting points arranged in a 3 by 3 grid was sufficient to find the LRTS. In a pilot study of 200 replicates, all 200 required 9 RSP or fewer to obtain the LRTS value observed with 100 RSP arranged in a 10 by 10 grid.

2.173 – Power Study:

I estimated the LRTS and the fraction of zero LRTS values with a linear regression calculated with data collected from simulations. In results section 3.23 I show that it is plausible the null distribution follows $\pi_0\chi_0^2 + (1-\pi_0)\chi_1^2$, where π_0 is the estimated fraction of zero LRTS values as given in equation 3.1 I used that distribution to estimate the 75th, 90th, 95th, 99th, and 99.9th percentiles for the sample sizes used. The analysis of these results are discussed in detail in section 3.23 . I used the estimated 99th percentile for each sample size as the 1% critical value when measuring power. I simulated the power of L_M using 1000 replications for the following settings:

1. Two censoring patterns *P*: exponential (1) or uniform (0) censoring pattern;
2. Six sample sizes *n* : 200, 350, 500, 750, 1000, or 2000;

3. Two study durations D: Longer Indicator (1, duration 5) or Shorter (0, duration 3)
4. Three censoring rates *R* of 15%, 30% or 45%.
5. Three proportions of Long Term Survivors LTS: 2%, 5%, or 8%.

In the regression analysis of power, I considered a probit regression with the dependent variable is $\Phi^{-1}(\hat{p})$, where $\Phi(z) = P(Z \leq z)$ with *Z* having a standard normal distribution and $\hat{p}$ is the observed power.

2.18 – L_M vs L_0:

Ye (2006) developed an algorithm that computes the maximum likelihood estimates (MLEs) of the mixing proportion and means of a survival distribution that is the mixture of two exponential components. Then the likelihood ratio test statistic (LRTS) of the null hypothesis that a survival distribution is exponential against the alternative that the survival distribution is the mixture of two exponentials is easily calculated.

I have extended this algorithm using a simulation program I wrote in C++. The base used for the coding was the software developed by Ye, in the programming language R. The programming methodology is discussed in section 2.3. The programming code I developed for L_M can be found in Appendix A2.

2.181 – Settings:

Each simulation incorporates a sample size (S), censoring pattern (P), censoring rate (R), mixing proportion (M), and a difference (D) in mixture means. I follow Peng *et al.* (2001), who used uniform $U[0, b]$ and exponential censoring distributions, where Exp(β) denotes the exponential censoring distribution with mean $\frac{1}{\beta}$. The values of b and β are calculated so that the expected fraction censored is either 10% or 30%. The means of the distributions are calculated from the properties of the mixing proportion and the desired mean differences.

I will define the vector generated by RSP random starting points on *rep* repetitions and $L^*_{m,RSP,rep}(S,P,M,D,R)$. For example, $L^*_{m,175,200}$(500, EXP, 0.50, 1.0, 0.10) is the vector of length 200 whose i^{th} entry is the maximum log likelihood found using the 500 observations, with exponential censoring, 50% mixing proportion, a difference of exponential means of 1.0 and a 10% censoring rate based on 175 random starting points.

2.182 – Choosing number of Random Starting Points:

A problem faced in the completion of a power study is the computing time needed to yield results. Ye (2006) suggested the use of 175 random starting points to maximize the likelihood. When simulating 500 replicates per setting, 175 random starting points required considerable computing time. I first ran a small pilot study to confirm that the use of 175 random starting points was sufficient. To do this I took three different sets of simulation settings at 100 replicates each and ran a simulation at 175 random starting points. I then ran the same simulation with 1000 random starting points. The pilot study confirmed that the difference between 175 random starting points and 1000 random starting points was not important, as there was no difference between the maximum LRTS value at 175 RSP compared to the maximum with 1000 RSP in 599 of the 600 samples. The one difference was an increase from 0.32 to 1.57. That is, I confirmed that a simulation with 175 RSP is highly likely to locate the maximum and will use it in my simulation study. This pilot study is summarized in Table 2.3.

Table 2.3:

Proportion of LRTS Values Consistent for 175 and 1000 random starting values

Censoring pattern	Sample size	Mixing proportion	Difference of means	Censoring Rate	Average LRTS	Fraction of zero LRTS	Replicates with larger LRTS for sample size 1000
EXP	200	0.50	1.5	0.10	50.5	0.00	0
EXP	500	0.75	1.5	0.30	15.8	0.00	0
EXP	750	0.85	1.0	0.10	7.8	0.00	0
UNIF	200	0.85	1.0	0.30	2.1	0.14	0
UNIF	500	0.50	0.5	0.30	2.9	0.04	1
UNIF	750	0.65	1.0	0.10	17.8	0.00	0

(Each setting was run with 100 replicates)

In the next part of my pilot study I used 800 samples to examine whether fewer than 175 random starting points could be used. I used 175 RSP as the threshold for an accurate LRTS calculation. That is, if the LRTS value calculated with *x* number of random starting points was the same as calculated with 175 RSP then I concluded that *x* random starting points were enough for an accurate calculation.

For each of 8 settings, I generated 100 replicates. A simulation was run at 12 different numbers of random starting points ranging from 1 to 175. In the pilot study it was found that 227 cases yielded a LRTS value of 30 or larger at 175 RSP, 413 cases yielded a LRTS value between 15 and 30, and 160 cases yielded a LRTS value less than 15. I will call an LRTS of 15 or less small, an LRTS between 15 and 30 medium and an LRTS greater than 30 large. In cases where the average LRTS value for a setting is large, the maximum value was located using the first or second starting point for 226 of 227 samples checked. Consequently, I set eight random starting points rather than two, concluding that was sufficient to locate an accurate result when the pilot mean LRTS value was over 30. For medium LRTS values, eighteen random starting points provided an accurate result for 410 of 413 samples checked. For medium cases I set 36 RSP. Even for small LRTS values, 150 random starting points was accurate for 158 of the 160 samples. I set 175 random starting points for small cases.

The LRTS value is calculated within a few seconds for a single random starting point. Therefore, LRTS size that leads to the chosen number of random starting points was obtained by running a simulation at $L^*_{m,1,100}$. An average of those 100 replicates was taken and the result gave an approximate pilot mean LRTS. This average classified the setting definition as producing small, medium or large LRTS values, and determined how many random starting points were to be used for this setting

2.183 – Checking Simulation output versus past results:

Simulated uniform and exponential censoring data were checked for accuracy by taking samples of size 100, 500, 1000, 5000, 10,000 and 50,000 and checking the mean, and variance and also applying a basic chi-square goodness of fit test to each to confirm distribution properties. The accuracy of the MLE choice made by the transcribed algorithm was checked by inputting the data simulated in Ye's R program into the converted C++ program and checking for identical results. Simulated C++ data was also used in the R program as a further check. I also re-ran the parameters Ye (2006) used in her simulations to confirm her results. These simulations were done independently of her simulations using new simulated data. In addition I incorporated additional sample sizes to model power values over a greater number of settings.

2.184 – Power Study:

Using the NM algorithm I estimated the LRTS and the fraction of zero LRTS values with a linear regression calculated with data collected from simulations. I assumed that the non-zero values follow the distribution $\Gamma(\frac{\nu_0}{2},2)$

to estimate the 75th, 90th, 95th, 99th, and 99.9th percentiles for the sample sizes used. The correlation between estimated percentile and simulated percentile is 0.985. I used the estimated 99th percentile for each sample size as our 1% critical value when measuring power. I simulated the power of L_M using 500 replications for the following settings:

1. Two censoring patterns *P*: exponential or uniform censoring pattern;
2. Four sample sizes *n* : 200, 350, 500, or 750;
3. Four mixing proportions M for the component with smaller mean: 0.50, 0.65, 0.75, or 0.85;
4. Three differences *D* between means: 0.50, 1.00, or 1.50, as shown in Table 2.4;
5. Two censoring rates *R* of 10% or 30%.

In the regression analysis of power, I considered a probit regression with the dependent variable is $\Phi^{-1}(\hat{p})$, where $\Phi(z) = P(Z \leq z)$ with *Z* having a standard normal distribution and $\hat{p}$ is the observed power.

Table 2.4.

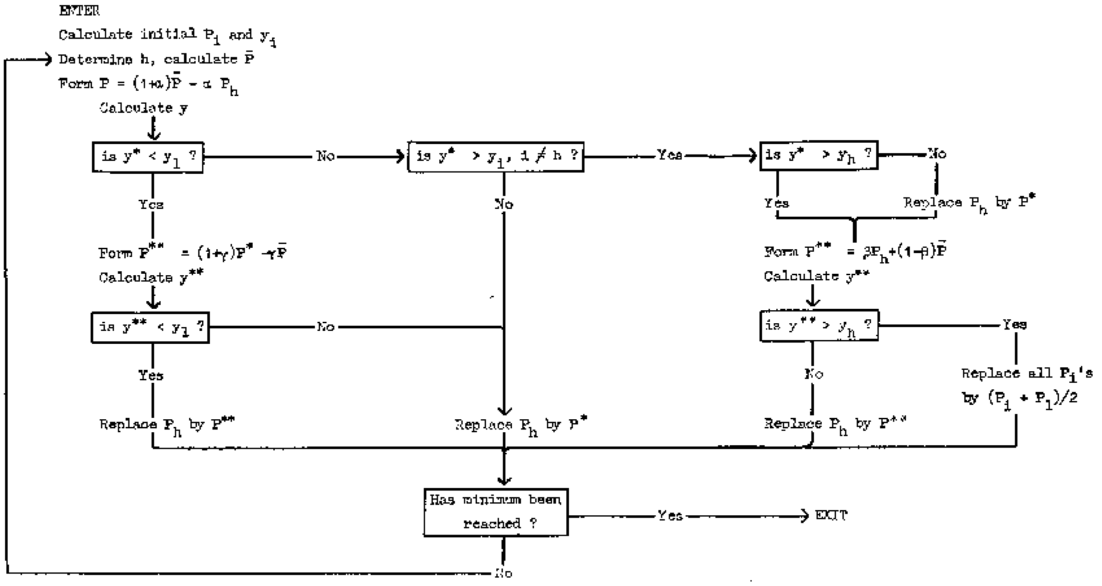
Parameters of the two-component mixture of exponentials used in power study.

The mixing proportion with smaller mean	Different Mean	
M	Smaller Mean	Larger Mean
0.50	0.75	1.25
	0.50	1.50
	0.25	1.75
0.65	0.825	1.325
	0.650	1.650
	0.475	1.975
0.75	0.875	1.375
	0.750	1.750
	0.625	2.125
0.85	0.925	1.424
	0.850	1.850
	0.775	2.275

2.2 - Nelder-Mead (NM) algorithm:

The Nelder Mead algorithm (1965) takes a function of n variables and minimizes it. It evaluates the function at the vertices of a ($n + 1$) simplex and then iteratively uses reflection, contraction and expansion of the simplex as better points are found. A vertex is replaced by points with a better value of the function until the minimal function value is obtained.

Figure 2.2 Flow diagram of Nelder-Mead algorithm.


ENTER
Calculate initial P_i and y_i
Determine h, calculate $\bar{P}$
Form $P = (1+\alpha)\bar{P} - \alpha P_h$
Calculate y
is $y^* < y_l$?
No
is $y^* > y_i, i \neq h$?
Yes
is $y^* > y_h$?
No
Yes
No
Yes
Replace P_h by P^*
Form $P^{**} = (1+\gamma)P^* - \gamma\bar{P}$
Calculate y^{**}
Form $P^{**} = \beta P_h + (1-\beta)\bar{P}$
Calculate y^{**}
is $y^{**} < y_l$?
No
is $y^{**} > y_h$?
Yes
Yes
No
Replace all P_i's by $(P_i + P_l)/2$
Replace P_h by P^{**}
Replace P_h by P^*
Replace P_h by P^{**}
Has minimum been reached ?
Yes
EXIT
No

Source: Nelder and Mead (1965).

Here α is a positive constant called the reflection coefficient; β is called the contraction coefficient and lies between 0 and 1; γ is called the expansion coefficient. I use the recommended reflection factor of 1.0, a contraction factor equal of 0.5, and an expansion factor equal of 2.0. (Nelder and Mead 1965)

Chapter 3. Results

3.1 – Programming:

3.11 – Programming Time-To-Event Data:

First, I used an if-then statement to determine whether each participant was a LTS. If the uniform value $U_i \le LTS$, then the subject was set to be an LTS (E_i = Max Value, $c_i = C$) and marked it censored. For each participant an indicator was assigned to indicate if a lack of censoring was observed (i.e., $E_i \le c_i$).

I checked the properties of the procedure by generating a group of size 10,000 and confirming the sample mean and variance closely matched the pattern parameters. I also ran a 14-group chi-square goodness of fit test. These results can be found in Table 3.1.

I checked the proportion of censoring by generating 50 replicates of time-to-event data and calculating the sample proportion of participants censored for each. I then ran a t test to check that the average sample proportion was consistent with the expected censoring rate. A sample of these results can be seen in Table 3.2.

Table 3.1: Chi-Square Fit Test for Distributions

Distribution Tested	Mean	Variance	Chi-Square Test Stat	p-value
Exponential(1)	1.004	1.009	9.72	0.716
Exponential(5)	4.982	5.021	7.56	0.871
Exponential(10)	10.04	10.09	10.21	0.677
Uniform(0,10)	5.02	2.93	21.25	0.323
Uniform(0, 5)	2.24	1.48	17.35	0.566

Table 3.2: Average Fraction Censored Compared To Expected Censoring Rate

Censoring Pattern	Expected Censoring Rate	Average Censored	t Stat	p-value
Exponential	10%	10.05%	0.533	0.596
Exponential	30%	29.68%	-1.38	0.1743
Uniform	10%	9.94%	-0.633	0.529
Uniform	30%	30.2%	0.769	0.446

3.12 – Programming MLE Calculations:

In methods I discussed the procedure for determining the number of random starting points used for general simulation situations. When calculating the MLE in my c-programming I used a publically available version of the NM algorithm (GSL-Website, 2007). This procedure was checked with grid searches globally and locally. Results are discussed in detail in section 3.22.

3.2 – Single Exponential w/ LTS:

3.21 – Null distribution of LRTS for L_S vs L_0

I begin with the LRTS for model L_S (single exponential with LTS) against model $L_{0.}$ I set the null hypothesis as L_0, where the survival time follows a single exponential and the alternative hypothesis is L_S, where there survival time follows a single exponential with a LTS proportion equal to 1-*p*. As discussed in methods, the LRTS is $d_n = -2(\log \hat{L}_{H00} - \log \hat{L}_{H10})$ and is calculated using the NM algorithm with 9 random starting points (in a 3 by 3 grid). I calculated 1000 replications per setting.

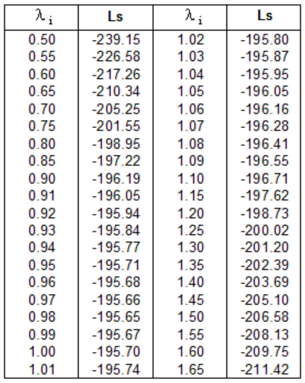
3.22 – Global and Local Maximum checks of maximization software

Recall $L_S(t_1, t_2, \ldots, t_n, \lambda, (LTS)) =$

$\prod_{i=1}^{n} \left((LTS)\lambda e^{-\lambda t_i}\right)^{(c_i)} \left(1-(LTS)+(LTS)e^{-\lambda t_i}\right)^{(1-c_i)}$. To test the validity of the maximizing routine, I took a sample of L_S output from the routine. For example, the sub-routine in C++ for a sample of 1000 was generated with uniform censoring at a 10% censoring rate and 5% LTS for a sample of 200. The log-likelihood value was -195.468 calculated from an MLE for λ equal 0.9877 and a $\hat{p}$ of 0.9615. Using Excel, I calculated L_S (λ_i, p_j) for a grid of λ_i and p_j values. For each λ_i I calculated L_S (λ_i, p_j) with p_j = 0.01, 0.05, 0.10, 0.15, … , 0.85, 0.90, 0.95 and 0.99 and report L_S (λ_i, ·) = $\max_{p_j}$ [L_S (λ_i, p_j)], as shown in Figure 3.1. I use $0.5\hat{\lambda}, 0.55\hat{\lambda}, \ldots, 0.90\hat{\lambda}, 0.91\hat{\lambda}, \ldots, 1.10\hat{\lambda}, 1.15\hat{\lambda}, 1.65\hat{\lambda}$, for the grid λ_i. I then find L_S (· , ·) = $\max_{\lambda_i}$ [L_S (λ_i, ·)] and calculate L_S ($\hat{\lambda}$, $\hat{p}$) - L_S (· , ·). The maximum log-likelihood found by the subroutine, -195.468, is larger than the largest log-likelihood found by the global grid search ($L_S(-195.65, 0.98)$).

Additionally, I used a grid search of the neighborhood of ($\hat{\lambda}$, $\hat{p}$). The settings of p$_j$ are p$_j$ = $\hat{p}-0.005$, $\hat{p}-0.0045$, … , $\hat{p}+0.005$. The settings of λ_i are $\lambda_i = 0.98\hat{\lambda}$, $0.981\hat{\lambda}$, … , $1.02\hat{\lambda}$. A local minimum grid search was done by using the exact MLE value calculated by the sub-routine and testing 20 values from $(\hat{p}-0.005, \hat{p}+0.005)$ within 0.01 of the calculated p-hat value to ensure the minimum was found as shown in Figure 3.2.

Figure 3.1: Plot of $L_S(\lambda_i,\cdot)$, $\lambda_i = 0.50, 0.55, \ldots, 1.65$



λ_i	Ls	λ_i	Ls
0.50	-239.15	1.02	-195.80
0.55	-226.58	1.03	-195.87
0.60	-217.26	1.04	-195.95
0.65	-210.34	1.05	-196.05
0.70	-205.25	1.06	-196.16
0.75	-201.55	1.07	-196.28
0.80	-198.95	1.08	-196.41
0.85	-197.22	1.09	-196.55
0.90	-196.19	1.10	-196.71
0.91	-196.05	1.15	-197.62
0.92	-195.94	1.20	-198.73
0.93	-195.84	1.25	-200.02
0.94	-195.77	1.30	-201.20
0.95	-195.71	1.35	-202.39
0.96	-195.68	1.40	-203.69
0.97	-195.66	1.45	-205.10
0.98	-195.65	1.50	-206.58
0.99	-195.67	1.55	-208.13
1.00	-195.70	1.60	-209.75
1.01	-195.74	1.65	-211.42

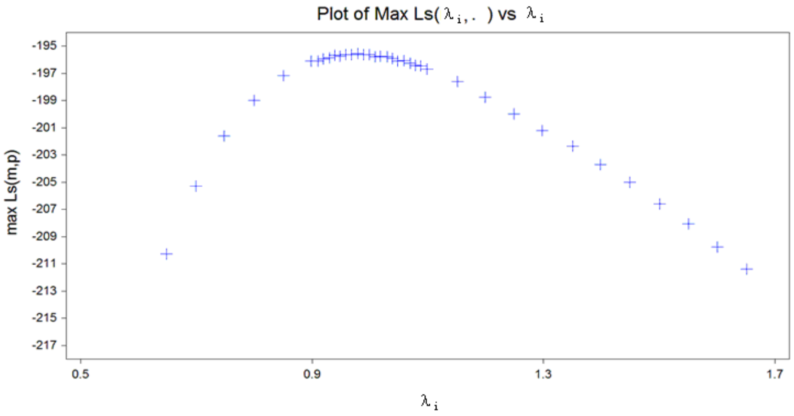
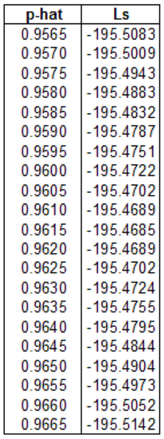
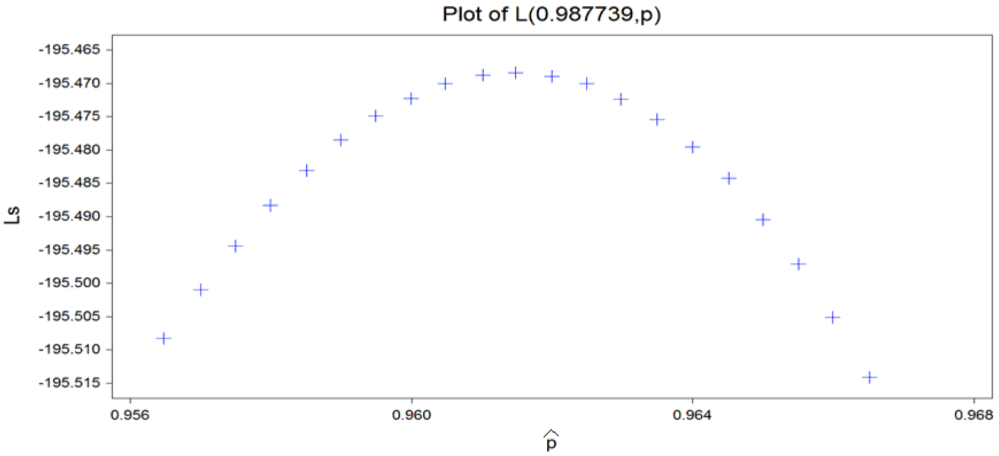


Figure 3.2: Plot of $L_S(0.987739, p)$, $\hat{p} = 0.9565, 0.9570, \ldots, 0.9665$

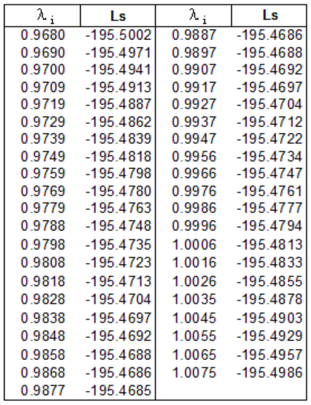


p-hat	Ls
0.9565	-195.5083
0.9570	-195.5009
0.9575	-195.4943
0.9580	-195.4883
0.9585	-195.4832
0.9590	-195.4787
0.9595	-195.4751
0.9600	-195.4722
0.9605	-195.4702
0.9610	-195.4689
0.9615	-195.4685
0.9620	-195.4689
0.9625	-195.4702
0.9630	-195.4724
0.9635	-195.4755
0.9640	-195.4795
0.9645	-195.4844
0.9650	-195.4904
0.9655	-195.4973
0.9660	-195.5052
0.9665	-195.5142

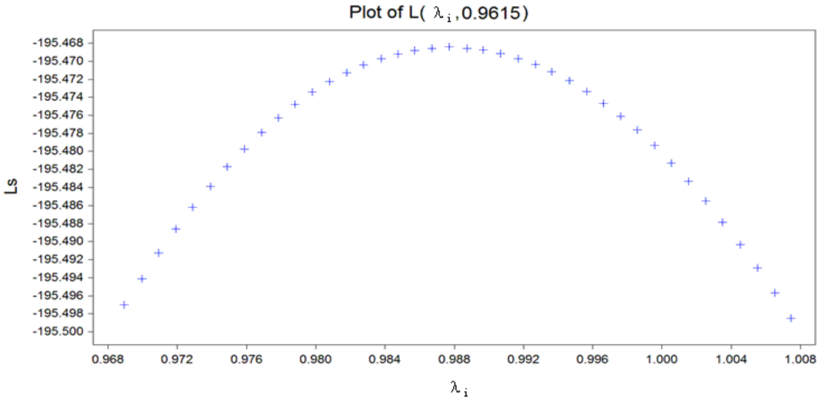


Next I tested the subroutine's max likelihood estimate of the λ parameter (which was 0.987739). In the local grid search for $\hat{p}$, the max log-likelihood estimate of -195.4685 at a p-hat of 0.9615 is plausibly the maximum as the local search found the same maximum at the calculated p-hat. Next I tested the subroutine's max likelihood estimate of the p parameter (which was 0.9615) which can be seen in Figure 3.3.

Figure 3.3: Plot of $L_S(\lambda, 0.9615)$, $0.968 \le \lambda \le 1.0075$



λ_i	Ls	λ_i	Ls
0.9680	-195.5002	0.9887	-195.4686
0.9690	-195.4971	0.9897	-195.4688
0.9700	-195.4941	0.9907	-195.4692
0.9709	-195.4913	0.9917	-195.4697
0.9719	-195.4887	0.9927	-195.4704
0.9729	-195.4862	0.9937	-195.4712
0.9739	-195.4839	0.9947	-195.4722
0.9749	-195.4818	0.9956	-195.4734
0.9759	-195.4798	0.9966	-195.4747
0.9769	-195.4780	0.9976	-195.4761
0.9779	-195.4763	0.9986	-195.4777
0.9788	-195.4748	0.9996	-195.4794
0.9798	-195.4735	1.0006	-195.4813
0.9808	-195.4723	1.0016	-195.4833
0.9818	-195.4713	1.0026	-195.4855
0.9828	-195.4704	1.0035	-195.4878
0.9838	-195.4697	1.0045	-195.4903
0.9848	-195.4692	1.0055	-195.4929
0.9858	-195.4688	1.0065	-195.4957
0.9868	-195.4686	1.0075	-195.4986
0.9877	-195.4685		



In the local grid search for λ, the max log-likelihood estimate of -195.4685 at a p-hat of 0.9615 is plausibly the maximum value, as the local search found the same maximum at the calculated intensity level.

3.23 –Estimated Null Percentiles For Test of LTS:

The fraction of zero LRTS, $\hat{\pi}_0$, is between 0.48 and 0.74 with average fraction of zero LRTS equal to 0.56. The fraction of zero LRTS values decreases on average as *n* increases and appears to approach 0.50 asymptotically. The mean of the non-zero LRTS appears to be 1. Tables 3.3 and 3.4 summarize the results of the null distribution of the LRTS. In my summary, I consider an LRTS value zero as an LRTS value less than 0.0001.

Table 3.3:
Table 3.11: Mean, Variance and Simulated Percentiles of the null distribution of LRT. (L_S) (exponential censoring)

Sample Size	Study Length	Censoring Rate	Mean	Variance	$\hat{\pi}_0$: Fraction of zero LRTS	Mean of Non-Zero LRTS	Percentile			
							75%	90%	95%	99%
50	Short	10%	0.419	0.959	0.556	0.944	0.342	1.515	2.232	4.642
		30%	0.488	1.259	0.549	1.083	0.437	1.555	2.591	5.01
		50%	0.412	0.992	0.619	1.081	0.22	1.502	2.467	4.997
	Long	10%	0.436	1.379	0.657	1.272	0.163	1.421	2.918	5.803
		30%	0.336	0.756	0.64	0.933	0.143	1.111	2.057	4.639
		50%	0.37	0.975	0.655	1.073	0.136	1.219	2.158	5.466
100	Short	10%	0.415	0.89	0.545	0.913	0.32	1.375	2.252	4.732
		30%	0.477	1.135	0.54	1.037	0.447	1.523	2.762	4.956
		50%	0.406	0.923	0.579	0.965	0.298	1.369	2.178	4.508
	Long	10%	0.405	0.948	0.654	1.171	0.277	1.434	2.2	5.214
		30%	0.382	1.018	0.686	1.216	0.078	1.348	2.393	5.224
		50%	0.436	1.429	0.648	1.239	0.138	1.446	2.835	5.495
200	Short	10%	0.429	0.965	0.544	0.941	0.390	1.391	2.454	4.571
		30%	0.445	1.346	0.511	0.91	0.306	1.343	2.282	5.235
		50%	0.509	1.464	0.531	1.086	0.414	1.77	2.639	6.065
	Long	10%	0.385	1.067	0.603	0.97	0.185	1.205	2.155	5.449
		30%	0.405	1.365	0.642	1.131	0.174	1.308	2.348	5.385
		50%	0.407	1.244	0.624	1.082	0.166	1.276	2.616	5.089
500	Short	10%	0.521	1.55	0.529	1.105	0.432	1.712	3.04	5.485
		30%	0.484	1.103	0.511	0.99	0.454	1.593	2.668	5.173
		50%	0.477	1.297	0.532	1.018	0.442	1.512	2.354	5.298
	Long	10%	0.449	0.998	0.529	0.952	0.353	1.525	2.504	4.871
		30%	0.375	0.926	0.593	0.921	0.198	1.3	2.272	5.282
		50%	0.384	0.87	0.608	0.98	0.19	1.347	2.406	4.252
1000	Short	10%	0.402	0.886	0.542	0.878	0.346	1.247	2.151	4.855
		30%	0.511	1.375	0.522	1.07	0.406	1.799	2.728	6.046
		50%	0.46	1.303	0.511	0.94	0.344	1.335	2.319	5.742
	Long	10%	0.431	0.944	0.508	0.875	0.365	1.4	2.208	4.909
		30%	0.462	1.303	0.561	1.051	0.362	1.535	2.476	4.719
		50%	0.434	1.188	0.572	1.013	0.273	1.282	2.616	5.678

Sample Size	Study Length	Censoring Rate	Mean	Variance	$\hat{\pi}_0$: Fraction of zero LRTS	Mean of Non-Zero LRTS	Percentiles			
							75%	90%	95%	99%
2000	Short	10%	0.475	1.142	0.527	1.005	0.492	1.659	2.502	4.419
		30%	0.444	1.007	0.519	0.924	0.396	1.485	2.34	5.457
		50%	0.46	0.953	0.508	0.935	0.459	1.579	2.247	5.102
	Long	10%	0.497	1.258	0.494	0.983	0.441	1.544	2.44	5.6
		30%	0.422	0.995	0.559	0.958	0.273	1.478	2.374	4.889
		50%	0.402	0.952	0.582	0.963	0.313	1.416	2.223	4.832
10000	Short	10%	0.523	1.412	0.493	1.031	0.468	1.59	2.929	5.844
		30%	0.53	1.31	0.5	1.06	0.524	1.644	2.873	5.479
		50%	0.491	1.071	0.501	0.985	0.491	1.622	2.59	4.952
	Long	10%	0.473	1.134	0.525	0.9965	0.383	1.53	2.603	5.527
		30%	0.525	1.59	0.545	1.154	0.359	1.774	2.949	6.286
		50%	0.432	1.195	0.535	0.931	0.337	1.309	2.535	4.956

Table 3.4: Mean, Variance and Simulated Percentiles of the null distribution of LRT. (L_S) (uniform censoring).

Sample Size	Study Length	Censoring Rate	Mean	Variance	$\hat{\pi}_0$: Fraction of zero LRTS	Mean of Non-Zero LRTS	Percentile			
							75%	90%	95%	99%
50	Short	10%	0.467	1.089	0.542	1.02	0.384	1.654	2.587	4.891
		30%	0.466	1.258	0.579	1.108	0.339	1.604	2.748	5.616
	Long	10%	0.364	0.964	0.740	1.398	0.012	1.311	2.274	5.463
100	Short	10%	0.465	1.198	0.513	0.955	0.417	1.437	2.370	4.978
		30%	0.467	1.447	0.56	1.062	0.3	1.453	2.645	5.105
	Long	10%	0.465	1.312	0.632	1.264	0.364	1.508	2.483	5.354
200	Short	10%	0.518	1.555	0.505	1.048	0.496	1.645	2.596	6.189
		30%	0.425	1.166	0.557	0.96	0.272	1.376	2.613	4.936
	Long	10%	0.426	1.151	0.551	0.949	0.264	1.422	2.299	5.688
500	Short	10%	0.414	0.88	0.534	0.889	0.382	1.19	2.4	4.509
		30%	0.458	1.178	0.523	0.96	0.383	1.388	2.495	5.483
	Long	10%	0.38	0.992	0.585	0.916	0.255	1.217	2.089	4.653
1000	Short	10%	0.443	1.173	0.535	0.953	0.33	1.457	2.438	5.205
		30%	0.541	1.365	0.484	1.048	0.597	1.77	2.56	5.389
	Long	10%	0.447	1.095	0.523	0.938	0.39	1.44	2.291	4.788
2000	Short	10%	0.493	1.306	0.52	1.026	0.425	1.542	2.64	5.774
		30%	0.5	1.392	0.508	1.017	0.436	1.585	2.808	5.289
	Long	10%	0.481	1.11	0.521	1.005	0.403	1.632	2.816	5.002
10000	Short	10%	0.449	0.933	0.516	0.929	0.395	1.518	2.614	4.551
		30%	0.551	1.374	0.489	1.079	0.462	1.812	3.134	6.26
	Long	10%	0.469	1.116	0.505	0.948	0.436	1.145	2.435	5.4

The function fit to the fraction of zero LRTS was dependent on sample size ($t = 8.40$, $p \approx 0$) and the interaction of sample size and study duration D ($t = 10.60$, $p \approx 0$). In a short study ($D = 3$) $I_D = 0$ and in a long study study ($D = 5$) $I_D = 1$. The function $0.50 + \frac{0.4864 + 0.9206 I_D}{\sqrt{n}}$ explains 79.1% of the variation when predicting the fraction of zero LRTS values. The coefficients of the censoring rate variable (p = 0.90) and censoring pattern indicator variable (p = 0.33) were not significant.

Equation 3.1:

$$0.50 + \frac{0.4864 + 0.9206 I_D}{\sqrt{n}}$$

When long study and short study results were fitted separately, both the long and short study duration intercepts appeared to be 0.50, with the rate of convergence for the longer study duration being slower as shown in Figure 3.4. The fit for the fraction of zero LRTS values for the longer study duration is $0.50 + \frac{1.382}{\sqrt{n}}$, which explained 71% of the variation in the model. The fit for the shorter study duration is $0.50 + \frac{0.4846}{\sqrt{n}}$, which explained 55% of the variation in the model. The plot of residuals versus fitted values for these two models are shown in Figures 3.5 and 3.6. I decided to model the fraction of zero LRTS values for the null distribution of the LRTS as a function of only the sample size and interaction of sample size and study duration so that it matched the Zhou and Maller asymptotic result. The study duration itself was marginally significant at the 5% significance level ($t = 2.1$, $p \approx 0.04$).

Figure 3.4 Observed Fraction of Zero LRTS Values (Long Study and Short Study)

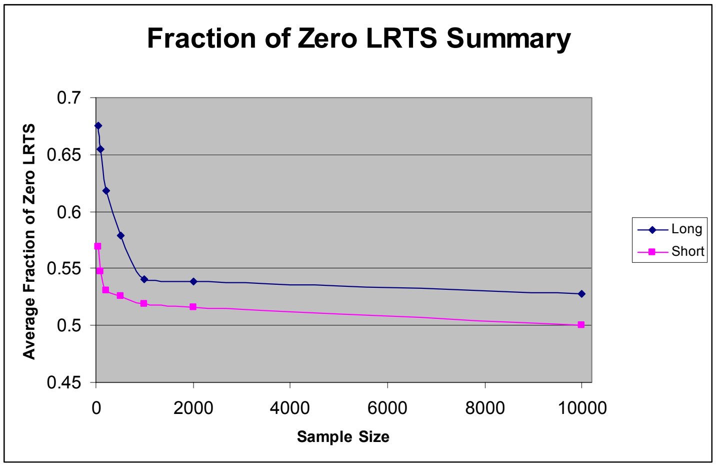


Figure 3.5 Residual vs Fit of Fraction of Zero LRTS Values

Long Study, Fit $0.50 + \dfrac{1.382}{\sqrt{n}}$

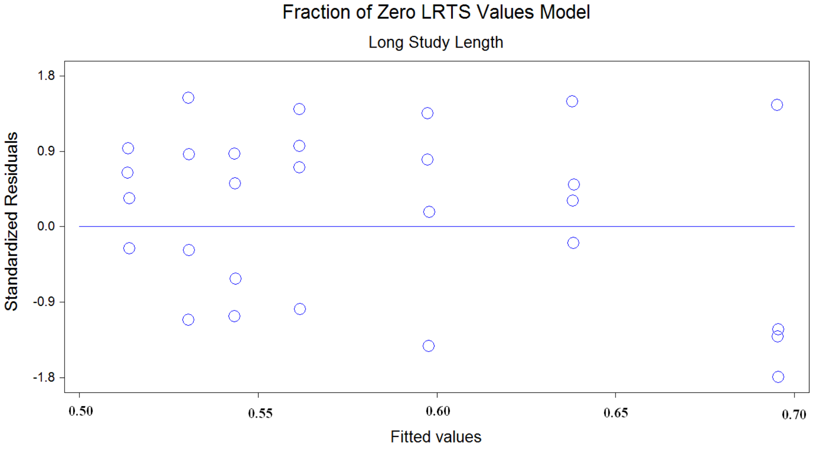
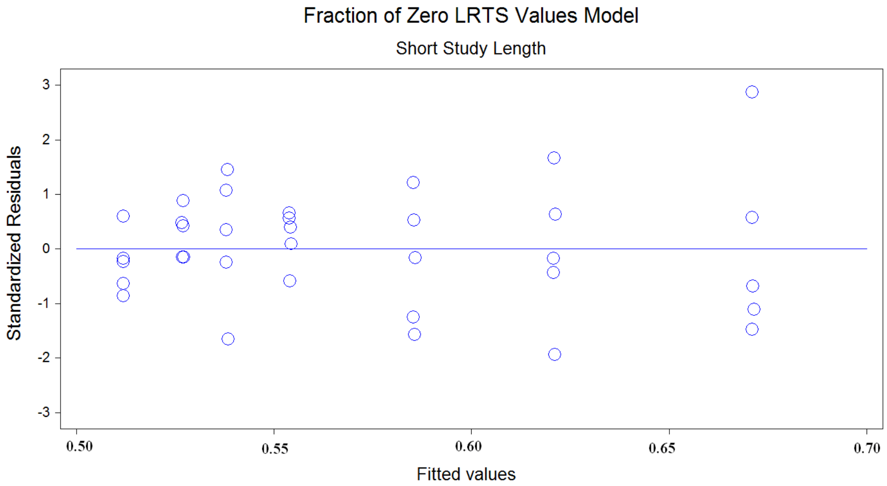


Figure 3.6 Residual vs Fit of Fraction of Zero LRTS Values

Short Study, Fit $0.50+\frac{0.4846}{\sqrt{n}}$



The mean of the non-zero LRTS values ranged from a minimum of 0.88 to a maximum of 1.40. The average of all non-zero LRTS values was 1.02 ($\sigma^2 = 0.011$) and 50% of simulations had a non-zero mean between 0.94 and 1.07. The average variance of the non-zero LRTS values was 2.05 with a standard deviation of 0.39.

As noted in chapter one, Zhou and Maller (1995) showed that the asymptotic null distribution is an equal mixture of a chi-square random variable with 1 degree of freedom and a mass at zero, $\frac{1}{2}\chi_0^2+\frac{1}{2}\chi_1^2$. The fraction of zeros is apparently 0.50 asymptotically, the mean of the non-zero LRTS values is near

1, and the average variance is near 2 ($\sigma^2 = 2.048$). My simulation results are consistent with the asymptotic distribution. To test, I selected eight simulations and ran a goodness of fit test to test whether a chi-square random variable with one degree of freedom described the non-zero LRTS values. I used 20 classes, 16 of width 0.25 (ranging from 0 through 4), 2 of width 1 (ranging from 4 through 6), 1 interval from 6 to 8, and the last from 8 to infinity. In each of the 8 distributions tested the null hypothesis of the chi-square one random variable was accepted. P-values ranged from 18.5% to 95.5%. These tests are summarized in Table 3.5.

Table 3.5 Chi Square Goodness of Fit Test
Non-Zero LRTS Values Fitted to χ_1^2

Sample Size	Censoring Distribution	Censoring Rate	Study Duration	Chi-Square Test Stat	P-value
500	Exponential	50%	5.0	9.95	0.954
1000	Exponential	30%	3.0	24.29	0.185
2000	Exponential	50%	3.0	21.56	0.307
2000	Exponential	30%	5.0	12.03	0.884
2000	Uniform	30%	3.0	9.90	0.955
10000	Exponential	30%	3.0	19.99	0.395
10000	Exponential	30%	5.0	22.65	0.253
10000	Uniform	10%	3.0	15.74	0.674

I estimated the null percentiles using the distribution $\pi_0\chi_0^2 + (1-\pi_0)\chi_1^2$, where π_0 is the estimated fraction of zero LRTS as given in equation 3.1. The estimated null percentiles and the Maller-Zhou percentiles are summarized in Table 3.6. The percentiles of the Maller-Zhou asymptotic distribution appear to be conservative. That is, the critical values estimated from the fitted distribution are smaller than those of the asymptotic distribution. The actual proportion of LRTS values greater than my calculated 99th percentile are reported in Table 3.7. For sample sizes over 500, there is little difference between the fitted percentiles and the asymptotic distribution. For the smaller sample sizes that are more common in clinical trials, the estimated critical values will result in increased power. A scatter plot for the estimated percentiles versus the simulated percentiles can be seen in Figure 3.7. The estimated percentiles had a coefficient of 0.992 and a model R^2 of 96.9%.

Table 3.6: Fitted Null Percentile Points For L_S

Sample Size	Study Duration	75%	90%	95%	**99%**	99.9%
50	0	0.31	1.43	2.47	**5.15**	9.28
	1	0.05	0.94	1.92	**4.53**	8.62
100	0	0.35	1.49	2.54	**5.23**	9.36
	1	0.15	1.18	2.19	**4.84**	8.94
200	0	0.38	1.54	2.59	**5.29**	9.42
	1	0.24	1.33	2.36	**5.03**	9.14
350	0	0.40	1.56	2.62	**5.32**	9.45
	1	0.29	1.41	2.45	**5.13**	9.25
500	0	0.41	1.58	2.63	**5.33**	9.47
	1	0.32	1.45	2.49	**5.18**	9.30
750	0	0.42	1.59	2.65	**5.35**	9.48
	1	0.34	1.49	2.53	**5.22**	9.35
1000	0	0.42	1.60	2.66	**5.36**	9.49
	1	0.36	1.51	2.56	**5.25**	9.38
2000	0	0.43	1.61	2.67	**5.37**	9.51
	1	0.39	1.55	2.60	**5.30**	9.43
10000	0	0.44	1.63	2.69	**5.39**	9.53
	1	0.42	1.60	2.66	**5.36**	9.50
Maller-Zhou		0.45	1.64	2.71	**5.41**	9.55

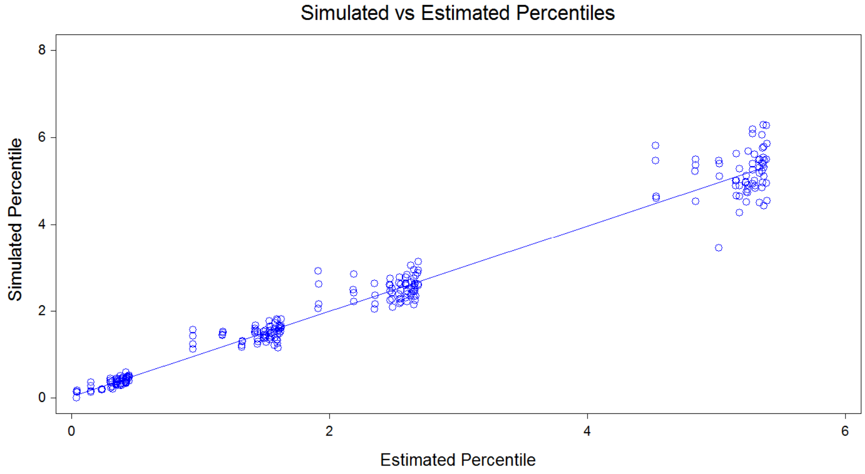
Note: Fit $\pi_0\chi_0^2+(1-\pi_0)\chi_1^2$, where $\pi_0 = 0.50+\frac{0.4864+0.9206I_D}{\sqrt{n}}$

Table 3.7: Proportion of LRTS Observed in Simulation Greater Than Estimated 99th Percentile

Sample Size	Censoring Distribution	Study Duration	Censoring Rate	Proportion above 99th percentile
50	Exponential	Short	10%	0.5%
		Short	30%	1.0%
		Short	50%	0.9%
		Long	10%	1.7%
		Long	30%	1.2%
		Long	50%	1.5%
	Uniform	Short	10%	0.8%
		Short	30%	1.2%
		Long	10%	1.4%
100	Exponential	Short	10%	0.0%
		Short	30%	0.9%
		Short	50%	0.7%
		Long	10%	1.3%
		Long	30%	1.3%
		Long	50%	1.3%
	Uniform	Short	10%	0.7%
		Short	30%	0.7%
		Long	10%	1.3%
200	Exponential	Short	10%	0.9%
		Short	30%	1.1%
		Short	50%	0.9%
		Long	10%	1.3%
		Long	30%	0.7%
		Long	50%	1.1%
	Uniform	Short	10%	1.1%
		Short	30%	0.0%
		Long	10%	1.1%
500	Exponential	Short	10%	1.1%
		Short	30%	0.7%
		Short	50%	0.9%
		Long	10%	0.4%
		Long	30%	0.9%
		Long	50%	0.4%
	Uniform	Short	10%	0.4%
		Short	30%	1.1%
		Long	10%	1.1%
1000	Exponential	Short	10%	0.4%
		Short	30%	1.5%
		Short	50%	0.9%
		Long	10%	1.3%
		Long	30%	0.7%
		Long	50%	1.3%
	Uniform	Short	10%	1.1%
		Short	30%	1.3%
		Long	10%	0.7%
2000	Exponential	Short	10%	1.3%
		Short	30%	0.7%
		Short	50%	0.7%
		Long	10%	1.1%
		Long	30%	1.3%
		Long	50%	1.3%
	Uniform	Short	10%	1.3%
		Short	30%	0.9%
		Long	10%	0.0%
10000	Exponential	Short	10%	1.4%
		Short	30%	1.0%
		Short	50%	0.8%
		Long	10%	1.4%
		Long	30%	1.1%
		Long	50%	0.5%
	Uniform	Short	10%	0.5%
		Short	30%	1.3%
		Long	10%	1.1%

(Standard Error = 0.03)

Figure 3.7: Observed Simulated LRTS Values vs Estimated LRTS Values for Summarized Percentiles.



Note: The four clusters are the 75th, 90th, 95th and 99th percentiles.

3.24 –Power:

There were 1000 replicates for each setting. I calculated the average LRTS, the fraction of LRTS values with value nearly equal to zero, the average of the non-zero LRTS values, and the power using an alpha level of 1% with the null percentiles in Table 3.6. Tables 3.8 and 3.9 contain the power of L_S vs L_0. As expected the sample size has a great effect on the power. For very small samples sizes (200) the proportion of situations were power exceeded 50% was less than 25%. When the sample size was 2000, over 75% of situations had power exceeding 50%. In samples of 500 or more, over 25% of situations had power exceeding 95%. In small samples (200 and 350), only 13 percent of situations had power exceeding 95%. The average power for the sample size 200 was 33.8%, and the average power was to 81.7% for the sample size 2000. When comparing a smaller study duration to a longer study duration, the dispersion of the power was similar ($\sigma = 0.35$ and $\sigma = 0.32$). The longer study duration had an average factor increase of 1.55 power over the smaller study duration. As the censoring rate increased, the power, on average, decreased.

Table 3.8: Simulated Power and Summary Statistics of the LRTS of L_S vs L_0 (exponential censoring)

Sample size	Study Length	Censoring rate	LTS	avg	var	0's	Non 0 avg	75%	90%	95%	99%	Power
200	3.0	15%	2%	1.29	3.74	0.269	1.76	1.90	3.73	5.11	8.83	0.048
			5%	3.41	11.18	0.057	3.61	5.01	7.97	9.61	14.48	0.233
			8%	6.92	25.66	0.01	6.98	9.80	13.91	16.40	22.51	0.565
		30%	2%	1.08	3.38	0.33	1.61	1.38	3.25	4.93	7.98	0.043
			5%	2.37	7.26	0.12	2.67	3.63	6.07	7.35	10.73	0.140
			8%	4.58	15.98	0.022	4.68	6.41	9.93	12.07	18.96	0.346
		45%	2%	0.82	2.40	0.402	1.36	0.95	2.60	4.24	6.97	0.029
			5%	1.51	5.06	0.221	1.94	2.15	4.21	5.81	10.00	0.067
			8%	2.51	8.94	0.107	2.81	3.64	6.21	8.52	12.60	0.147
	5.0	15%	2%	3.64	16.12	0.096	4.03	5.45	8.99	11.32	17.31	0.269
			5%	12.26	57.74	0.001	12.27	16.26	22.53	26.38	34.78	0.833
			8%	23.56	98.30	0	23.56	29.72	37.32	40.93	49.61	0.988
		30%	2%	1.98	7.47	0.287	2.77	3.09	5.78	7.60	11.71	0.123
			5%	6.02	27.69	0.049	6.34	8.84	13.20	15.70	23.06	0.476
			8%	11.72	52.44	0.006	11.79	16.04	21.45	25.55	32.47	0.808
		45%	2%	1.07	3.68	0.436	1.89	1.36	3.34	5.21	8.47	0.052
			5%	2.49	10.04	0.213	3.16	3.62	6.87	9.57	13.45	0.164
			8%	4.59	19.28	0.093	5.07	6.99	10.75	13.09	17.04	0.369
350	3.0	15%	2%	1.77	5.49	0.197	2.21	2.52	4.77	6.72	10.19	0.088
			5%	5.49	19.87	0.016	5.58	7.85	11.71	13.75	19.53	0.44
			8%	11.28	40.08	0	11.28	15.15	19.65	22.46	29.03	0.828
		30%	2%	1.26	3.35	0.25	1.68	1.76	3.63	5.15	8.14	0.046
			5%	3.55	13.29	0.057	3.76	5.06	7.98	10.66	17.23	0.237
			8%	7.37	28.17	0.004	7.40	10.08	14.56	18.04	23.12	0.581
		45%	2%	0.930	2.52	0.342	1.413	1.26	2.94	4.17	7.71	0.030
			5%	2.16	7.27	0.143	2.52	3.24	5.57	7.25	12.45	0.108
			8%	4.20	16.13	0.041	4.38	6.18	9.66	11.89	17.28	0.313
	5.0	15%	2%	5.69	25.59	0.032	5.88	8.08	12.73	15.62	22.59	0.444
			5%	20.94	98.52	0	20.94	26.74	34.42	39.02	49.24	0.972
			8%	41.98	192.3	0	41.98	49.96	59.10	65.87	82.32	1
		30%	2%	2.96	11.90	0.137	3.44	4.29	7.67	10.16	15.85	0.186
			5%	10.04	51.38	0.013	10.17	14.22	19.77	23.24	30.99	0.701
			8%	19.98	99.51	0	19.98	26.25	33.52	37.81	49.38	0.960
		45%	2%	1.31	4.82	0.357	2.04	1.87	4.17	5.50	10.30	0.057
			5%	3.69	14.68	0.128	4.23	5.44	8.97	11.03	16.70	0.265
			8%	7.28	35.97	0.031	7.52	10.53	15.51	19.41	25.56	0.551

Sample size	Study Length	Censoring rate	LTS	avg	var	0’s	Non 0 avg	75%	90%	95%	99%	Power
500	3.0	15%	2%	2.21	6.60	0.131	2.54	3.43	5.73	7.62	10.97	0.122
			5%	7.32	27.91	0.008	7.39	10.25	14.25	17.64	23.60	0.591
			8%	15.93	61.43	0	16.93	20.65	25.90	30.42	39.46	0.95
		30%	2%	1.62	4.71	0.209	2.05	2.41	4.45	6.11	9.39	0.08
			5%	5.19	20.30	0.017	5.28	7.26	11.39	13.57	19.98	0.398
			8%	9.84	37.80	0.002	9.86	13.38	17.97	21.63	29.59	0.747
		45%	2%	1.01	2.88	0.322	1.49	1.30	3.27	4.40	8.65	0.028
			5%	2.81	9.53	0.100	3.12	4.14	6.97	8.96	13.60	0.178
			8%	5.30	18.95	0.027	5.44	7.54	11.28	13.62	18.72	0.427
	5.0	15%	2%	7.85	34.32	0.005	7.89	10.83	15.71	19.25	26.67	0.600
			5%	30.39	155.77	0	30.39	37.53	46.50	53.03	66.62	0.998
			8%	59.40	253.1	0	59.40	70.29	79.29	85.50	98.24	1
		30%	2%	3.68	16.09	0.092	4.05	5.28	9.08	11.79	17.51	0.252
			5%	13.40	67.31	0.004	13.45	18.23	24.90	28.36	36.20	0.84
			8%	28.33	133.6	0	28.33	35.29	43.93	49.31	59.47	0.991
		45%	2%	1.68	6.59	0.296	2.38	2.40	5.02	6.83	11.71	0.093
			5%	4.82	22.83	0.078	5.22	7.08	11.08	14.19	21.17	0.359
			8%	10.13	49.24	0.008	10.21	13.91	19.18	23.63	31.14	0.723
750	3.0	15%	2%	2.78	8.69	0.74	2.30	4.09	6.87	8.82	12.63	0.166
			5%	11.00	42.86	0.001	11.01	14.99	20.08	23.53	28.25	0.785
			8%	23.23	84.85	0	23.23	28.84	35.86	39.55	46.83	0.991
		30%	2%	2.05	6.87	0.173	2.47	3.04	5.57	7.35	11.11	0.106
			5%	6.41	23.35	0.013	6.50	9.20	13.05	15.67	20.72	0.505
			8%	14.6	60.15	0	14.6	18.88	25.89	29.44	35.82	0.921
		45%	2%	1.29	4.10	0.268	1.77	1.75	3.69	5.28	9.22	0.05
			5%	3.61	13.15	0.073	3.89	5.44	8.44	10.41	15.27	0.255
			8%	7.60	27.48	0.003	7.62	10.57	14.82	17.49	23.27	0.619
	5.0	15%	2%	11.35	50.78	0.003	11.38	15.26	20.48	25.06	32.80	0.791
			5%	46.24	215.65	0	46.24	55.70	65.48	72.53	85.97	1
			8%	89.30	393.6	0	89.30	102.7	116.7	123.6	135.3	1
		30%	2%	5.11	22.70	0.057	5.42	7.73	11.67	14.83	20.55	0.376
			5%	20.02	99.74	0	20.02	25.87	33.69	38.59	47.84	0.952
			8%	41.97	205.0	0	41.97	51.41	60.94	65.4	77.9	1
		45%	2%	1.92	7.18	0.214	2.44	2.72	5.36	7.31	11.65	0.102
			5%	7.01	33.81	0.03	7.22	9.76	14.59	18.54	24.86	0.530
			8%	14.3	67.6	0.002	14.33	19.18	25.0	28.82	37.98	0.873

Sample size	Study Length	Censoring rate	LTS	avg	var	0’s	Non 0 avg	75%	90%	95%	99%	Power
1000	3.0	15%	2%	3.57	13.29	0.066	3.818	5.16	8.01	10.81	16.84	0.244
			5%	14.64	58.01	0.001	14.66	19.30	25.15	29.39	37.01	0.928
			8%	30.45	114	0	30.45	37.27	44.8	48.53	58.39	0.999
		30%	2%	2.3	6.85	0.112	2.59	3.36	5.74	7.19	11.95	0.124
			5%	8.71	36.67	0.004	8.75	11.90	16.63	19.64	29.02	0.668
			8%	19.91	79.09	0	19.9	25.3	31.7	35.5	46.1	0.980
		45%	2%	1.52	4.59	0.223	1.93	2.16	4.13	5.51	9.96	0.054
			5%	4.49	15.48	0.039	4.67	6.55	10.02	12.39	16.81	0.334
			8%	9.82	37.22	0.005	9.87	13.24	18.19	21.70	27.46	0.744
	5.0	15%	2%	14.88	76.38	0.001	14.90	19.8	26.63	30.96	41.52	0.879
			5%	59.83	263.0	0	59.83	70.25	81.90	87.97	102.62	1
			8%	118	531.3	0	118	132.5	147.5	155.9	178.0	1
		30%	2%	6.76	31.92	0.026	6.95	9.67	14.21	17.8	25.56	0.513
			5%	27.25	131.90	0	27.25	34.59	42.08	47.77	57.68	0.997
			8%	55.37	277.8	0	55.37	65.16	78.59	85.42	98.07	1
		45%	2%	2.44	9.38	0.177	2.97	3.59	6.79	8.79	13.44	0.155
			5%	8.99	38.39	0.011	9.09	12.94	17.67	20.85	26.10	0.669
			8%	19	87.5	0	19	24.8	32.19	35.58	43.06	0.957
2000	3.0	15%	2%	5.91	22.1	0.013	5.99	8.40	12.06	15.2	20.77	0.46
			5%	26.92	109.39	0	26.92	33.12	41.05	46.37	55.03	0.997
			8%	60.61	221.5	0	60.61	70.62	80.65	85.23	97.47	1
		30%	2%	3.8	14.67	0.056	4.0	5.51	8.79	11.59	17.4	0.26
			5%	16.32	63.02	0	16.32	21.48	27.12	30.64	37.65	0.95
			8%	37.32	136.2	0	37.32	44.41	52.78	57.28	68.64	0.999
		45%	2%	2.06	5.91	0.148	2.42	3.08	5.20	7.02	10.87	0.097
			5%	8.19	31.94	0.002	8.21	11.38	15.90	18.43	25.11	0.632
			8%	18.76	83.48	0	18.76	24.4	30.66	35.22	44.27	0.953
	5.0	15%	2%	28.45	141.5	0	28.45	35.4	44.8	49.7	60.4	0.993
			5%	118.2	587.9	0	118.2	132.9	150.1	160.9	181.4	1
			8%	236.6	1051	0	236.6	258	279.3	292.2	319.9	1
		30%	2%	12.4	59.7	0.003	12.46	16.8	22.6	27.7	34.9	0.823
			5%	53.2	285.2	0	53.2	62.7	75.6	84.2	100.8	1
			8%	110.3	499	0	110.3	124.8	138.9	149	167.2	1
		45%	2%	4.2	17.47	0.068	4.51	6.46	9.65	12.43	17.79	0.312
			5%	17.17	86.8	0.001	17.19	22.57	29.78	34.88	44.15	0.924
			8%	37.85	189.2	0	37.85	46.72	55.47	62.51	74.03	1

*(level of significance 0.01)

Table 3.9: Simulated Power and Summary Statistics of the LRTS of L_S vs L_0 (uniform censoring)

Sample size	Study Length	Censoring rate	LTS	avg	var	0's	Non 0 avg	75%	90%	95%	99%	Power
200	3.0	15%	2%	1.4	4.05	0.255	1.88	2.10	3.94	5.33	9.08	0.056
			5%	3.44	12.24	0.068	3.69	4.82	7.78	10.61	15.39	0.224
			8%	6.69	23.19	0.012	6.78	9.33	13.50	16.25	20.68	0.552
		30%	2%	1.06	3.23	0.326	1.57	1.29	3.52	4.60	7.80	0.037
			5%	2.28	7.29	0.135	2.63	3.34	6.08	7.67	11.29	0.132
			8%	4.7	16.92	0.04	4.89	6.61	10.11	12.89	18.53	0.361
	5.0	15%	2%	3.32	12.73	0.113	3.74	5.09	8.44	10.26	14.93	0.246
			5%	12.01	56.68	0.002	12.04	16.08	21.84	25.95	33.71	0.819
			8%	23.75	104.5	0	23.75	30.04	37.63	41.98	52.96	0.98
350	3.0	15%	2%	1.94	6.86	0.185	2.38	2.86	4.84	6.97	12.41	0.085
			5%	5.57	21.14	0.01	5.63	7.88	11.84	14.83	19.79	0.44
			8%	11.29	41.35	0	11.29	15.19	20.11	23.04	29.91	0.818
		30%	2%	1.3	4.14	0.269	1.78	1.82	3.88	5.79	9.33	0.06
			5%	3.54	13.1	0.051	3.73	5.23	8.50	10.46	16.41	0.25
			8%	7.5	28.17	0.004	7.53	10.55	14.83	17.77	23.09	0.599
	5.0	15%	2%	5.13	22.95	0.033	5.31	7.49	11.90	14.96	20.06	0.392
			5%	20.52	97.87	0	20.52	26.61	34.10	38.93	47.45	0.971
			8%	41.15	178.2	0	41.15	49.49	58.45	63.89	77.79	1
500	3.0	15%	2%	2.15	7.7	0.154	2.54	2.96	6.06	7.90	12.19	0.135
			5%	7.54	29.23	0.007	7.59	10.31	15.08	18.80	23.63	0.591
			8%	15.64	57.02	0	15.64	20.04	25.41	29.54	36.88	0.95
		30%	2%	1.37	4.03	0.211	1.74	1.83	3.85	5.51	9.03	0.057
			5%	4.81	16.66	0.03	4.96	6.72	10.82	12.79	17.21	0.371
			8%	9.96	40.06	0	9.96	13.57	18.45	21.34	28.29	0.748
	5.0	15%	2%	7.03	31.77	0.013	7.13	10.46	14.66	17.65	24.22	0.532
			5%	28.55	135.4	0	28.55	35.43	44.28	48.62	59.90	0.996
			8%	58	241	0	58	69.17	78.26	84.86	95.88	1
750	3.0	15%	2%	2.93	10.56	0.095	3.23	4.26	7.32	9.29	14.95	0.179
			5%	11.21	43.88	0	11.21	14.73	20.36	24.29	30.20	0.803
			8%	23.4	91.62	0	23.4	29.37	35.72	40.49	49.98	0.99
		30%	2%	1.79	5.45	0.166	2.15	2.59	4.84	6.49	10.67	0.081
			5%	6.89	24.43	0.012	6.97	9.77	14.06	16.70	21.09	0.558

			8%	14.43	54.81	0	14.43	18.59	24.79	28.26	36.17	0.919
	5.0	15%	2%	10.09	43.7	0.005	10.14	13.98	19.13	22.15	30.28	0.727
			5%	42.4	190.9	0	42.4	51.22	60.08	66.20	81.29	1
			8%	87.36	387.7	0	87.36	100.1	112.8	119.5	134.8	1
Sample size	Study Length	Censoring rate	LTS	avg	var	0's	Non 0 avg	75%	90%	95%	99%	Power
1000	3.0	15%	2%	3.44	12.3	0.053	3.63	5.02	8.24	10.68	15.43	0.229
			5%	14.68	55.23	0	14.68	19.07	24.60	28.36	35.15	0.92
			8%	30.46	110.24	0	30.46	36.87	44.89	49.89	57.88	1
		30%	2%	2.24	7327	0.124	2.56	3.31	5.87	7.58	11.41	0.124
			5%	8.8	36.67	0.004	8.83	12.60	17.37	20.91	25.45	0.658
			8%	19.66	75.8	0	19.66	25.07	31.45	34.97	43.76	0.966
	5.0	15%	2%	13.12	60.12	0.005	13.19	18.08	23.09	27.38	35.78	0.84
			5%	56.5	285.8	0	56.5	66.89	79.84	86.60	100.2	1
			8%	116	537.6	0	116	130.8	145.6	156.3	175.9	1
2000	3.0	15%	2%	5.97	22.18	0.012	6.05	8.46	12.37	14.79	20.71	0.475
			5%	27.89	102.78	0	27.89	34.02	41.55	46.86	54.9	0.999
			8%	61.4	244.2	0	61.4	71.53	82.25	87.75	100.3	1
		30%	2%	3.58	14.02	0.054	3.78	5.30	8.31	11.01	15.08	0.248
			5%	16.32	65	0	16.32	20.94	27.15	31.17	39.36	0.943
			8%	37.37	142.2	0	37.37	45.15	52.96	58.17	65.34	1
	5.0	15%	2%	25.4	121.5	0	25.4	32.40	40.20	45.59	52.91	0.989
			5%	113.1	556	0	113.1	127.4	143.2	153.8	170.5	1
			8%	231.6	979.3	0	231.6	251.3	272.2	283.4	310.7	1

*(level of significance 0.01)

I used a multiple regression model with each of the factors (i.e. *n*, *P*, *r*, *D*, and (LTS)), the two factor interactions *nP, nr, nD, n(LTS), Pr, PD, P(LTS), rc, r(LTS)* and *c(LTS,)* and all sub-hierarchical interactions. The dependent variable was the inverse normal cdf of the power of the LRTS test using level of significance 1%. I recorded all simulations with 100% power as $3.719 = \phi^{-1}(0.9999)$. The censoring pattern was not found to be significant in any of the interactions and was removed from the model (p-values ranging from 0.25 to 0.78). Since the fitted model had a non-significant constant value (t=0.32, p=0.75), the model was fit without a constant term. The fitted model is:

$$\Phi^{-1}(\hat{p}) = \frac{-36.2}{\sqrt{n}} - 3.93r + 2.16I_D + 79.19(LTS) + \frac{39.88r}{\sqrt{n}} - \frac{514.34(LTS)}{\sqrt{n}} - 3.08rI_D - 45.36r(LTS)$$

(t = -8.3) (t = -5.3) (t = 12.0) (t = 19.6) (t = 2.4) (t = -6.2) (t = -4.9) (t = -83.9)

The R^2 for the fitted model is 92% and shows a good fit. All variables were highly significant. The variable $\frac{r}{\sqrt{n}}$ had the largest p-value (0.0167). All other factors have p-values less than 0.001. There is increase in power with increased sample size and increased study duration. The power also increases as the proportion of LTS increases. The interactions of the censoring rate, sample size, study duration and proportion of LTS are significant. When LTS or study duration in the interaction are held constant, the interaction associations are that power decreases as the expected proportion censored increases. However, the cumulative effect of the interaction with LTS or Study duration is not large enough to negate the increase of power for increasing LTS and increased study length. For example, the proportion of LTS increases the estimated transformed power variable by a factor of 0.8 for every 1% increase of LTS. The interaction with censoring rate has a fitted decrease of a factor of 0.45. Even for censoring rates = 0.50 the interaction decrease is 0.225 (=0.45x0.50), so that an increase in LTS will result in a net gain in power.

3.25 –Model Limits:

The estimated power as a function of sample size is summarized by censoring rate and proportion of *LTS* in Figures 3.8 through 3.15. Each Figure is summarized over censoring rates that range from 10% to 50% and proportion of LTS that ranges from 2% to 8%. These are the range of the independent variables used when the model was selected.

When the proportion of LTS is small, the power is small, unless the sample size is large. In the event of many censored observations a sample size of many thousand would be required to reject the null hypothesis. In the case of long study durations, power near 50% can be achieved in some smaller sample size situations. Thus, if it is hypothesized that there is a very small proportion of LTS and an increase in sample size is not possible, the duration of the study can be extended to increase the probability of the model being detected.

Specifically, when 10% of observations are observed as censored in a longer study, there is an estimated power of at least 50% with fewer than 150 observations for models with $LTS \geq 0.04$, as shown in Figure 3.8. For models with $0.02 \leq LTS < 0.04$, 50% power is estimated with fewer than 300 observations. In addition, fewer than 200 observations are required for an estimated 80% power with $LTS \geq 0.05$.

In a shorter study duration with 10% censoring at least 50% estimated power is observed with fewer than 300 observations for models with $LTS \geq 0.05$, as shown in Figure 3.9. At least 80% estimated power is observed with fewer than 550 observations for models with $LTS \geq 0.05$.

As shown in Figures 3.10 through 3.13, as the proportion censored increases from 10%, the power drops rapidly for smaller sample sizes. However, in longer studies high power is still possible

In situations with censoring rates 50%, estimated power of at least 50% is still achievable in both long and short studies for samples of less than 1000, as shown in Figures 3.14 and 3.15. This level of power requires at least 5% LTS. Only small censoring rates will estimate power over 50% for $LTS < 0.05$.

Figure 3.8: Estimated Power by Sample Size
10% Expected Censoring, Longer Study

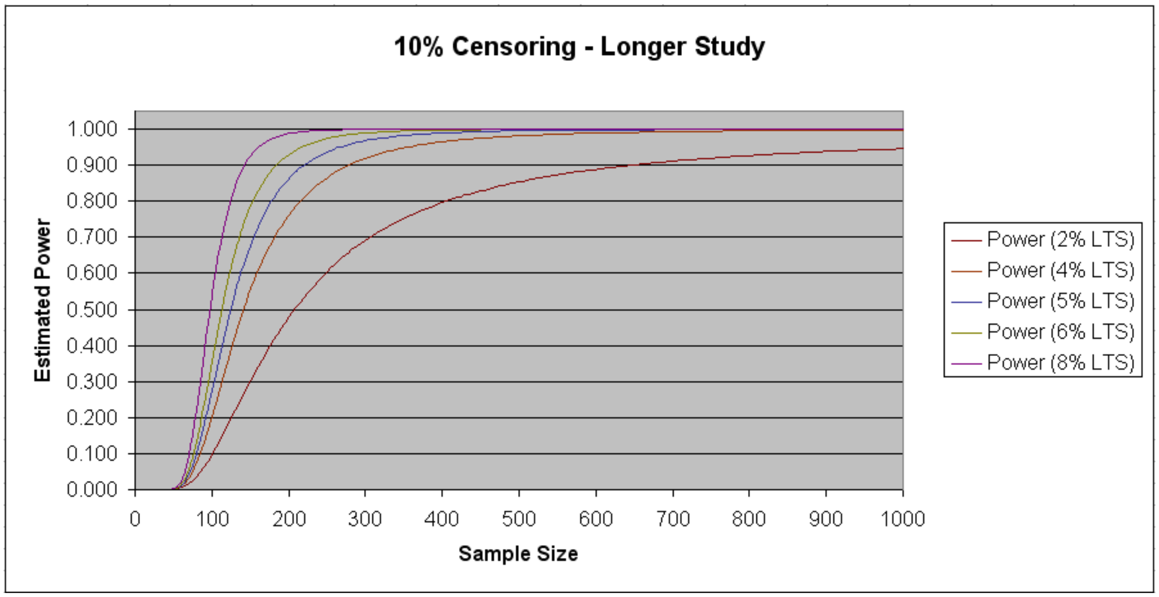


Figure 3.9: Estimated Power by Sample Size
10% Expected Censoring, Shorter Study

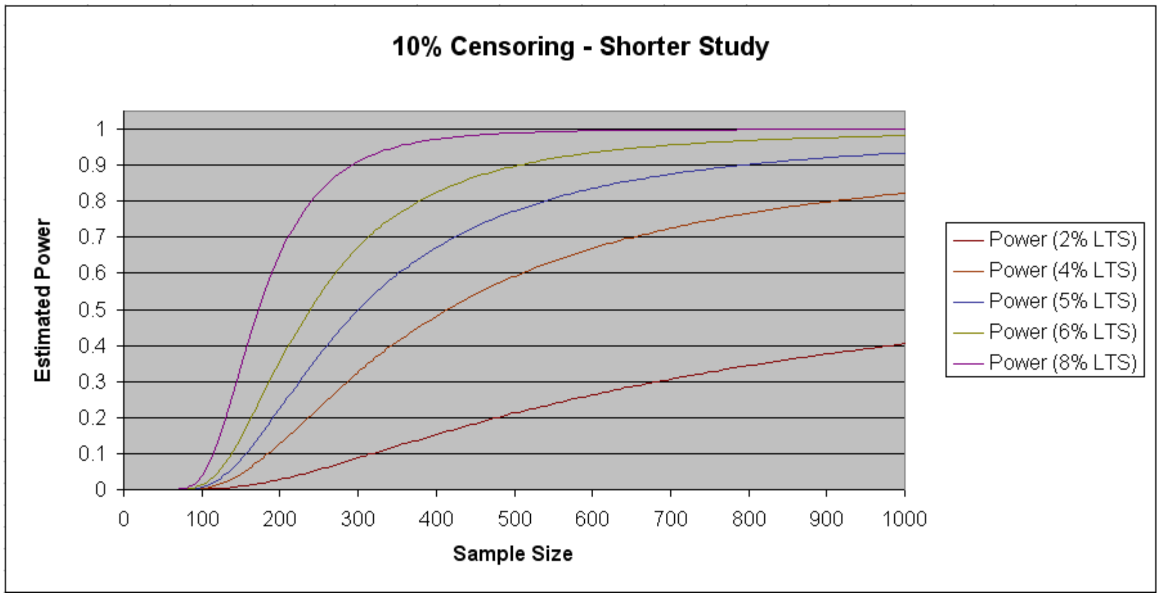


Figure 3.10: Estimated Power by Sample Size
20% Expected Censoring, Longer Study

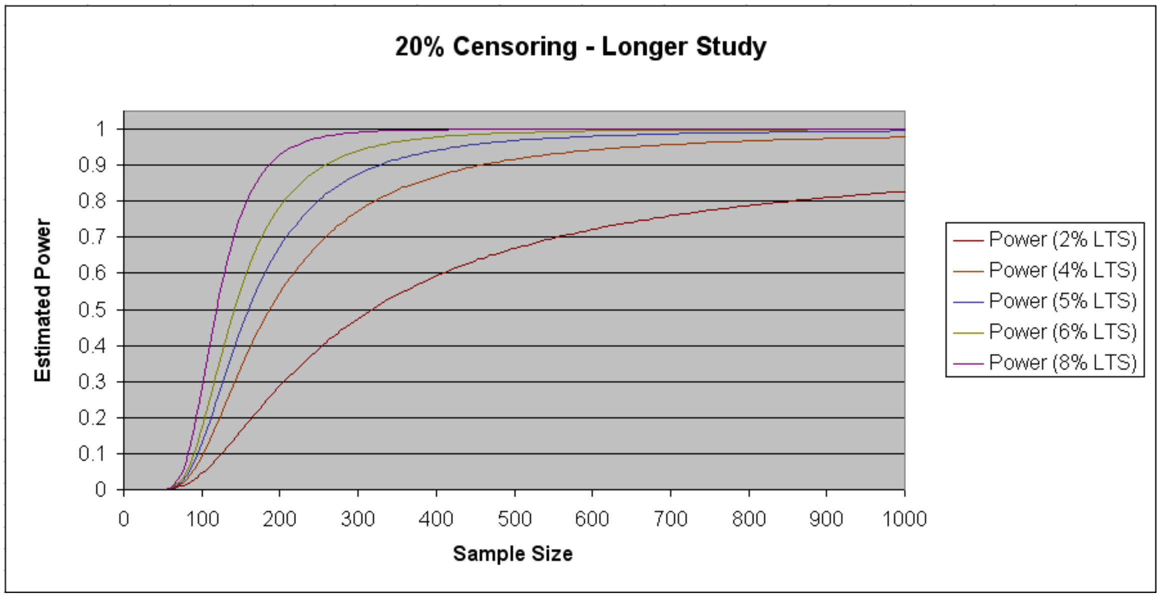


Figure 3.11: Estimated Power by Sample Size
20% Expected Censoring, Shorter Study

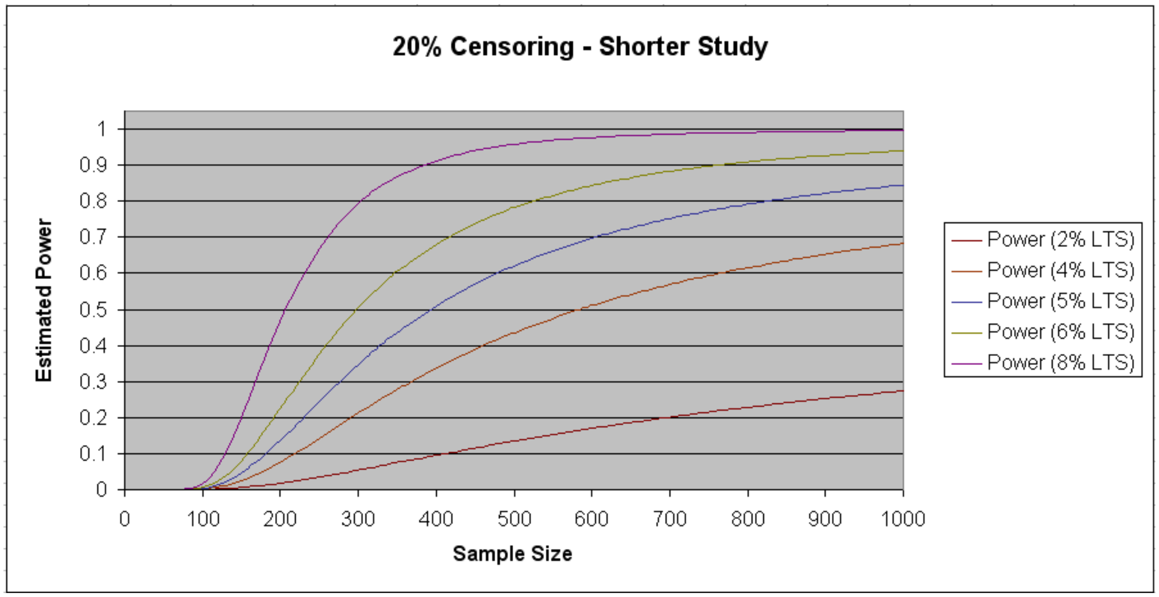


Figure 3.12: Estimated Power by Sample Size
30% Expected Censoring, Longer Study

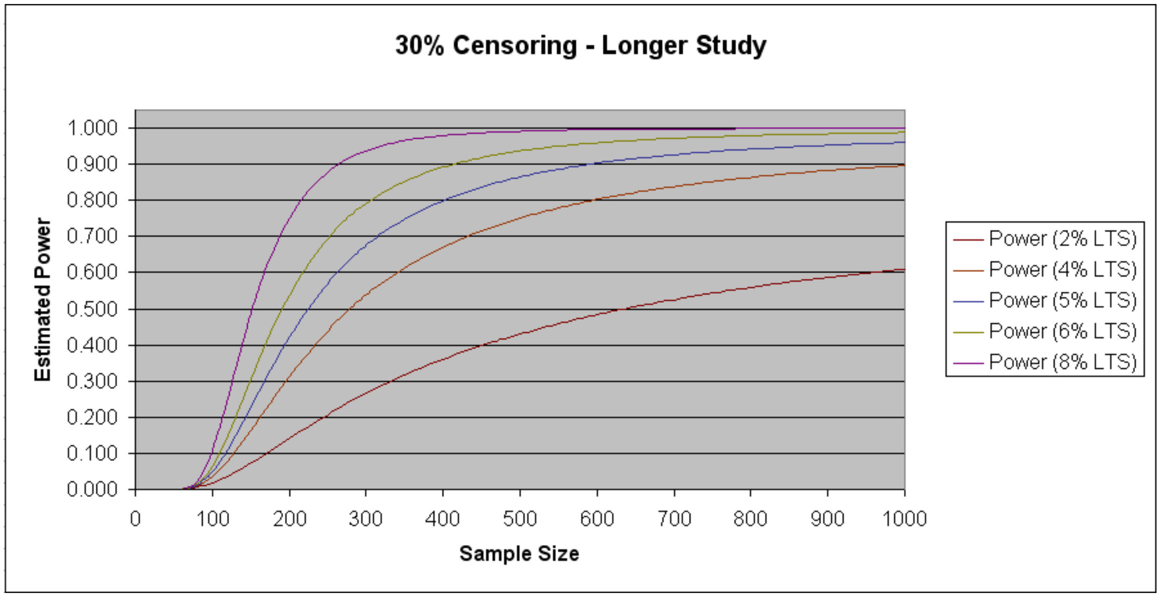


Figure 3.13: Estimated Power by Sample Size
30% Expected Censoring, Shorter Study

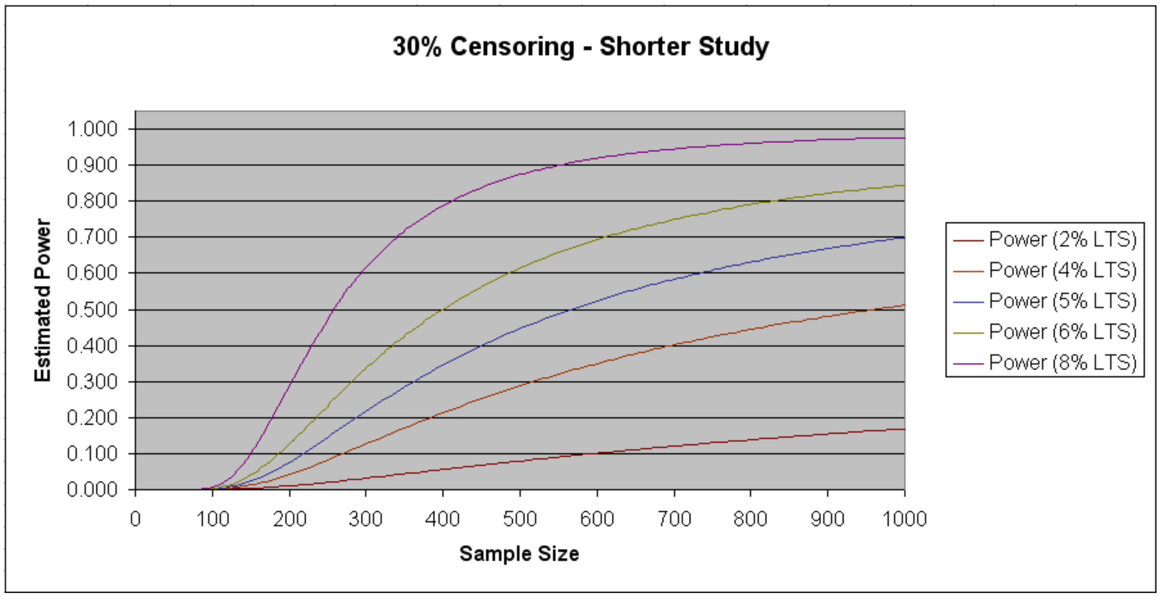


Figure 3.14: Estimated Power by Sample Size
50% Expected Censoring, Longer Study

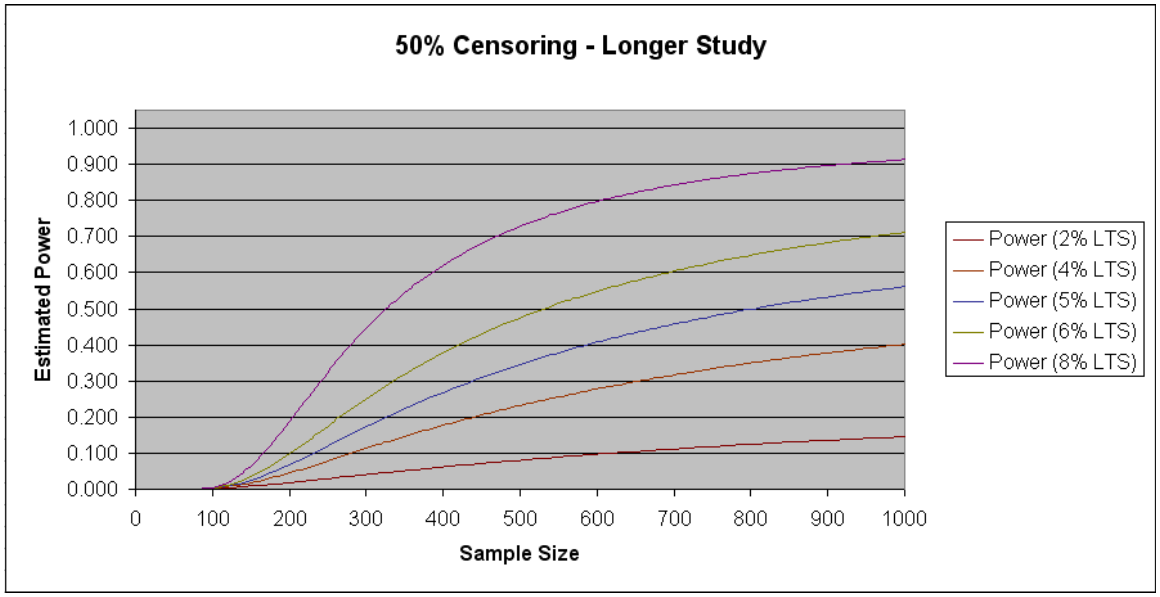


Figure 3.15: Estimated Power by Sample Size
50% Expected Censoring, Shorter Study

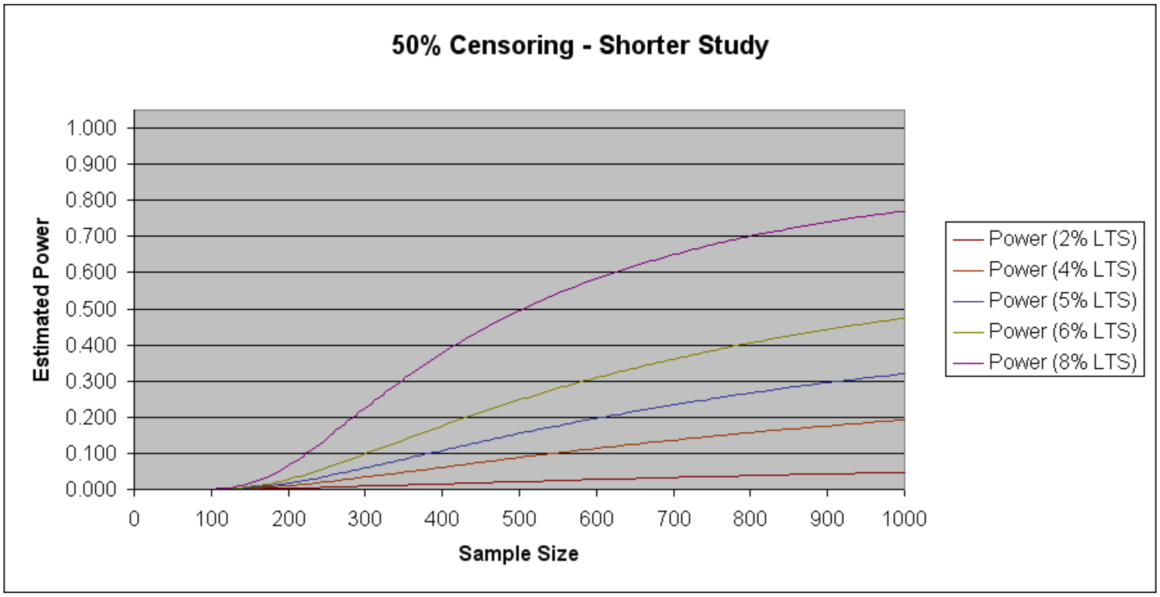


Table 3.10 provides the minimum sample size required for an estimated 50% and 80% power. It is clear that for $LTS = 0.02$ one must run a long study to be able to have large power. However, even in long studies if the proportion censored is large it will take a large sample size to obtain power over 50%. In the event that a study is short, there will be estimated power of over 50% for large sample sizes with proportion of LTS at least 5%.

Table 3.10: Summary of Estimated Sample Size Required for 50% and 80% Power

Censoring Rate	Study Duration	LTS	Sample Size Required for 50% Estimated Power	Sample Size Required for 80% Estimated Power
10%	Longer	2%	210	410
		5%	125	180
		8%	100	125
	Shorter	2%	1495	**
		5%	305	540
		8%	175	240
20%	Longer	2%	320	855
		5%	165	250
		8%	120	160
	Shorter	2%	3950	**
		5%	395	825
		8%	210	305
30%	Longer	2%	640	4400
		5%	225	405
		8%	155	215

	Shorter	2%	**	**
		5%	570	1580
		8%	260	415
40%	Longer	2%	2800	**
		5%	365	865
		8%	210	330
	Shorter	2%	**	**
		5%	965	**
		8%	345	620
50%	Longer	2%	**	**
		5%	805	4400
		8%	330	605
	Shorter	2%	**	**
		5%	2400	**
		8%	505	1125

** indicates a sample over 5000 is required.

When observing 250 or fewer observations, my estimated power function shows that an estimated power of at least 50% is predicted for samples sizes as small as 100 for certain situations.

In Figures 3.16 through 3.19, the power function is modeled as a function of censoring rate ($r \geq 0.08$) for a longer study. For these smaller sample sizes, when the proportion of censored responses exceeds 20%, the estimated power is generally under 50%. However, in a clinical trial with under 250 participants, if over 80% of the participants complete the study, the LTS model can yield high power. It is clear in comparing Figures 3.16 and 3.17 and an increase in sample size from 100 to 150 has a large increase in power. While a sample size of 100 may not be practical, a sample of 150 is.

Figure 3.16: Estimated Power by Censoring Rate
Sample Size 100, Longer Study

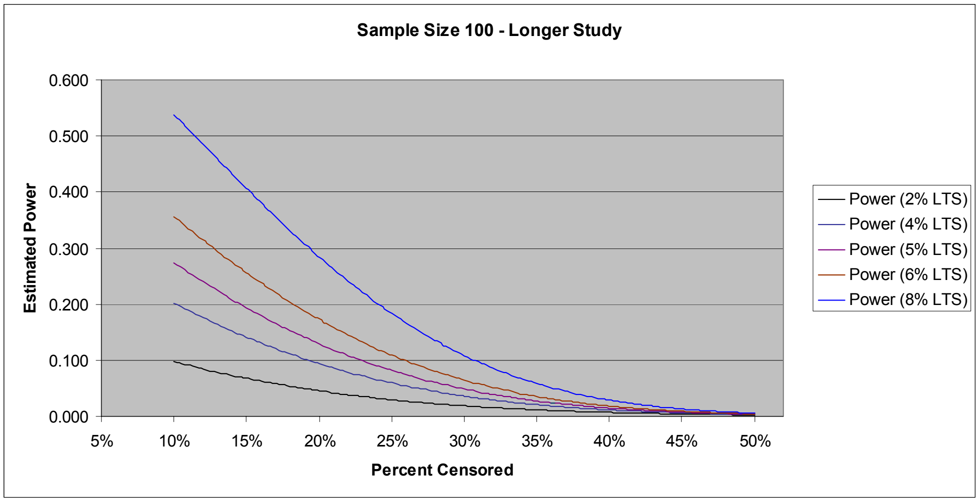


Figure 3.17: Estimated Power by Censoring Rate
Sample Size 150, Longer Study

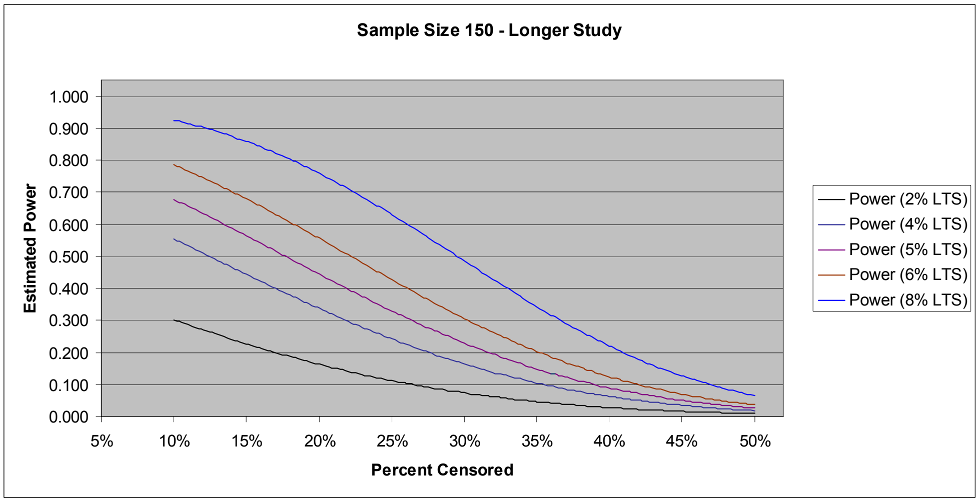


Figure 3.18: Estimated Power by Censoring Rate
Sample Size 200, Longer Study

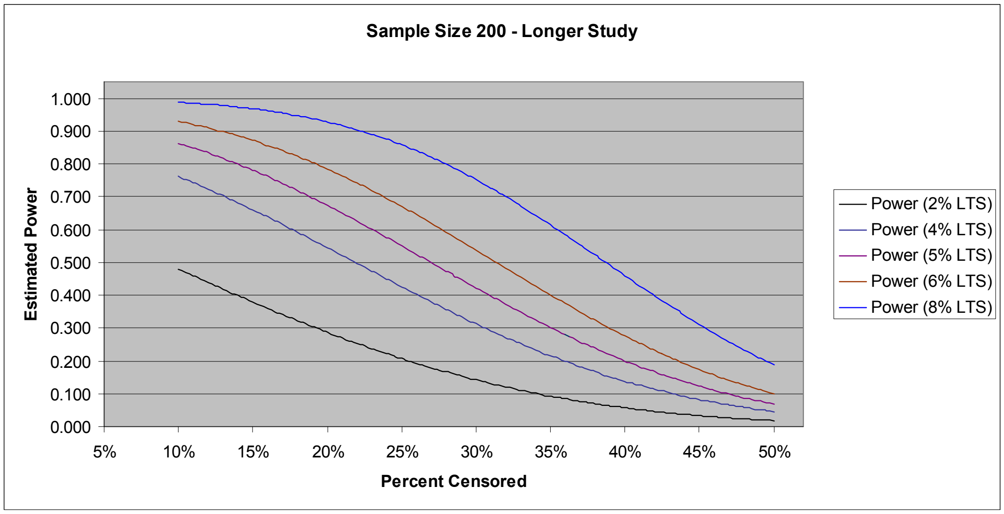
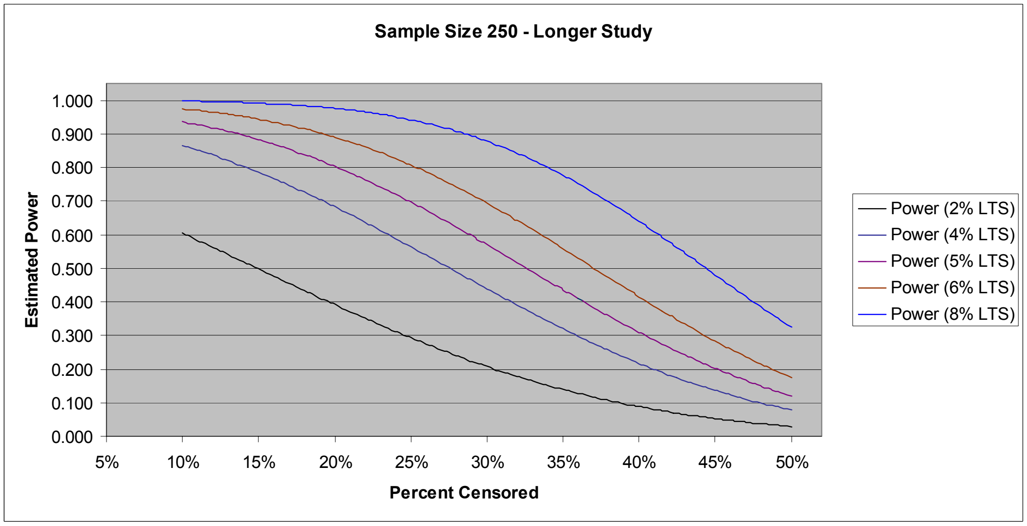


Figure 3.19: Estimated Power by Censoring Rate
Sample Size 250, Longer Study



In a shorter study, only models with at least 5% LTS had estimated power above 25% in smaller sample sizes. Once the sample size exceeded 200, the estimated power consistently was estimated at over 50% for LTS of at least 5%.

For sample sizes under 150, the shorter study duration did not yield an estimated power of over 20% and is not summarized. This is summarized in Figures 3.20 through 3.22.

Figure 3.20: Estimated Power by Censoring Rate
Sample Size 150, Shorter Study

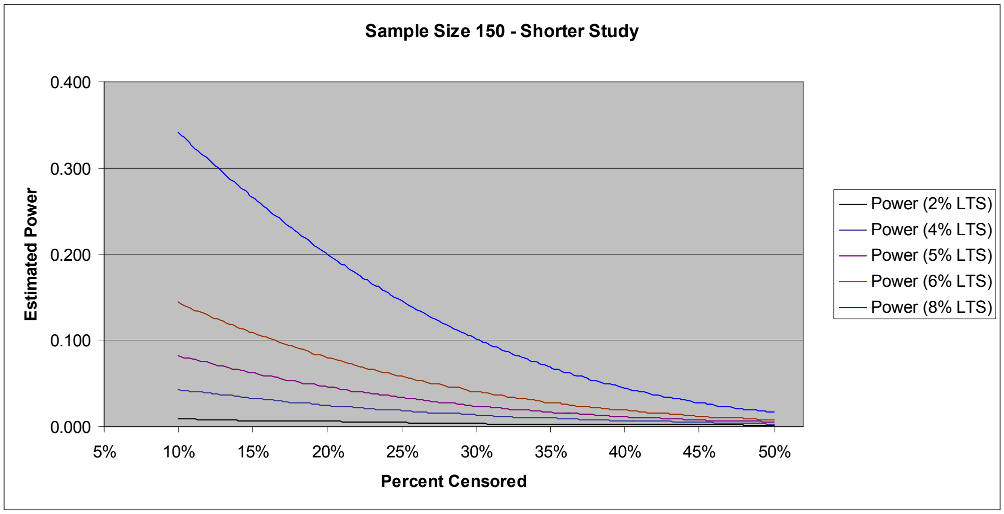


Figure 3.21: Estimated Power by Censoring Rate
Sample Size 200, Shorter Study

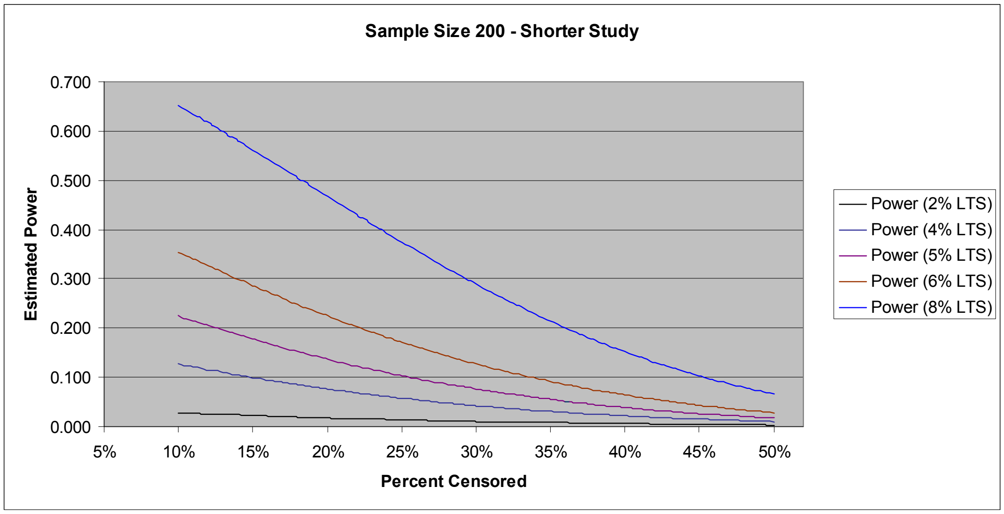
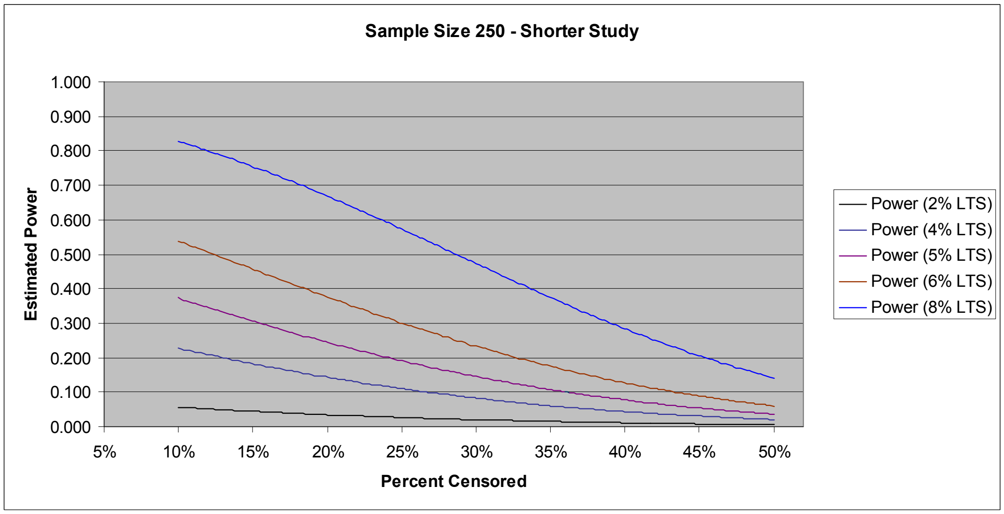


Figure 3.22: Estimated Power by Censoring Rate
Sample Size 250, Shorter Study



It is clear from these Figures that as the sample sizes increases, the power also increases, but for relatively smaller sample sizes one can find moderate power in models with a proportion of LTS above 5%.

3.3 – Mixture of Two Exponentials:

3.31 –Estimated Null Percentiles:

In preparation for studying the power of the mixture of two exponential models (L_M), I decided to expand the simulation study in Ye (2006). Liu et al. (2004), proved that in a two-component normal mixture model, the LRTS statistic $2\lambda_n$ diverges at a rate of loglog(*n*). In a related paper the asymptotic behavior of the LRTS for homogeneity against a mixture of gammas, is also shown to be divergent at the rate loglog(*n*) (Liu et al. 2003). I used the log-log *n* transformation in the regression analysis below. In my simulation I added sample sizes of 1000 and 2000. Table 3.11 summarizes the results of the null distribution of the LRTS. The fraction of zero LRTS, $\hat{\pi}_1$, is between 0.14 and 0.32 with average fraction of zero LRTS equal to 0.22. It decreases on average as *n* increases. The regression function $\Phi[(0.385 \pm 0.09) - (0.641 \pm 0.052)\log(\log(n))]$ explains 88.6% of the variation in the fraction of zero LRTS ($p < 8 \times 10^{-12}$, the standard error of regression coefficient is given after $\pm$). The fraction of zero LRTS was not sensitive to the censoring rate (p>0.98) or censoring pattern (p>0.11).

Table 3.11: Mean, Variance and Simulated Percentiles of the null distribution of LRT. (L_M)

Sample size	Censoring Pattern	Censoring rate	Mean	Variance	$\hat{\pi}_1$: Fraction of zero LRTS	Mean of Non-Zero LRTS	Percentile			
							75%	90%	95%	99%
50	Exponential	10%	1.18	3.43	0.32	1.74	1.68	3.58	4.87	8.17
		30%	1.16	3.43	0.29	1.63	1.61	3.42	4.99	7.82
	Uniform	10%	1.21	3.52	0.30	1.71	1.65	3.56	5.06	8.08
		30%	1.28	3.31	0.26	1.74	1.96	3.59	4.98	8.01
100	Exponential	10%	1.25	3.64	0.28	1.73	1.73	3.69	5.44	8.24
		30%	1.37	3.81	0.26	1.44	1.94	3.87	5.17	8.82
	Uniform	10%	1.41	3.79	0.24	1.87	2.11	4.10	5.51	8.57
		30%	1.23	3.07	0.25	1.65	1.82	3.45	4.41	8.44
200	Exponential	10%	1.43	3.64	0.23	1.84	2.11	4.03	5.26	7.35
		30%	1.38	4.23	0.25	1.78	1.92	4.02	5.57	7.50
	Uniform	10%	1.30	3.50	0.22	1.67	1.79	3.81	4.98	8.34

		30%	1.34	3.30	0.22	1.73	1.97	4.00	5.18	7.93
500	Exponential	10%	1.41	3.83	0.18	1.71	1.96	3.95	5.05	9.57
		30%	1.60	4.44	0.18	1.94	2.37	4.41	5.45	10.70
	Uniform	10%	1.61	3.20	0.17	1.93	2.44	4.53	5.59	8.08
		30%	1.39	3.43	0.20	1.73	2.02	4.09	5.68	7.98
1000	Exponential	10%	1.51	4.02	0.19	1.86	2.23	3.91	5.75	8.87
		30%	1.47	3.78	0.21	1.87	2.17	4.27	5.57	8.30
	Uniform	10%	1.64	4.23	0.17	1.97	2.32	4.58	5.91	9.44
		30%	1.59	4.85	0.19	1.97	2.15	4.36	6.18	9.98
2000	Exponential	10%	1.63	4.64	0.16	1.94	2.32	4.37	5.77	10.65
		30%	1.62	3.75	0.14	1.90	2.37	4.09	5.68	9.09
	Uniform	10%	1.68	3.86	0.16	2.00	2.56	4.28	5.75	8.30
		30%	1.57	4.20	0.16	1.87	2.25	4.22	5.37	9.13

Based on 1000 replications for each setting.

The estimated “degrees of freedom,” $\hat{\nu}_1$ (that is, the mean of the non-zero LRTS values), is between 1.44 and 2.00 with an average of 1.80. It also increases on average as n increases. The regression function $(1.11 \pm 0.15) + (0.40 \pm 0.09)\log(\log(n))$ explains 49% of the variation in the mean of the non-zero LRTS ($p < 0.0002$, standard error of regression coefficient given after $\pm$). The mean non-zero LRTS was not sensitive to the censoring rate (p>0.13) or censoring pattern (p>0.35).

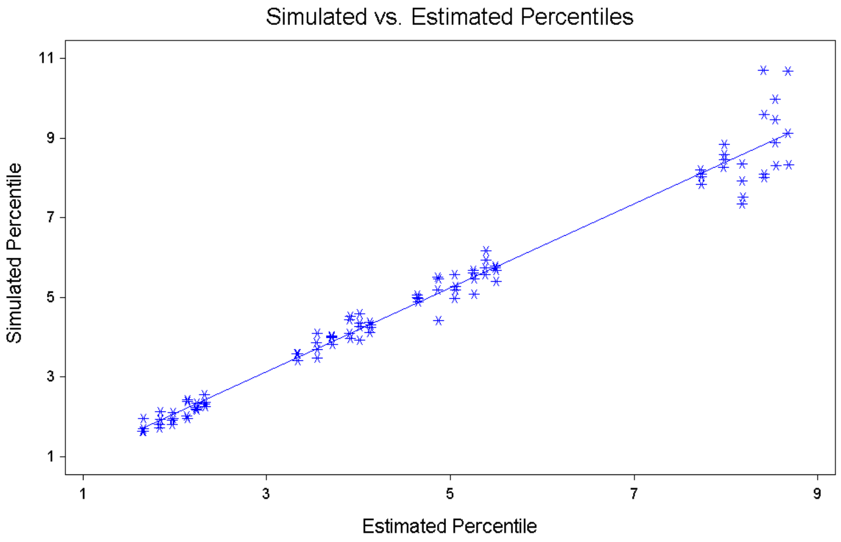
I used these functions to estimate the percentiles of the null distribution following the null distribution considered by Ye (2006). Following Ye (2006) the null distribution of the LRTS used to test the mixture of two exponential components without LTS is $\pi_1\chi_0^2 + (1-\pi_1)\chi_{\nu_1}^2$, where π_1 is the fraction of zero LRTS and $\chi_{\nu_1}^2$ is a chi-square distribution with degrees of freedom ν_1. Table 3.12 contains the estimated 75th, 90th, 95th, 99th, and 99.9th percentiles for the sample sizes used in the simulation study. I ran a bivariate regression to test for correlation between the expected and observed percentile values. Figure 3.23 is a plot of the simulated percentile against the estimated percentile for the 75th, 90th, 95th, and 99th percentiles. The correlation between estimated percentile and simulated percentile is 0.985. That is, estimated percentiles explain 97.1% of the variance of the simulated percentiles in Table 3.12.

Table 3.12. Estimated Null Distribution Percentiles of LRTS

L_M vs L_0

Sample Size	**75%**	**90%**	**95%**	**99%**	**99.9%**
50	1.66	3.34	4.65	7.73	12.20
100	1.84	3.55	4.87	7.98	12.48
200	1.98	3.72	5.06	8.19	12.71
350	2.08	3.84	5.18	8.33	12.87
500	2.14	3.91	5.26	8.42	12.96
750	2.20	3.98	5.33	8.50	13.06
1000	2.24	4.02	5.39	8.55	13.19
2000	2.33	4.13	5.50	8.69	13.27

Figure 3.23: Observed Simulated LRTS Values vs Estimated LRTS Values for Summarized Percentiles (L_M).



Note: The four clusters are the 75th, 90th, 95th and 99th percentiles.

I used the estimated 99th percentiles to calculate the power of the mixture alternative.

3.32 – Power:

To study power I used a simulation study with 192 settings. I considered expected censoring rate (10% or 30%), censoring pattern (exponential or uniform), mixing proportion (50%, 65%, 75% or 85%), difference of expected event times (0.5, 1.0 or 1.5) and sample size (200, 350, 500 or 750). For each setting I ran 500 replicates. I calculated the average LRTS, the fraction of LRTS values with result nearly equal to zero, where nearly zero is defined as an LRTS less than 0.0001, the average of the non-zero LRTS values and the power using an alpha level of 1% with the null percentiles in Table 3.12. Table 3.13 contains the simulated power of this LRTS and other summary statistics. The 50-50 mixture with greatest difference of component means (difference equals 1.5) has power near 1 for both censoring patterns and censoring rates, even for sample size 200. For skewed mixing proportions (that is, m =0.85) with greatest difference of component means, the power increases with increasing sample size, as expected. For smaller difference of component means (difference equals 0.5) for both symmetric (m =0.50) and skewed mixtures, the power is low for both censoring patterns and both censoring rates.

Table 3.13: Simulated Power and Summary Statistics of the LRTS of L_M vs L_0

Censoring Pattern	Sample size	Mixing proportion	Difference of means	Censoring Rate	Average LRTS	Fraction of zero LRTS	Mean of non-zero LRTS	Power
P	S	M	D	R	Y_1	Y_3	Y_4	Y_2
1	200	0.50	1.5	0.10	48.40	0.00	48.40	1.00
1	200	0.50	1.5	0.30	36.43	0.00	36.43	1.00*
1	350	0.50	1.5	0.10	81.87	0.00	81.87	1.00
1	350	0.50	1.5	0.30	61.58	0.00	61.71	1.00*
1	500	0.50	1.5	0.10	118.44	0.00	118.44	1.00
1	500	0.50	1.5	0.30	88.38	0.00	88.38	1.00*
1	750	0.50	1.5	0.10	178.21	0.00	178.21	1.00
1	750	0.50	1.5	0.30	133.27	0.00	133.27	1.00
2	200	0.50	1.5	0.10	47.79	0.00	47.79	1.00
2	200	0.50	1.5	0.30	32.04	0.00	32.04	1.00*
2	350	0.50	1.5	0.10	81.89	0.00	81.89	1.00
2	350	0.50	1.5	0.30	55.19	0.00	55.3	1.00*
2	500	0.50	1.5	0.10	118.15	0.00	118.15	1.00
2	500	0.50	1.5	0.30	77.90	0.00	77.90	1.00*
2	750	0.50	1.5	0.10	174.88	0.00	174.88	1.00
2	750	0.50	1.5	0.30	115.56	0.00	115.56	1.00
1	200	0.65	1.5	0.10	23.25	0.00	23.25	0.96
1	200	0.65	1.5	0.30	14.71	0.00	14.71	0.74
1	350	0.65	1.5	0.10	39.59	0.00	39.59	1.00
1	350	0.65	1.5	0.30	24.05	0.00	24.05	0.96
1	500	0.65	1.5	0.10	56.24	0.00	56.24	1.00
1	500	0.65	1.5	0.30	33.2	0.00	33.2	0.99
1	750	0.65	1.5	0.10	85.70	0.00	85.70	1.00
1	750	0.65	1.5	0.30	49.52	0.00	49.52	1.00
2	200	0.65	1.5	0.10	21.93	0.00	21.93	0.94
2	200	0.65	1.5	0.30	10.81	0.00*	10.86	0.62
2	350	0.65	1.5	0.10	37.76	0.00	37.76	1.00
2	350	0.65	1.5	0.30	17.79	0.00	17.79	0.84
2	500	0.65	1.5	0.10	53.83	0.00	53.83	1.00
2	500	0.65	1.5	0.30	23.85	0.00	23.85	0.96
2	750	0.65	1.5	0.10	81.23	0.00	81.23	1.00
2	750	0.65	1.5	0.30	35.51	0.00	35.51	1.00

Censoring Pattern	Sample size	Mixing proportion	Difference of means	Censoring Rate	Average LRTS	Fraction of zero LRTS	Mean of non-zero LRTS	Power
1	200	0.75	1.5	0.10	13.56	0.00*	13.58	0.72
1	200	0.75	1.5	0.30	7.16	0.01	7.24	0.35
1	350	0.75	1.5	0.10	22.67	0.00	22.67	0.91
1	350	0.75	1.5	0.30	11.82	0.00*	11.87	0.62
1	500	0.75	1.5	0.10	31.54	0.00	31.54	0.99
1	500	0.75	1.5	0.30	16.20	0.00	16.20	0.79
1	750	0.75	1.5	0.10	48.34	0.00	48.34	1.00
1	750	0.75	1.5	0.30	24.46	0.00	24.46	0.96
2	200	0.75	1.5	0.10	12.95	0.00*	12.98	0.68
2	200	0.75	1.5	0.30	5.17	0.02	5.29	0.20
2	350	0.75	1.5	0.10	21.07	0.00	21.07	0.91
2	350	0.75	1.5	0.30	7.36	0.00	7.36	0.36
2	500	0.75	1.5	0.10	30.32	0.00	30.32	0.98
2	500	0.75	1.5	0.30	10.02	0.00	10.02	0.55
2	750	0.75	1.5	0.10	44.46	0.00	44.46	1.00
2	750	0.75	1.5	0.30	14.17	0.00	14.17	0.75
1	200	0.85	1.5	0.10	7.07	0.02	7.25	0.35
1	200	0.85	1.5	0.30	4.27	0.04	4.44	0.15
1	350	0.85	1.5	0.10	11.26	0.00*	11.30	0.58
1	350	0.85	1.5	0.30	5.41	0.03	5.59	0.22
1	500	0.85	1.5	0.10	15.04	0.00	15.04	0.72
1	500	0.85	1.5	0.30	7.59	0.01	7.63	0.38
1	750	0.85	1.5	0.10	22.30	0.00	22.30	0.89
1	750	0.85	1.5	0.30	10.36	0.00*	10.38	0.54
2	200	0.85	1.5	0.10	6.50	0.02	6.60	0.31
2	200	0.85	1.5	0.30	2.47	0.09	2.72	0.04
2	350	0.85	1.5	0.10	9.75	0.00	9.75	0.51
2	350	0.85	1.5	0.30	3.16	0.04	3.28	0.07
2	500	0.85	1.5	0.10	12.98	0.00	12.98	0.69
2	500	0.85	1.5	0.30	3.98	0.03	4.10	0.11
2	750	0.85	1.5	0.10	19.88	0.00	19.88	0.86
2	750	0.85	1.5	0.30	4.98	0.01	5.04	0.18

Censoring Pattern	Sample size	Mixing proportion	Difference of means	Censoring Rate	Average LRTS	Fraction of zero LRTS	Mean of non-zero LRTS	Power
1	200	0.50	1.0	0.10	9.14	0.00*	9.16	0.51
1	200	0.50	1.0	0.30	6.95	0.00*	6.96	0.33
1	350	0.50	1.0	0.10	15.86	0.00	15.86	0.81
1	350	0.50	1.0	0.30	10.63	0.00*	10.65	0.62
1	500	0.50	1.0	0.10	22.07	0.00	22.07	0.94
1	500	0.50	1.0	0.30	14.52	0.00	14.52	0.76
1	750	0.50	1.0	0.10	31.81	0.00	31.81	0.99
1	750	0.50	1.0	0.30	20.64	0.00	20.64	0.92
2	200	0.50	1.0	0.10	9.25	0.00*	9.27	0.47
2	200	0.50	1.0	0.30	5.41	0.01	5.46	0.21
2	350	0.50	1.0	0.10	14.91	0.00	14.91	0.78
2	350	0.50	1.0	0.30	8.61	0.00	8.61	0.47
2	500	0.50	1.0	0.10	21.68	0.00	21.68	0.93
2	500	0.50	1.0	0.30	11.23	0.00	11.23	0.64
2	750	0.50	1.0	0.10	31.12	0.00	31.12	0.99
2	750	0.50	1.0	0.30	15.83	0.00	15.83	0.84
1	200	0.65	1.0	0.10	6.81	0.01	6.88	0.32
1	200	0.65	1.0	0.30	4.65	0.02	4.77	0.15
1	350	0.65	1.0	0.10	10.78	0.00*	10.80	0.56
1	350	0.65	1.0	0.30	6.70	0.00*	6.72	0.31
1	500	0.65	1.0	0.10	15.09	0.00*	15.12	0.77
1	500	0.65	1.0	0.30	8.77	0.00*	8.78	0.44
1	750	0.65	1.0	0.10	21.19	0.00	21.19	0.92
1	750	0.65	1.0	0.30	12.17	0.00	12.17	0.63
2	200	0.65	1.0	0.10	6.96	0.02	7.11	0.33
2	200	0.65	1.0	0.30	3.63	0.04	3.79	0.10
2	350	0.65	1.0	0.10	10.44	0.01	10.50	0.56
2	350	0.65	1.0	0.30	5.05	0.02	5.16	0.18
2	500	0.65	1.0	0.10	13.73	0.00	13.73	0.74
2	500	0.65	1.0	0.30	6.17	0.01	6.25	0.26
2	750	0.65	1.0	0.10	19.74	0.00	19.74	0.90
2	750	0.65	1.0	0.30	7.64	0.00*	7.66	0.34

Censoring Pattern	Sample size	Mixing proportion	Difference of means	Censoring Rate	Average LRTS	Fraction of zero LRTS	Mean of non-zero LRTS	Power
1	200	0.75	1.0	0.10	4.77	0.03	4.91	0.20
1	200	0.75	1.0	0.30	3.57	0.05	3.76	0.10
1	350	0.75	1.0	0.10	7.46	0.01	7.55	0.36
1	350	0.75	1.0	0.30	4.74	0.02	4.84	0.18
1	500	0.75	1.0	0.10	10.05	0.01	10.14	0.49
1	500	0.75	1.0	0.30	5.83	0.01	5.87	0.25
1	750	0.75	1.0	0.10	13.59	0.00	13.59	0.69
1	750	0.75	1.0	0.30	7.71	0.00*	7.72	0.37
2	200	0.75	1.0	0.10	4.77	0.02	4.88	0.17
2	200	0.75	1.0	0.30	2.64	0.08	2.87	0.05
2	350	0.75	1.0	0.10	7.17	0.02	7.28	0.35
2	350	0.75	1.0	0.30	3.36	0.04	3.51	0.08
2	500	0.75	1.0	0.10	9.55	0.00	9.55	0.49
2	500	0.75	1.0	0.30	3.53	0.03	3.64	0.10
2	750	0.75	1.0	0.10	13.41	0.00	13.41	0.70
2	750	0.75	1.0	0.30	5.03	0.01	5.08	0.17
1	200	0.85	1.0	0.10	3.03	0.06	3.23	0.06
1	200	0.85	1.0	0.30	2.49	0.10	2.77	0.04
1	350	0.85	1.0	0.10	4.46	0.03	4.61	0.16
1	350	0.85	1.0	0.30	2.88	0.07	3.09	0.06
1	500	0.85	1.0	0.10	5.63	0.02	5.76	0.24
1	500	0.85	1.0	0.30	3.59	0.03	3.70	0.10
1	750	0.85	1.0	0.10	7.54	0.00*	7.57	0.33
1	750	0.85	1.0	0.30	4.63	0.03	4.78	0.16
2	200	0.85	1.0	0.10	3.52	0.05	3.72	0.10
2	200	0.85	1.0	0.30	1.95	0.12	2.21	0.02
2	350	0.85	1.0	0.10	4.29	0.03	4.42	0.13
2	350	0.85	1.0	0.30	2.17	0.06	2.32	0.02
2	500	0.85	1.0	0.10	5.03	0.02	5.14	0.21
2	500	0.85	1.0	0.30	2.45	0.07	2.67	0.04
2	750	0.85	1.0	0.10	7.13	0.01	7.19	0.34
2	750	0.85	1.0	0.30	2.78	0.05	2.91	0.04

Censoring Pattern	Sample size	Mixing proportion	Difference of means	Censoring Rate	Average LRTS	Fraction of zero LRTS	Mean of non-zero LRTS	Power
1	200	0.50	0.5	0.10	2.33	0.10	2.60	0.04
1	200	0.50	0.5	0.30	2.04	0.14	2.37	0.03
1	350	0.50	0.5	0.10	2.75	0.07	2.94	0.04
1	350	0.50	0.5	0.30	2.70	0.09	2.98	0.04
1	500	0.50	0.5	0.10	3.44	0.04	3.59	0.06
1	500	0.50	0.5	0.30	2.59	0.06	2.75	0.03
1	750	0.50	0.5	0.10	4.16	0.02	4.26	0.07
1	750	0.50	0.5	0.30	3.16	0.04	3.30	0.13
2	200	0.50	0.5	0.10	2.43	0.11	2.73	0.05
2	200	0.50	0.5	0.30	1.95	0.12	2.21	0.01
2	350	0.50	0.5	0.10	3.02	0.07	3.23	0.07
2	350	0.50	0.5	0.30	2.11	0.12	2.41	0.03
2	500	0.50	0.5	0.10	3.34	0.04	3.49	0.08
2	500	0.50	0.5	0.30	2.37	0.08	2.57	0.04
2	750	0.50	0.5	0.10	4.41	0.02	4.49	0.14
2	750	0.50	0.5	0.30	3.91	0.03	4.04	0.11
1	200	0.65	0.5	0.10	2.02	0.15	2.39	0.03
1	200	0.65	0.5	0.30	1.78	0.16	2.12	0.03
1	350	0.65	0.5	0.10	2.56	0.08	2.77	0.05
1	350	0.65	0.5	0.30	2.28	0.10	2.54	0.03
1	500	0.65	0.5	0.10	3.06	0.04	3.18	0.05
1	500	0.65	0.5	0.30	2.47	0.07	2.67	0.04
1	750	0.65	0.5	0.10	3.80	0.05	4.00	0.10
1	750	0.65	0.5	0.30	3.01	0.07	3.24	0.07
2	200	0.65	0.5	0.10	2.25	0.13	2.59	0.04
2	200	0.65	0.5	0.30	1.74	0.13	2.00	0.02
2	350	0.65	0.5	0.10	2.82	0.06	3.02	0.04
2	350	0.65	0.5	0.30	1.91	0.12	2.18	0.02
2	500	0.65	0.5	0.10	2.84	0.09	3.11	0.04
2	500	0.65	0.5	0.30	2.09	0.09	2.31	0.02
2	750	0.65	0.5	0.10	3.72	0.04	3.90	0.08
2	750	0.65	0.5	0.30	2.24	0.08	2.43	0.03

Censoring Pattern	Sample size	Mixing proportion	Difference of means	Censoring Rate	Average LRTS	Fraction of zero LRTS	Mean of non-zero LRTS	Power
1	200	0.75	0.5	0.10	2.01	0.12	2.29	0.02
1	200	0.75	0.5	0.30	1.82	0.12	2.07	0.03
1	350	0.75	0.5	0.10	2.44	0.09	2.70	0.04
1	350	0.75	0.5	0.30	2.03	0.13	2.34	0.03
1	500	0.75	0.5	0.10	2.57	0.09	2.83	0.04
1	500	0.75	0.5	0.30	2.25	0.08	2.46	0.03
1	750	0.75	0.5	0.10	3.07	0.05	3.24	0.06
1	750	0.75	0.5	0.30	2.45	0.06	2.61	0.03
2	200	0.75	0.5	0.10	2.16	0.17	2.60	0.04
2	200	0.75	0.5	0.30	1.73	0.16	2.07	0.02
2	350	0.75	0.5	0.10	2.04	0.10	2.28	0.03
2	350	0.75	0.5	0.30	1.74	0.11	1.95	0.01
2	500	0.75	0.5	0.10	2.66	0.06	2.85	0.04
2	500	0.75	0.5	0.30	2.03	0.12	2.32	0.02
2	750	0.75	0.5	0.10	2.89	0.06	3.09	0.06
2	750	0.75	0.5	0.30	2.13	0.10	2.37	0.03
1	200	0.85	0.5	0.10	1.77	0.18	2.16	0.01
1	200	0.85	0.5	0.30	1.54	0.18	1.89	0.01
1	350	0.85	0.5	0.10	2.16	0.12	2.46	0.02
1	350	0.85	0.5	0.30	1.69	0.17	2.03	0.02
1	500	0.85	0.5	0.10	2.12	0.12	2.42	0.02
1	500	0.85	0.5	0.30	1.88	0.12	2.14	0.02
1	750	0.85	0.5	0.10	2.44	0.07	2.64	0.03
1	750	0.85	0.5	0.30	2.03	0.12	2.31	0.03
2	200	0.85	0.5	0.10	1.69	0.18	2.07	0.03
2	200	0.85	0.5	0.30	1.38	0.15	1.63	0.01
2	350	0.85	0.5	0.10	2.05	0.14	2.38	0.03
2	350	0.85	0.5	0.30	1.71	0.12	1.96	0.01
2	500	0.85	0.5	0.10	2.07	0.14	2.42	0.02
2	500	0.85	0.5	0.30	1.74	0.14	2.02	0.02
2	750	0.85	0.5	0.10	2.31	0.08	2.51	0.04
2	750	0.85	0.5	0.30	1.73	0.14	2.03	0.01

I used a probit regression model with each of the factors (*n*, *P*, *m*, *R*, and *D*), the two factor interactions *Pm*, *PD*, *mD*, *PR*, *MR*, *DR*, *P*log(*n*), *m*log(*n*), *D*log(*n*), and *R*log(*n*) and all sub-hierarchical interactions to fit the inverse normal cdf of the power of the LRTS test using level of significance 1%. The fitted model is:

$\Phi^{-1}(\hat{p})$=-6.26+

1.62D + 4.03m + 3.32R - 0.43P - 0.048log(log(S)) - 8.41(mD) - 5.84(mR) -

(t=1.1)   (t=5.66) (t=2.0) (t=-2.3) (t=0.1)   (t=-16.0)   (t=-2.7)

3.65(DR) + 0.235(DP) + 1.93(RP) + 4.54(Dlog(log(S)))

(t=-5.4)   (t=1.7)   (t=3.5)   (t=5.5)

The R^2 for the fitted model of 95.4% and shows a good fit. The logarithm of the sample size *S* is not significant at the 0.05 level, but its interaction with the difference between means is significant, with increasing power associated with increasing sample size. The mixing proportion *m* and difference between means *D* are significant, especially in the *m·D* interactions. The censoring rate *R*, and censoring pattern *P* are marginally significant. Larger difference between means, more symmetric mixing proportion, and lower censoring rate are associated with greater power.

Chapter 4. Conclusions

I studied the survival models assuming LTS (L_S) or the mixture of two exponential components (L_M) each with finite mean to test whether there is indication of a mixture mechanism. The estimated power of the LRTS for the LTS model and mixture survival model are modeled numerically in a simulation study. In the model L_S a finite study duration is considered

In L_S, the null distribution of the LRTS used to test LTS is shown to be plausibly asymptotically distributed as 50-50 mixture of a chi-square random variable with 1 degrees of freedom and a mass at zero as proved by Zhou and Maller (1995). Under the sample sizes, censoring patterns and study durations studied, the simulation results show the null distribution of LRTS for L_S to be well approximated by $\pi_{0,n}\chi_0^2 + (1-\pi_{0,n})\chi_1^2$, where $\pi_{0,n}$ is the fraction of zero LRTS, which varies with *n*. When considering study duration (*D*), I define I_D as either longer (I_D =1) or shorter (I_D =0). The simulation results show $\pi_{0,n}$ is fit by $0.50 + \dfrac{0.4864 + 0.9206 I_D}{\sqrt{n}}$, such that for large sample sizes the fraction of zeros is 50%. For longer study durations, the convergence to 50% fraction of zeros is shown to be slower than for shorter study durations. The censoring pattern and censoring rates are not significant in fitting $\pi_{0,n}$.

Power was estimated using the 99th percentile calculated under $\pi_{0,n}\chi_0^2 + (1-\pi_{0,n})\chi_1^2$ fit. A probit regression model was used to model the power over the variables used in the simulation. The estimated power model fit is

$$\Phi^{-1}(\hat{p}) = \frac{-36.2}{\sqrt{n}} - 3.93r + 2.16D + 79.19(LTS) + \frac{39.88r}{\sqrt{n}} - \frac{514.34(LTS)}{\sqrt{n}} - 3.08rD - 45.36r(LTS).$$

The model was fitted for censoring rates between 10% and 50%, study durations of 3 and 5 times the expected event time, LTS between 2% and 8% and sample sizes through 2,000. The censoring pattern was not significant. The R^2 for the fitted model is 92% and shows a good fit. All variables were highly significant. When the proportion of LTS is small the power is small, unless the sample size is large. In the event of high censoring rate a sample size of many thousand would be required to reject the null hypothesis, especially for small LRTS. In the case of long study durations, power near 50% can be achieved in some smaller sample size situations. In a longer study fewer than 150 observations are required for power of at least 50% when $LTS \geq 0.04$ and 10% of observations are observed as censored. For models with $0.02 \leq LTS < 0.04$, 50% power is estimated with fewer than 300 observations. In addition, fewer than 200 observations are required for an estimated 80% power with $LTS \geq 0.05$. The power steeply drops from a longer study to a shorter study. In a shorter study duration at least 50% estimated power is observed with fewer than 300 observations for models with $LTS \geq 0.05$. This is twice as many observations as required for the longer study.

At least 80% estimated power is observed with fewer than 550 observations for models with $LTS \geq 0.05$ for a shorter study.

The null distribution of the LRTS for L_M vs L_0 is well approximated by $\pi_1 \chi_0^2 + (1-\pi_1)\chi_{\nu_1}^2$, where π_1 is the fraction of zero LRTS values and is estimated by $\Phi[(0.385 \pm 0.09) - (0.641 \pm 0.052)\log(\log(n))]$ and ν_1 given by $(1.11 \pm 0.15) + (0.40 \pm 0.09)\log(\log(n))$. The fraction of zero LRTS, $\hat{\pi}_1$, is between 0.14 and 0.32 with average fraction of zero LRTS equal to 0.22. It decreases on average as *n* increases. The estimated “degrees of freedom,” $\hat{\nu}_1$ (that is, the mean of the non-zero LRTS values), is between 1.44 and 2.00 with an average of 1.80. It also increases on average as *n* increases.

A $2 \times 2 \times 3 \times 4 \times 4$ factorial experiment was run to estimate the power of the LRTS to detect the mixture of two exponential components. The 50-50 mixture with greater difference of component means (difference equals 1.5) has power near 1 for both censoring patterns and censoring rates, even for sample size 200. For skewed mixing proportions (mixing proportion equals 0.85) with greater difference of component means, power increases with increasing sample size, as expected. For smaller difference of component means (difference equals 0.50) for both symmetric (mixing proportion equals 0.50) and skewed mixtures, power is low for both censoring patterns and both censoring rates, and the power of test does not change much when sample size increases. In intermediate settings, a difference of 1.0 and mixing proportions of 0.65 and 0.75, power ranges from near zero to near one. The power is larger for smaller censoring rates. The probit multiple regression analysis is applied to the estimated power. The fitted model is:

$$\begin{aligned}\Phi^{-1}(\hat{p}) &= \text{-}6.26 + 1.62\mathrm{D} + 4.03\mathrm{m} + 3.32\mathrm{R} - 0.43\mathrm{P} - 0.048\log(\log(\mathrm{S})) - 8.41(\mathrm{mD}) \\ &- 5.84(\mathrm{mR}) - 3.65(\mathrm{DR}) + 0.235(\mathrm{DP}) + 1.93(\mathrm{RP}) + 4.54(\mathrm{D}\log(\log(\mathrm{S})))\end{aligned}$$

In general, the mixing proportion, difference of two component means and their interaction affected the average LRTS. The mixing proportion, difference of two component means, censoring rate and their interaction, affected the power and fraction of zero LRTS.

The extension of this dissertation is to study a model selection technique in determining whether L_0, L_S or L_M best fits a sample. A study of the Bayesian Information Criteria (BIC) to select the model would be a valuable contribution. The study of a mixture model with a long term survivor component is a further generalization of interest. The study of a Box-Cox transformation to each of these models would also be a valuable contribution.

References

Berkson, J., Gage, R.P. (1952). “Survival curve for cancer patients following treatment.” *Journal of the American Statistical Association*, 47, pp. 501-515.

Boag, J.W. (1949). “Maximum likelihood estimates of the proportion of patients cured by cancer therapy.” *Journal of the Royal Statistical Society*, 11, pp. 15-53.

Böhning, D, Seidel, W. (2003). “Editorial: recent developments in mixture models.” *Computational Statistics and Data Analysis*, 41, pp. 349-357.

Broet, P., Tubert-Bitte,r P., De Rycke, Y., and Moreau T. (2003), “A score test for establishing non-inferiority with respect to short-term survival in two-sample comparisons with identical proportions of long-term survivors.” *Statistics in Medicine*, 22, pp 931-940.

Broet, P., De Rycke, Y., (2001), “A semiparametric approach for the two-sample comparison of survival times with long-term survivors.” *Biometrics,* 57, pp 844-852.

Copas, J. B., Heydari, F. (1997). “Estimating the Risk of Reoffending by Using Exponential Mixture Models.” *Journal of the Royal Statistical Society*. Series A, 160, pp. 237-252.

Farewell, V. T. (1982). “The Use of Mixture Models for the Analysis of Survival Data with Long-Term Survivors.” *Biometrics*, 38, pp. 1041-1046.

Farewell, V. T. (1986). “Mixture models in survival analysis: Are they worth the risk?” *the Canadian Journal of Statistics*, 14, pp. 257-262.

Furukawa, K., Cologne, JB., Shimizu, Y. (2009). “Predicting Future Excess Events in Risk Assessment.” *Risk Analysis,* 29, pp 885-899

GNU Scientific Library (GSL) Website, http://www.gnu.org/software/gsl/

Haybittle, J.L. (1965). “A two-parameter model for the survival curve of treated cancer patients.” *Journal of the American Statistical Association*, 60, pp. 16-26.

Klein, J. P. ,Moeschberger, M. L. (2003). *Survival Analysis*. Springer, 2ed.

Lam, KF., Fong, DYT., Tang, OY. (2005). “Estimating the proportion of cured patients in a censored sample.” *Statistics in Medicine*, 24, pp 1865-1879

Liu X, Shao Y. (2004) Asymptotics for the likelihood ratio test in a two-component normal mixture model. *Journal of Statistical Planning and Inference* 2004; **123-1**:61-81.

Liu X, Pasarica C, Shao Y. (2003) Testing homogeneity in gamma mixture models. *Scand J Stat* 2003; **30**:227-239.

Maller, R. A., Zhou, S. (1992). “Estimating the Proportion of Immunes in a Censored Sample.” *Biometrika*, 79, pp. 731-739.

Maller, R. A., Zhou, S. (1995). “Testing for the Presence of Immune or Cured Individuals in Censored Survival Data.” *Biometrics*, 51, pp. 1197-1205.

Maller, R. A., Zhou, S. (1996). Survival Analysis with Long-Term Survivors. Wiley, Chichester, New York, London.

Nelder, J.A., Mead, R.,(1965) “A simplex method for function minimization.” *The Computer Journal*, 7 (4), pp. 308-313.

Peng,Y., Dear, K. B. G. and Denham, J. W. (1998). “A generalized F mixture model for cure rate estimation.” *Statistics in Medicine*, 17, pp. 813-830.

Peng,Y., Dear, K. B. G. and Carriere, K. C. (2001). “Testing for presence of cured patients: a simulation study.” *Statistics in Medicine*, 20, pp. 1783-1796.

Tsodikov, A. (2001). “Estimation of survival based on proportional hazards when cure is a possibility.” *Mathematical and Computer Modelling,* 33, pp. 1227-1236.

Tsodikov, A. (2002). “Semi-parametric models of long- and short-term survival: an application to the analysis of breast cancer survival in Utah by age and stage.” *Statistics in Medicine*, 21, pp. 895-920.

Tsodikov, A. D., Ibrahim, J. G. and Yakovlev, A. Y. (2003). “Estimation cure rates from survival data: an alternative to two-component mixture models.” *Journal of the American Statistical Association*, Vol. 98, No. 464, pp. 1063-1078.

Ye, Q (2006) “Survival Analysis Estimation and Testing Assuming a Two Component Exponential Mixture” SUNY Stony Brook Dissertation

Zhou, S., Maller, R. A. (1995) “Likelihood ratio test for the presence of immunes in a censored sample.” *Statistics*, 27, pp. 181-201.

Appendix:

A1: C Code for simulation of L_0 vs L_S

```
#include <stdio.h>
#include <math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
#include <gsl/gsl_multimin.h>
#define debug 0

struct _power_data
{
 double *t;
 double *u;
 double *c;
 int size;
};
typedef struct _power_data power_data;

double  my_f(const gsl_vector *v, void *params)
{
        power_data *my_pwr_data;
        my_pwr_data = (power_data*)params;

        double sumlog01 = 0;
        double *t, *u, *c;
        int n,i;
        t = my_pwr_data->t;
        u = my_pwr_data->u;
        c = my_pwr_data->c;
        n = my_pwr_data->size;
        double mu10 = gsl_vector_get(v, 0);
        double mu20 = gsl_vector_get(v, 1);
        double phi = gsl_vector_get(v, 2);
        double lambda10,lambda20,m0,expphi,x1,x2,logitem01;
        lambda10 = exp(mu10);
        lambda20 = exp(mu20);
        expphi =  exp(phi);
        m0 =  expphi /(1.0+expphi);

        for(i = 0; i < n; i++)
        {
                x1 = m0* exp(-(lambda10) * t[i]);
                x2 = (1-m0)*exp(-(lambda20)* t[i]);
                logitem01= c[i] * (log(lambda10*x1+ lambda20*x2))+ (1-
c[i])*log(x1+x2);
```

```
                sumlog01 = sumlog01+ logitem01;
        }

return -sumlog01;
}




main()
{
//////// GSL Initialization ////////////

// GSL random number realated varibles
const gsl_rng_type * T;
gsl_rng * r;
gsl_rng_env_setup();
T = gsl_rng_default;
r = gsl_rng_alloc (T);


// GSL minimizer realated varibles

size_t np = 3;
const gsl_multimin_fminimizer_type *TT =
        gsl_multimin_fminimizer_nmsimplex;

gsl_multimin_fminimizer *s = NULL;
gsl_vector *ss, *xx;
gsl_multimin_function minex_func;
size_t iter = 0, q;
xx = gsl_vector_alloc (np);


//////// End GSL Initialization ////////////


// Changable parameters
int n = 500; // sample size
int rep = 100; // repition number
double m = 0.85; // mixing proportion  (if no mixture then set to 1)
double d = 0.8; // difference in means
double LTS = 1.0;  // Long Term Survivor rate (if no LTS then set to 1)
double p = 1;  // censoring pattern (exp = 1  & unif = 2)
double  lambda1 = 1 + d*(m-1);
double  lambda2 = 1+d*m;

double a = 0.0; // uniform censoring min
double b = 10.66; // uniform censoring max 10.66=10% & 3.2=30%
double beta = 9.52; // exponential censoring param:  9.52=10% & 2.56=30%


// storage arrays and counters
int k;
double *select, *expLeft, *expRight, *u, *t, *tt, *c, *select_LTS, *expLTS, *tt_orig;
double *maxsumlog00, *lambda00;
double *maxsumlog01, *lambda01hat1, *lambda01hat2, *m01hat, *d0001,
*smallermean01hat, *largermean01hat, *mean00hat;

// storage arrays and counters
double *v, *lam1s, *lam2s, *ms;

/////// allocating memory


//arrays of size the sample size
select = malloc(sizeof(double)*n);
select_LTS = malloc(sizeof(double)*n);
expLeft = malloc(sizeof(double)*n);
expRight = malloc(sizeof(double)*n);
expLTS = malloc(sizeof(double)*n);
u = malloc(sizeof(double)*n);
t = malloc(sizeof(double)*n);
tt_orig = malloc(sizeof(double)*n);
tt = malloc(sizeof(double)*n);
c = malloc(sizeof(double)*n);

//arrays of size of rep
maxsumlog00 = malloc(sizeof(double)*rep);
lambda00 = malloc(sizeof(double)*rep);
maxsumlog01 = malloc(sizeof(double)*rep);
lambda01hat1 = malloc(sizeof(double)*rep);
lambda01hat2 = malloc(sizeof(double)*rep);
m01hat = malloc(sizeof(double)*rep);
d0001 = malloc(sizeof(double)*rep);
smallermean01hat = malloc(sizeof(double)*rep);
largermean01hat = malloc(sizeof(double)*rep);
mean00hat = malloc(sizeof(double)*rep);
```

```
// arrays of random size

lam1s = malloc(sizeof(double)*10);
lam2s = malloc(sizeof(double)*10);
ms = malloc(sizeof(double)*10);

/////// done allocating memory
printf("d0001\tmean00hat\tm01hat\tsmallermean01hat\tlargermean01hat\n");
for(k = 0; k < rep; k++)
{ // start of k/repition loop


// C code 1
int i;
//filling the array of select expLeft and expRight
if(debug == 2)
printf(" select\t expLeft\t expRight\t tt \n");

double expLeft_mean = 0 ,expRight_mean = 0, select_mean = 0 ;
for(i = 0; i < n; i++)
{
        select[i] = gsl_ran_flat(r,0.0,1.0);
        expLeft[i] = gsl_ran_exponential(r,lambda1);
        expRight[i] = gsl_ran_exponential(r,lambda2);

        expLeft_mean  = expLeft_mean  + expLeft[i]/(1.0*n);
        expRight_mean = expRight_mean + expRight[i]/(1.0*n);
        select_mean = select_mean + select[i]/(1.0*n);

        if (select[i] <= m )
                tt_orig[i] = expLeft[i];
        else
                tt_orig[i] = expRight[i];

        if(debug == 2)
        printf("%f %f %f %f \n",select[i],expLeft[i],expRight[i],tt[i]);

}


//expLeft_mean2 = expLeft_mean2/(1.0*n);
//select_mean = select_mean/(1.0*n);
//expLeft_mean = expLeft_mean/(1.0*n);
//expRight_mean= expRight_mean/(1.0*n);
```

```
double expLeft_var =0 ,expRight_var = 0, select_var =0;
for(i = 0; i < n; i++)
{
        select_var= select_var +(select[i]-select_mean)*(select[i]-select_mean);
        expLeft_var= expLeft_var +(expLeft[i]-expLeft_mean)*(expLeft[i]-
expLeft_mean);
        expRight_var= expRight_var +(expRight[i]-expRight_mean)*(expRight[i]-
expRight_mean);

}
        select_var = select_var/(1.0*n - 1.0);
        expLeft_var = expLeft_var/(1.0*n - 1.0);
        expRight_var = expRight_var/(1.0*n - 1.0);


        if(debug == 2)
        {
        printf("lambda1= %f lambda2= %f \n"
            "expLeft_mean = %f expRight_mean = %f ,  \n"
            "select_mean = %f realselect_mean = %f ,  \n"
            ,lambda1,lambda2,expLeft_mean,expRight_mean,select_mean,1.0/2);

     printf("real_expLeft_var = %f real_expRight_var = %f \n"
          "expLeft_var = %f expRight_var = %f \n"
            "real_select_var = %f \n"
            "select_var = %f \n"
            ,lambda1*lambda1,lambda2*lambda2,
            expLeft_var,expRight_var,
            1.0/12.0,
            select_var);
        }

// choosing a tt

// end C code 1


// C code 1.5
//filling the array of select expLeft and expRight
if(debug == 2)
printf(" select_LTS\t exponLTS\t");


for(i = 0; i < n; i++)
{
        select_LTS[i] = gsl_ran_flat(r,0.0,1.0);
        expLTS[i] = gsl_ran_exponential(r,9999); // assume the LTS mean = 9999

        //select_LTS_mean = select_LTS_mean + select_LTS[i]/(1.0*n);

        if (select_LTS[i] <= LTS )
                tt[i] = tt_orig[i];
        else
                tt[i] = expLTS[i];

        //if(debug == 2)
        //printf("%f %f %f %f \n",select_LTS[i],tt_orig[i],tt[i]);

}


//expLeft_mean2 = expLeft_mean2/(1.0*n);
//select_mean = select_mean/(1.0*n);
//expLeft_mean = expLeft_mean/(1.0*n);
//expRight_mean= expRight_mean/(1.0*n);

// end C code 1.5






// C code 2

//filling the array of u



double mean_u=0,mean_t=0;

if(debug == 2)
printf("tt u t c \n");

if(p == 2)
for(i = 0; i < n; i++)
{
        u[i] = gsl_ran_flat(r,a,b);
        t[i] = fmin((double)tt[i],(double)u[i]);
     mean_t = mean_t+t[i];
     mean_u = mean_u+u[i];
        if (tt[i] <= u[i])
```

```
    c[i] = 1.0;
    else c[i] = 0.0;

      if(debug == 2)
      printf("%f %f %f %f \n",tt[i],u[i],t[i],c[i]);
}


if(p == 1)
for(i = 0; i < n; i++)
{
      u[i] = gsl_ran_exponential(r,beta);
      t[i] = fmin((double)tt[i],(double)u[i]);
    mean_t = mean_t+t[i];
    mean_u = mean_u+u[i];
      if (tt[i] <= u[i])
    c[i] = 1.0;
    else c[i] = 0.0;

      if(debug == 2)
      printf("%f %f %f %f \n",tt[i],u[i],t[i],c[i]);
}


////////////////////
mean_t = mean_t/(1.0*n);
mean_u = mean_u/(1.0*n);

double t_var =0 ,u_var = 0;
for(i = 0; i < n; i++)
{
      u_var= u_var +(u[i]-mean_u)*(u[i]-mean_u);

}
u_var = u_var/(1.0*n - 1.0);

if(debug == 2)
{
printf( "mean_t = %f mean_u = %f \n"
    ,mean_t,mean_u);

printf("real_u_var = %f \n"
    "u_var = %f \n"
    ,((b-a)*(b-a))/12.0 ,u_var);

}
```

```
/////////////////////
// end C code 2

// C code 3

lambda00[k]  = 0;
double csum = 0;
double tsum = 0;
double  p0 = 1.0;
double logitem00;
double x;
double sumlog00 = 0;
for(i = 0; i < n; i++)
{
        csum = csum + c[i];
        tsum = tsum + t[i];
}
     lambda00[k] =  csum/tsum;

for(i = 0; i < n; i++)
{
          x  =  exp(-lambda00[k] * t[i]);
          logitem00 = c[i] * (log(p0)+log(lambda00[k]) - lambda00[k] * t[i] ) + (1.0-
c[i])*log(1.0 - p0 + p0*x);
               if(debug == 2)
               {
                     printf("x = %f \n",x);
                     printf("c[i] = %f \n",c[i]);
                     printf("p0 = %f \n",p0);
                     printf("lambda00[k] = %f \n",lambda00[k]);
                     printf("t[i] = %f \n",t[i]);
               printf("logitem00 = %f   sumlog00 = %f \n",logitem00,  sumlog00);
          }
               sumlog00 = sumlog00 + logitem00;
}
maxsumlog00[k] = sumlog00;

// end C code 3
// C code 4
for(i = 0; i < 10; i++)
{
lam1s[i] =  1.0/(mean_t*(-log(1-gsl_ran_flat(r,0.0,1.0))));
//lam1s <-  1/(mean(t)*(-log(1-v)));
//lam2s <- 1/(mean(t)*(-log(1-w)));
}

for(i = 0; i < 10; i++)
lam2s[i] =  1.0/(mean_t*(-log(1-gsl_ran_flat(r,0.0,1.0))));

for(i = 0; i < 10; i++)
            ms[i]=1-0.1*gsl_ran_flat(r,0.0,1.0);
//u  <- runif(7,0,1);
//ms <- 1-0.1*u;

int j,kk;
power_data my_pwr_data;
my_pwr_data.t = t;
my_pwr_data.u = u;
my_pwr_data.c = c;
my_pwr_data.size = n;


//i =1;
//j =1;
//kk = 1;

double maxf=-100000, max1 = -100000,max2=-100000,max3 = -100000;

for(i = 0; i < 3; i++)
{
for(j = 0; j < 3; j++)
{
for(kk = 0; kk < 1; kk++)
{
       if(debug == 1)
           printf("i=%i j=%i kk=%i\n",i,j,kk);

           s = gsl_multimin_fminimizer_alloc(TT, np);
           ss = gsl_vector_alloc(np);
           gsl_vector_set_all (ss, 1.0);
           gsl_vector_set(xx, 0, log(lam1s[i]));
           gsl_vector_set(xx, 1, log(lam2s[j]));
           double tmpdata= log(ms[kk]/(1.0- ms[kk])) ;
```

```
        gsl_vector_set(xx, 2, tmpdata );
minex_func.f = &my_f;
        minex_func.n = np;
        minex_func.params = (void *)&my_pwr_data;
        gsl_multimin_fminimizer_set(s, &minex_func, xx, ss);
     iter = 0;
        int status;
        double size;
     do
     {
       iter++;
       status = gsl_multimin_fminimizer_iterate(s);

       if(status)
        break;

       size = gsl_multimin_fminimizer_size (s);
       status = gsl_multimin_test_size (size, 1e-5);
       if(status == GSL_SUCCESS && debug == 5)
       {
        printf("\n++++++++++++++++++++++++\n");
        printf("step   = %d \ninputs = (", iter);
        printf("%f,", gsl_vector_get (s->x, 0));
        printf("%f,", gsl_vector_get (s->x, 1));
        printf("%f", gsl_vector_get (s->x, 2));
        printf(")\nf      = %f \n", s->fval);
        printf("size   = %f\n", size);
        printf("++++++++++++++++++++++++\n");
       }
     }
    while (status == GSL_CONTINUE && iter < 1200);
  //  if(iter > 1000)
  //  {
  //        printf("\n++++++++++++++++++++++++\n");
  //        printf("step   = %d ????? \n", iter);
  //        printf("++++++++++++++++++++++++\n");

   // }
    //return status;
/////////////////// done minimizing ////////////////////////
//if (- randommaxlog01result $value > maxsumlog10[k])  flag01 <- 1 else flag01 <- 0;
        if(maxf < s->fval*-1.0)
           {
                if(debug == 7)
                printf("maxf %10.17f will be replaced with %10.17f \n",maxf, s-
>fval*-.0);
```

```
          max1=  gsl_vector_get (s->x, 0);
            max2=  gsl_vector_get (s->x, 1);
            max3=  gsl_vector_get (s->x, 2);
            maxf= -1.0*s->fval;
        }
      if(debug == 7)
      {
      printf("============================\n");
      //printf("For starting points %f %f %f  is %f \n",s->fval);
      //
      printf(" starting point 1 %f \n",log(lam1s[i]));
      printf(" starting point 2 %f \n",log(lam2s[j]));
      printf(" starting point 3 %f \n",log(ms[kk]/(1.0- ms[kk])));
      printf("fval is %10.17f \n",s->fval);
      printf("============================\n");
      }
      gsl_multimin_fminimizer_free (s);
    gsl_vector_free(ss);

}}}
      if(debug == 7)
      {
      printf("============================\n");
      printf("the last f is %f input (%f,%f,%f)\n",maxf,max1,max2,max3);
      printf("============================\n");
      }

// end C code 4




// C code 5

 maxsumlog10[k] = maxf;
 lambda10hat[k] = exp(max1);
 m01hat[k] = exp(max2) /(1.0+exp(max2));

if(debug == 7)
printf("maxsumlog00[k] = %10.17f  maxsumlog10[k]) =%10.17f
\n",maxsumlog00[k],maxsumlog10[k]);


d0001[k] = -2.0*maxsumlog00[k]-(-2.0* maxsumlog10[k]) ;
mean10hat[k] = 1/lambda10hat[k];
```

```
mean00hat[k] = 1/lambda00[k];

//End C code 5


fflush(stdout);
fflush(stdout);

if (summary == 1)
{
printf("\n");
printf("Single Exponential:");
printf("\n");
printf("MLE mean: ");
printf("%f\n",mean00hat[k]);


printf("Max log0: ");
printf("%f\n",maxsumlog00[k]);
printf("\n");

printf("Single Exponential w/ LTS:");
printf("\n");

printf("MLE mean: ");
printf("%f\n",mean10hat[k]);

printf("p-hat: ");
printf("%f\n",m01hat[k]);


printf("Max log1: ");
printf("%f\n",maxsumlog10[k]);
printf("\n");

printf("LRTs sinlge w/ LTS vs single exponential:  ");
printf("%f\n",d0001[k]);
printf("\n");
}

if (summary == 2)
{
printf("%f\n",d0001[k]);
}
```

```
if (summary == 3)
{
printf("%10.17f\t%f\t%f\t%f\t%f
\n",d0001[k],mean00hat[k],m01hat[k],mean10hat[k],mean10hat[k]);

}
fflush(stdout);

//////////////////////////////

} // end of k/repition loop



        gsl_vector_free(xx);

        //gsl_multimin_fminimizer_free(s);

}
```

A2: C Code for simulation of L_0 vs L_M

```
#include <stdio.h>
#include <math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
#include <gsl/gsl_multimin.h>
#define debug 0

struct _power_data
{
 double *t;
 double *u;
 double *c;
 int size;
};
typedef struct _power_data power_data;

double  my_f(const gsl_vector *v, void *params)
{
      power_data *my_pwr_data;
      my_pwr_data = (power_data*)params;

      double sumlog01 = 0;
      double *t, *u, *c;
      int n,i;
      t = my_pwr_data->t;
      u = my_pwr_data->u;
      c = my_pwr_data->c;
      n = my_pwr_data->size;
      double mu10 = gsl_vector_get(v, 0);
      double mu20 = gsl_vector_get(v, 1);
      double phi = gsl_vector_get(v, 2);
      double lambda10,lambda20,m0,expphi,x1,x2,logitem01;
      lambda10 = exp(mu10);
      lambda20 = exp(mu20);
      expphi =  exp(phi);
      m0 =  expphi /(1.0+expphi);

      for(i = 0; i < n; i++)
      {
              x1 = m0* exp(-(lambda10) * t[i]);
              x2 = (1-m0)*exp(-(lambda20)* t[i]);
              logitem01= c[i] * (log(lambda10*x1+ lambda20*x2))+ (1-
c[i])*log(x1+x2);
```

```
                sumlog01 = sumlog01+ logitem01;
        }

return -sumlog01;
}




main()
{
//////// GSL Initialization ////////////

// GSL random number realated varibles
const gsl_rng_type * T;
gsl_rng * r;
gsl_rng_env_setup();
T = gsl_rng_default;
r = gsl_rng_alloc (T);


// GSL minimizer realated varibles

size_t np = 3;
const gsl_multimin_fminimizer_type *TT =
        gsl_multimin_fminimizer_nmsimplex;

gsl_multimin_fminimizer *s = NULL;
gsl_vector *ss, *xx;
gsl_multimin_function minex_func;
size_t iter = 0, q;
xx = gsl_vector_alloc (np);


//////// End GSL Initialization ////////////


// Changable parameters
int n = 500; // sample size
int rep = 100; // repition number
double m = 0.85; // mixing proportion  (if no mixture then set to 1)
double d = 0.8; // difference in means
double LTS = 1.0;  // Long Term Survivor rate (if no LTS then set to 1)
double p = 1;  // censoring pattern (exp = 1  & unif = 2)
double  lambda1 = 1 + d*(m-1);
double  lambda2 = 1+d*m;

double a = 0.0; // uniform censoring min
double b = 10.66; // uniform censoring max 10.66=10% & 3.2=30%
double beta = 9.52; // exponential censoring param:  9.52=10% & 2.56=30%


// storage arrays and counters
int k;
double *select, *expLeft, *expRight, *u, *t, *tt, *c, *select_LTS, *expLTS, *tt_orig;
double *maxsumlog00, *lambda00;
double *maxsumlog01, *lambda01hat1, *lambda01hat2, *m01hat, *d0001,
*smallermean01hat, *largermean01hat, *mean00hat;

// storage arrays and counters
double *v, *lam1s, *lam2s, *ms;

/////// allocating memory


//arrays of size the sample size
select = malloc(sizeof(double)*n);
select_LTS = malloc(sizeof(double)*n);
expLeft = malloc(sizeof(double)*n);
expRight = malloc(sizeof(double)*n);
expLTS = malloc(sizeof(double)*n);
u = malloc(sizeof(double)*n);
t = malloc(sizeof(double)*n);
tt_orig = malloc(sizeof(double)*n);
tt = malloc(sizeof(double)*n);
c = malloc(sizeof(double)*n);

//arrays of size of rep
maxsumlog00 = malloc(sizeof(double)*rep);
lambda00 = malloc(sizeof(double)*rep);
maxsumlog01 = malloc(sizeof(double)*rep);
lambda01hat1 = malloc(sizeof(double)*rep);
lambda01hat2 = malloc(sizeof(double)*rep);
m01hat = malloc(sizeof(double)*rep);
d0001 = malloc(sizeof(double)*rep);
smallermean01hat = malloc(sizeof(double)*rep);
largermean01hat = malloc(sizeof(double)*rep);
mean00hat = malloc(sizeof(double)*rep);
```

```
// arrays of random size

lam1s = malloc(sizeof(double)*10);
lam2s = malloc(sizeof(double)*10);
ms = malloc(sizeof(double)*10);

/////// done allocating memory
printf("d0001\tmean00hat\tm01hat\tsmallermean01hat\tlargermean01hat\n");
for(k = 0; k < rep; k++)
{ // start of k/repition loop


// C code 1
int i;
//filling the array of select expLeft and expRight
if(debug == 2)
printf(" select\t expLeft\t expRight\t tt \n");

double expLeft_mean = 0 ,expRight_mean = 0, select_mean = 0 ;
for(i = 0; i < n; i++)
{
        select[i] = gsl_ran_flat(r,0.0,1.0);
        expLeft[i] = gsl_ran_exponential(r,lambda1);
        expRight[i] = gsl_ran_exponential(r,lambda2);

        expLeft_mean  = expLeft_mean  + expLeft[i]/(1.0*n);
        expRight_mean = expRight_mean + expRight[i]/(1.0*n);
        select_mean = select_mean + select[i]/(1.0*n);

        if (select[i] <= m )
                tt_orig[i] = expLeft[i];
        else
                tt_orig[i] = expRight[i];

        if(debug == 2)
        printf("%f %f %f %f \n",select[i],expLeft[i],expRight[i],tt[i]);

}


//expLeft_mean2 = expLeft_mean2/(1.0*n);
//select_mean = select_mean/(1.0*n);
//expLeft_mean = expLeft_mean/(1.0*n);
//expRight_mean= expRight_mean/(1.0*n);
```

```
double expLeft_var =0 ,expRight_var = 0, select_var =0;
for(i = 0; i < n; i++)
{
        select_var= select_var +(select[i]-select_mean)*(select[i]-select_mean);
        expLeft_var= expLeft_var +(expLeft[i]-expLeft_mean)*(expLeft[i]-
expLeft_mean);
        expRight_var= expRight_var +(expRight[i]-expRight_mean)*(expRight[i]-
expRight_mean);

}
        select_var = select_var/(1.0*n - 1.0);
        expLeft_var = expLeft_var/(1.0*n - 1.0);
        expRight_var = expRight_var/(1.0*n - 1.0);


        if(debug == 2)
        {
        printf("lambda1= %f lambda2= %f \n"
             "expLeft_mean = %f expRight_mean = %f ,  \n"
             "select_mean = %f realselect_mean = %f ,  \n"
             ,lambda1,lambda2,expLeft_mean,expRight_mean,select_mean,1.0/2);

     printf("real_expLeft_var = %f real_expRight_var = %f \n"
          "expLeft_var = %f expRight_var = %f \n"
             "real_select_var = %f \n"
             "select_var = %f \n"
             ,lambda1*lambda1,lambda2*lambda2,
             expLeft_var,expRight_var,
             1.0/12.0,
             select_var);
        }

// choosing a tt

// end C code 1


// C code 1.5
//filling the array of select expLeft and expRight
if(debug == 2)
printf(" select_LTS\t exponLTS\t");


for(i = 0; i < n; i++)
{
        select_LTS[i] = gsl_ran_flat(r,0.0,1.0);
        expLTS[i] = gsl_ran_exponential(r,9999); // assume the LTS mean = 9999

        //select_LTS_mean = select_LTS_mean + select_LTS[i]/(1.0*n);

        if (select_LTS[i] <= LTS )
                tt[i] = tt_orig[i];
        else
                tt[i] = expLTS[i];

        //if(debug == 2)
        //printf("%f %f %f %f \n",select_LTS[i],tt_orig[i],tt[i]);

}


//expLeft_mean2 = expLeft_mean2/(1.0*n);
//select_mean = select_mean/(1.0*n);
//expLeft_mean = expLeft_mean/(1.0*n);
//expRight_mean= expRight_mean/(1.0*n);

// end C code 1.5
```

```
// C code 2

//filling the array of u



double mean_u=0,mean_t=0;

if(debug == 2)
printf("tt u t c \n");

if(p == 2)
for(i = 0; i < n; i++)
{
        u[i] = gsl_ran_flat(r,a,b);
        t[i] = fmin((double)tt[i],(double)u[i]);
     mean_t = mean_t+t[i];
     mean_u = mean_u+u[i];
        if (tt[i] <= u[i])
```

```
    c[i] = 1.0;
    else c[i] = 0.0;

      if(debug == 2)
      printf("%f %f %f %f \n",tt[i],u[i],t[i],c[i]);
}


if(p == 1)
for(i = 0; i < n; i++)
{
      u[i] = gsl_ran_exponential(r,beta);
      t[i] = fmin((double)tt[i],(double)u[i]);
    mean_t = mean_t+t[i];
    mean_u = mean_u+u[i];
      if (tt[i] <= u[i])
    c[i] = 1.0;
    else c[i] = 0.0;

      if(debug == 2)
      printf("%f %f %f %f \n",tt[i],u[i],t[i],c[i]);
}


///////////////////
mean_t = mean_t/(1.0*n);
mean_u = mean_u/(1.0*n);

double t_var =0 ,u_var = 0;
for(i = 0; i < n; i++)
{
      u_var= u_var +(u[i]-mean_u)*(u[i]-mean_u);

}
u_var = u_var/(1.0*n - 1.0);

if(debug == 2)
{
printf( "mean_t = %f mean_u = %f \n"
    ,mean_t,mean_u);

printf("real_u_var = %f \n"
    "u_var = %f \n"
    ,((b-a)*(b-a))/12.0 ,u_var);

}
```

```
////////////////////
// end C code 2

// C code 3

lambda00[k]  = 0;
double csum = 0;
double tsum = 0;
double  p0 = 1.0;
double logitem00;
double x;
double sumlog00 = 0;
for(i = 0; i < n; i++)
{
            csum = csum + c[i];
            tsum = tsum + t[i];
}
        lambda00[k] =  csum/tsum;

for(i = 0; i < n; i++)
{
                x  =  exp(-lambda00[k] * t[i]);
                logitem00 = c[i] * (log(p0)+log(lambda00[k]) - lambda00[k] * t[i] ) + (1.0-
c[i])*log(1.0 - p0 + p0*x);
                        if(debug == 2)
                        {
                                    printf("x = %f \n",x);
                                    printf("c[i] = %f \n",c[i]);
                                    printf("p0 = %f \n",p0);
                                    printf("lambda00[k] = %f \n",lambda00[k]);
                                    printf("t[i] = %f \n",t[i]);
                        printf("logitem00 = %f   sumlog00 = %f \n",logitem00,  sumlog00);
                }
                        sumlog00 = sumlog00 + logitem00;
}
maxsumlog00[k] = sumlog00;

// end C code 3


// C code 4
for(i = 0; i < 10; i++)
{
lam1s[i] =  1.0/(mean_t*(-log(1-gsl_ran_flat(r,0.0,1.0))));
```

```
//lam1s <-  1/(mean(t)*(-log(1-v)));
//lam2s <- 1/(mean(t)*(-log(1-w)));
}

for(i = 0; i < 10; i++)
lam2s[i] =  1.0/(mean_t*(-log(1-gsl_ran_flat(r,0.0,1.0))));

for(i = 0; i < 10; i++)
          ms[i]=1-0.1*gsl_ran_flat(r,0.0,1.0);
//u  <- runif(7,0,1);
//ms <- 1-0.1*u;

int j,kk;
power_data my_pwr_data;
my_pwr_data.t = t;
my_pwr_data.u = u;
my_pwr_data.c = c;
my_pwr_data.size = n;


//i =1;
//j =1;
//kk = 1;

double maxf=-100000, max1 = -100000,max2=-100000,max3 = -100000;

for(i = 0; i < 3; i++)
{
for(j = 0; j < 4; j++)
{
for(kk = 0; kk < 5; kk++)
{
      if(debug == 1)
         printf("i=%i j=%i kk=%i\n",i,j,kk);

         s = gsl_multimin_fminimizer_alloc(TT, np);
         ss = gsl_vector_alloc(np);
         gsl_vector_set_all (ss, 1.0);
         gsl_vector_set(xx, 0, log(lam1s[i]));
         gsl_vector_set(xx, 1, log(lam2s[j]));
         double tmpdata= log(ms[kk]/(1.0- ms[kk])) ;
         gsl_vector_set(xx, 2, tmpdata );
         minex_func.f = &my_f;
         minex_func.n = np;
         minex_func.params = (void *)&my_pwr_data;
         gsl_multimin_fminimizer_set(s, &minex_func, xx, ss);
```

```
        iter = 0;
            int status;
            double size;
        do
        {
          iter++;
          status = gsl_multimin_fminimizer_iterate(s);

          if(status)
            break;

          size = gsl_multimin_fminimizer_size (s);
          status = gsl_multimin_test_size (size, 1e-5);
          if(status == GSL_SUCCESS && debug == 5)
          {
            printf("\n++++++++++++++++++++++++\n");
            printf("step   = %d \ninputs = (", iter);
            printf("%f,", gsl_vector_get (s->x, 0));
            printf("%f,", gsl_vector_get (s->x, 1));
            printf("%f", gsl_vector_get (s->x, 2));
            printf(")\nf      = %f \n", s->fval);
            printf("size   = %f\n", size);
            printf("++++++++++++++++++++++++\n");
          }
        }
       while (status == GSL_CONTINUE && iter < 1200);
    //  if(iter > 1000)
    //  {
    //        printf("\n++++++++++++++++++++++++\n");
    //        printf("step   = %d ????? \n", iter);
    //        printf("++++++++++++++++++++++++\n");

     // }
       //return status;
//////////////////// done minimizing ////////////////////////
//if (- randommaxlog01result $value > maxsumlog01[k])  flag01 <- 1 else flag01 <- 0;
            if(maxf < s->fval*-1.0)
                {
                        if(debug == 7)
                        printf("maxf %10.17f will be replaced with %10.17f \n",maxf, s-
>fval*-1.0);

                    max1=  gsl_vector_get (s->x, 0);
                        max2=  gsl_vector_get (s->x, 1);
                        max3=  gsl_vector_get (s->x, 2);
                        maxf= -1.0*s->fval;
                }
            if(debug == 7)
            {
            printf("============================\n");
            //printf("For starting points %f %f %f  is %f \n",s->fval);
            //
            printf(" starting point 1 %f \n",log(lam1s[i]));
            printf(" starting point 2 %f \n",log(lam2s[j]));
            printf(" starting point 3 %f \n",log(ms[kk]/(1.0- ms[kk])));
            printf("fval is %10.17f \n",s->fval);
            printf("============================\n");
            }
            gsl_multimin_fminimizer_free (s);
        gsl_vector_free(ss);

}}}
            if(debug == 7)
            {
            printf("============================\n");
            printf("the last f is %f input (%f,%f,%f)\n",maxf,max1,max2,max3);
            printf("============================\n");
            }

// end C 4



// C code 5

 maxsumlog01[k] = maxf;
 lambda01hat1[k] = exp(max1);
 lambda01hat2[k] = exp(max2);
 m01hat[k] = exp(max3) /(1.0+exp(max3));

if(debug == 7)
printf("maxsumlog00[k] = %10.17f  maxsumlog01[k]) =%10.17f
\n",maxsumlog00[k],maxsumlog01[k]);


d0001[k] = -2.0*maxsumlog00[k]-(-2.0* maxsumlog01[k]) ;
smallermean01hat[k] = fmin(1/lambda01hat1[k],1/lambda01hat2[k]);
largermean01hat[k] = fmax(1/lambda01hat1[k], 1/lambda01hat2[k]);
if(lambda01hat1[k] < lambda01hat2[k])
     m01hat[k]  = 1- m01hat[k];

mean00hat[k] = 1/lambda00[k];
```

```
//End C code 5

fflush(stdout);
printf("%10.17f\t%f\t%f\t%f\t%f
\n",d0001[k],mean00hat[k],m01hat[k],smallermean01hat[k],largermean01hat[k]);
fflush(stdout);


//////////////////////////////

} // end of k/repition loop
      gsl_vector_free(xx);
      //gsl_multimin_fminimizer_free(s);

}
```