

INFERENCE ON SURVIVAL DATA UNDER NONPROPORTIONAL HAZARDS

by

Qing Xu

B. Med., HeBei Medical College, 1994

M. S., Biostatistics, University of Pittsburgh, 2004

Submitted to the Graduate Faculty of
Graduate School of Public Health in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Pittsburgh

2007

UNIVERSITY OF PITTSBURGH

Graduate School of Public Health

This dissertation was presented

by

Qing Xu

It was defended on

April 23, 2007

and approved by

Dissertation Advisor:
Jong-Hyeon Jeong, PhD
Associate Professor
Department of Biostatistic
Graduate School of Public Health
University of Pittsburgh

Committee Member:
Stewart Anderson, PhD
Professor
Department of Biostatistics
Graduate School of Public Health
University of Pittsburgh

Committee Member:
Lawrence Kingsley, PhD
Associate Professor
Department of Infectious Diseases and Microbiology
Graduate School of Public Health
University of Pittsburgh

Committee Member:
Abdus Wahed, PhD
Assistant Professor
Department of Biostatistics
Graduate School of Public Health
University of Pittsburgh

INFERENCE ON SURVIVAL DATA UNDER NONPROPORTIONAL HAZARDS

Qing Xu, Ph. D.

University of Pittsburgh, 2007

Abstract

The objective of this research is to develop optimal (efficient) test methods for analysis of survival data under random censorship with nonproportional hazards. For the first part we revisit the weighted log-rank test where the weight function was derived by assuming the inverse Gaussian distribution for an omitted exponentiated covariate that induces a nonproportionality under the proportional hazards model. We perform a simulation study to compare the new procedure with ones using other popular weight functions including members of the Harrington-Fleming's G-rho family. The nonproportional hazards data are generated by changing the hazard ratios over time under the proportional hazards model. The results indicate that the inverse Gaussian-based test tends to have higher power than some of the members that belong to the G-rho family in detecting a difference between two survival distributions when populations become homogeneous as time progresses.

The second part of the research includes development of a parametric method in detecting the validity of the proportional odds model assumption between two groups of survival data. The research is based on the premise that the test procedure developed would take advantage of knowledge of the distributional information about the data, which will improve the sensitivity of a nonparametric test method. We evaluate type I error and power probabilities of the new parametric test by using the simulated survival data following the log-logistic distribution. The error probabilities are compared with ones in the literature. The results indicate that the extended test performs with a higher sensitivity than the existing nonparametric method.

The results from the proposed study provide statistical test methods that are more sensitive than existing ones under certain situations which can be used in public health relevance applications such as clinical trials.

TABLE OF CONTENTS

1.	INTRODUCTION	1
1.1.	Survival Data	1
1.2.	Proportional Hazards Model.....	3
1.3.	Proportional Odds Model.....	5
1.4.	Project Overview	7
2.	LITERATURE REVIEW	9
2.1.	Proportional Hazards Modeling.....	9
2.2.	Proportional Odds Modeling.....	14
3.	TEST FOR PROPORTIONAL HAZARDS.....	18
3.1.	Weighted Log-rank Test	18
3.2.	Derivation of the New Weighting Function.....	19
3.3.	Test of Frailty Distributions.....	27
3.4.	Simulation Methodology	33
3.5.	Results and Discussion	36
3.5.1.	Test for proportional hazard assumptions.....	36
3.5.2.	Choosing parameters.....	41
3.5.3.	Simulation test for data difference.....	42
3.5.4.	Optimal value of parameter ψ	52
3.5.5.	Comparison with supreme weighted log-rank test.....	54
3.5.6.	Application in real data.....	56
3.6.	Conclusion	59
4.	TEST FOR PROPORTIONAL ODDS.....	60
4.1.	Introduction to the Dauxois-Kirmani Test.....	60
4.2.	Extending the Method with Parametric Assumptions	63
4.3.	Estimation of Parameters in Assumed Distributions	68
4.4.	Simulation Methodology	72

4.5.	Results and Discussions.....	75
4.5.1.	Examination of the existing method.....	75
4.5.2.	Simulation analysis of the extended method.....	78
4.5.3.	Application in real data.....	80
4.6.	Conclusion.....	81
5.	FUTURE WORK.....	82
5.1.	Determination of Optimal Values for ψ and ρ	82
5.2.	Linear Regression Test of Nonproportional Odds.....	87
	APPENDIX A.....	90
	Simulation Programs.....	90
	BIBLIOGRAPHY.....	124

LIST OF TABLES

Table 1. Simulation results for samples with early difference using the simple log-rank, Harrington-Fleming's weighted log-rank test, and the new weighted logrank test.....	46
Table 2. Simulation results for samples with middle difference using the simple log-rank, Harrington-Fleming's weighted log-rank test, and the new weighted logrank test.....	47
Table 3. Simulation results for samples with late difference using the simple log-rank, Harrington-Fleming's weighted log-rank test, and the new weighted logrank test.....	48
Table 4. Impact of censoring level on the sensitivity of the methods (early difference).....	49
Table 5. Impact of sample size on the sensitivity of the methods (early difference).	50
Table 6. Simulation results for samples with Weibull distribution (τ, ρ), with parameter $\tau = 1.5$	51
Table 7. Comparison of supreme and traditional version of the new weighted log-rank test.	56
Table 8. Reproduction of the Monte Carlo simulation results of the original paper using the Dauxois-Kirmani approach.....	77
Table 9. Monte Carlo simulation results of the log-logistic distribution data using the extended method.....	79

LIST OF FIGURES

Figure 1. Diagnostic plots of the Weibull survival function with the Gamma frailty.	31
Figure 2. Diagnostic plots for the Weibull survival function with the Inverse Gaussian frailty. .	32
Figure 3. Example of a single simulated data set with late difference.	38
Figure 4. Example of a single simulated data set with early difference.	39
Figure 5. Example of a single simulated data set with middle difference.	40
Figure 6. New weight function as a function of survival time and parameter ψ	53
Figure 7. Survival function in NSABP B-14 data estimated from Kaplan-Meier method.	58
Figure 8. Kaplan-Meier estimators of survival distribution of the log-logistic data.	76

1. INTRODUCTION

1.1. Survival Data

Survival data usually refers to data in the form of a time from a well-defined time origin until the occurrence of some particular event of interest. In medical research, the time origin often correspond to the recruitment of an individual into an experimental study, such as a clinical trial to compare two or more treatments. The end point may correspond to the relief of pain, the recurrence of symptoms, or the death of a patient.

In summarizing survival data, two functions of central interest are the survival function and the hazard function. The survival function, $S(t)$, is defined to be the probability that the time to event is greater than or equal to t ,

$$S(t) = P(T \geq t), \quad (1.1)$$

and the hazard function is defined as

$$h(t) = \lim_{\Delta \rightarrow 0} \left\{ \frac{P(t \leq T < t + \Delta | T \geq t)}{\Delta} \right\}, \quad (1.2)$$

which is the limiting conditional probability of experiencing an end point immediately after time t given the event has not occurred to the individual up to time t (Collett, 2003).

If T is a continuous random variable, then $S(t)$ is a continuous, strictly decreasing function from unity to zero. It can be conveniently expressed as the complement of the cumulative distribution function of the variable T , that is,

$$S(t) = 1 - F(t). \quad (1.3)$$

where

$$F(t) = P(T \leq t). \quad (1.4)$$

Therefore, the survival function can also be obtained by

$$S(t) = P(T \geq t) = \int_t^{\infty} f(v)dv, \quad (1.5)$$

where $f(t)$ is the probability distribution function of the survival time variable t . Accordingly, we have

$$f(t) = -\frac{dS(t)}{dt}. \quad (1.6)$$

When T is continuous, the relationship between the hazard function and the survival function is given by

$$h(t) = \frac{f(t)}{S(t)} = -\frac{d \ln[S(t)]}{dt}, \quad (1.7)$$

and

$$S(t) = \exp\left[-\int_0^t h(v)dv\right]. \quad (1.8)$$

For discrete survival variables, the hazard function is given by

$$h(t_j) = P(T = t_j | T \geq t_j) = \frac{p(t_j)}{S(t_{j-1})}, \quad (1.9)$$

where $p(t_j)$ is the probability that the interested event occurs at time t_j . And the survival function is related to the hazard function by

$$S(t) = \prod_{t_j \leq t} [1 - h(t_j)] = \sum_{t_j \geq t} p(t_j). \quad (1.10)$$

As shown above, for continuous time variables, expressions of $S(t)$ and $h(t)$ can be derived analytically if the underlying distribution of survival variable is known or specified. Analysis of survival data with a distributional assumption is referred to as parametric. Parametric inference is often based on the maximum likelihood theory.

If the analysis is performed without assumption about the underlying distribution of the survival data, the method is referred to as nonparametric. Commonly used nonparametric estimation methods of survival functions include Kaplan-Meier estimator (Kaplan and Meier, 1958) and Nelson-Aalen estimator (Altshuler, 1970; Nelson, 1972, Aalen, 1978). For either method, one first needs to arrange the times by rank. The Kaplan-Meier estimate of the survival function is calculated by

$$\hat{S}(t) = \prod_{t_j \leq t} \left(1 - \frac{d_j}{Y_j} \right), \quad (1.11)$$

where d_j is number of events occurred at time t_j , and Y_j is number of objects at risk at the beginning of time t_j . The Nelson-Aalen estimator is given by

$$\hat{S}(t) = \prod_{t_j \leq t} \exp \left(-\frac{d_j}{Y_j} \right). \quad (1.12)$$

1.2. Proportional Hazards Model

A regression model can be used when we want to relate potential prognostic factors or covariates to the length of time to a particular end point (survival time). Often we want to make

inferences about the association between the survival time and certain covariates (explanatory variables) rather than only estimate the survival function for one-sample. Therefore, we need to compare survival distributions among at least two groups adjusted for some covariates. For such comparison, the usual null hypothesis is that there is no difference among survival distributions from different selected comparison groups. The survival distributions are believed to be statistically different if the null hypothesis is rejected, and vice versa.

One of the most widely used regression model for survival data is the proportional hazards model (Cox, 1972). In the proportional hazards model, the ratio of hazards functions between two groups of interest is constant over time. In clinical studies the proportional hazards model can be used to compare survival distributions of two groups of patients, with one group being a treatment group and the other group being the placebo group, adjusted for important prognostic factors.

The proportional hazards model can be expressed by the relationship

$$h_1(t) = \psi h_2(t), \quad \psi = e^{\beta z} \quad (1.13)$$

where $h_1(t)$ and $h_2(t)$ are the hazard functions at time t for groups I and II, respectively, when there exists only one covariate of the group indicator z . The proportionality parameter ψ is a constant over time t . If $S_1(t)$ and $S_2(t)$ are the survival functions for two groups, from Eqs. (1.8) and (1.13) we can derive that

$$S_1(t) = [S_2(t)]^\psi. \quad (1.14)$$

Since the survival function takes values between zero and unity, Eq. (1.14) implies that $S_1(t)$ is always greater than or less than $S_2(t)$, depending on whether the proportionality parameter ψ is less than or greater than unity. This implies that if two hazard functions are proportional, the

survival functions for the two groups do not cross one another. This is a necessary, but not a sufficient condition for proportional hazards.

Based on this condition, we can evaluate the validity of the proportional hazards assumption by plotting the two estimated survival functions together. If the two estimated survival functions do not cross, the assumption of proportional hazards is justified. This method is straightforward and simple for most survival data. In most cases, unless a plot of the estimated survival functions indicates that there is good reason to doubt the proportional hazards assumption, proportional hazards model can often be used to compare the two samples.

Standard procedures for performing the proportional hazards regression are built in the commonly used statistical softwares such as SAS and S-plus.

In this work, we propose a log-rank test method with a newly derived weight function to compare two survival distributions with nonproportional hazards. We performed Monte Carlo simulations and real data analysis to evaluate performance of the new method comparing with the existing simple log-rank test and Harrington-Fleming's test.

1.3. Proportional Odds Model

Another less widely used but important model in survival analysis is the proportional odds model (Bennett, 1983a). In the proportional odds model, the hazard functions of the two groups are not proportional over time but their ratio converges to unity as time progresses. Such a situation is encountered, for example, when the initial effects of treatment, or the differences between stages of the disease at diagnosis, tend to diminish with time so that different groups of patients become homogeneous.

The odds of an individual surviving beyond some time t (survival odds) are expressed as

$$\phi(t) = \frac{S(t)}{F(t)} = \frac{S(t)}{1 - S(t)}. \quad (1.15)$$

Sometimes also used is the failure odds, which is defined as the inverse of survival odds.

The proportional odds implies that the relationship between the survival odds from group I and from group II,

$$\phi_1(t) = \frac{S_1(t)}{1 - S_1(t)} = \psi, \quad \phi_2(t) = \psi \frac{S_2(t)}{1 - S_2(t)}, \quad (1.16)$$

where ψ is a constant proportionality parameter that does not depend on time t .

Particularly important property of the proportional odds model concerns the hazard ratio of the two group. Eq. (1.16) gives

$$S_1(t) = \frac{S_2(t)}{1/\psi + (1 - 1/\psi)S_2(t)}. \quad (1.17)$$

Taking logarithms and the first derivatives upon t on both sides of Eq. (1.17) gives

$$\frac{h_1(t)}{h_2(t)} = \frac{1}{1 + (\psi - 1)S_2(t)}. \quad (1.18)$$

As t increases from 0 to ∞ , the survival function $S_2(t)$ decreases monotonically from unity to zero. Therefore, Eq. (1.18) indicates that under the proportional odds assumption the hazard ratio between two groups would converge from the value ψ at time $t = 0$, to unity at $t = \infty$.

In practical applications, it is common for the hazard functions in two or more groups to converge with time. For example, in a follow-up study of patients in a clinical trial, the effect on survival of the treatment, or the initial stage of disease, may wear out, converging over time. So the proportional odds model, with its property of convergent hazard functions, can be of considerable value. Of course, the inference under this model would only be valid when the

proportional odds assumption holds. In a two group comparison study, a preliminary examination of the likely suitability of the proportional odds model can be undertaken by making use of the relationship in Eq. (1.16). The Kaplan-Meier estimate of the survival function is obtained for each group and the estimated log-odds of survival beyond time t , $\ln \left[\frac{\hat{S}_1(t)}{1 - \hat{S}_1(t)} / \frac{\hat{S}_2(t)}{1 - \hat{S}_2(t)} \right]$, are plotted against $\ln t$. If the plot shows a flat line, this would

indicate that the proportional odds assumption is valid. However, the results from this graphical method can be quite ambiguous, mainly due to the significant fluctuations in $\ln \frac{\hat{S}_i(t)}{1 - \hat{S}_i(t)}$

calculated from survival data. This problem is especially severe when the size of survival data is small. In concern of this, a parametric or semiparametric method with less ambiguity for formal testing of the validity of proportional odds assumption between two groups is highly desired. In this study, we examine a nonparametric test statistic proposed by Dauxois and Kirmani (Dauxois and Kirmani, 2003) and propose a new parametric method to test the proportional odds assumption.

1.4. Project Overview

This thesis includes two different topics. The first part is on the development of a new method for testing any difference between two survival distributions under non-proportional hazards situations. Namely, we propose a new weighted log-rank test method to test the difference of two groups of survival data with various types of departure patterns. The second part is on testing the validity of model assumption. We extend the Dauxois-Kirmani method (Dauxois and Kirmani,

2003) by using the weight function derived from the parametric assumption for testing the validity of the proportional odds assumption. In both topics, we show that the modified test methods perform with a higher sensitivity compared with their existing counterparts.

The project is organized as follows

1. For the development of a new statistic method for testing the difference between two survival distribution with non-proportional hazards, there are following contents (1) Test for difference between two survival distributions generated from Monte Carlo simulations with nonproportional hazards using the simple log-rank test and the Harrington-Fleming's weighted log-rank test; (2) Derivation of a weight function for the new test method (Oakes and Jeong, 1998); (3) Test for difference between two simulated survival distributions with nonproportional hazards using the new test method; (4) Comparison of the sensitivity of the new method with that of the existing methods; (5) Applications to real data.

2. For the development of new parametric method for testing the validity of proportional odds assumption between two survival distributions, there are following contents. (1) Examination of nonparametric Dauxois-Kirmani method for testing the validity of proportional odds assumption for two survival distributions; (2) Derivation of a new parametric test method by parameterizing the Dauxois-Kirmani method; (3) Test for proportional odds assumption for simulated the log-logistic survival data using the new method; (4) Comparison of the sensitivity of the new method with that of the existing nonparametric method; (6) Applications to real data.

2. LITERATURE REVIEW

In this chapter we review the key literature on comparing non-proportional hazards distributions difference and proportional odds model assumption.

2.1. Proportional Hazards Modeling

The most widely used method of estimating the effects of covariates on survival times is the proportional hazards model proposed by Cox (Cox, 1972). The Cox model assumes that the ratio of the hazards between two levels of a covariate (i.e treatment group) is constant over time. It is analytically expressed in the form

$$h_i(t) = e^{\beta x_i} h_0(t), \quad (2.1)$$

where $h_j(t)$ denotes the hazard function for the i^{th} patient, $i = 1, 2, \dots, n$ and x_i is the value that the i^{th} patient takes for the explanatory variable X . The term $h_0(t)$ is the baseline hazard function. Thus, the null hypothesis that there is no difference in survival distribution between groups corresponds to the null hypothesis $\beta = 0$ in the model presented in Eq. (2.1) when $i = 1, 2, \dots, n$ is a group indicator.

A variety of testing methods have been developed to compare two survival distributions. In particular, the simple log-rank test (Savage, 1956; Mantel, 1966; Peto, 1972) is suitable for data with proportional hazards, and the G-rho tests (Harrington and Fleming 1982) are good for comparing survival distributions with nonproportional hazards.

The simple log-rank test (Savage, 1956; Mantel, 1966; Peto, 1972) is perhaps the most widely used method in two-sample comparisons of time-to-event data. It is simple to use, nonparametric in nature, and highly efficient under the proportional hazards assumptions. It incorporates the commonly encountered right censorship of survival data without adding complicated elements to the method itself. The general idea behind any test of the difference of K groups of survival data is to compare the survival distributions by calculating the statistics

$$Z_j(\tau) = \sum_{i=1}^D W_j(t_i) \left[\frac{d_{ij}}{Y_{ij}} - \frac{d_i}{Y_i} \right], \quad j = 1, \dots, K. \quad (2.2)$$

Here, we assume the survival data consist of independent right censored samples from K populations, and $t_1 < t_2 < \dots < t_D$ are the distinct event times in the pooled sample. At time t_i , we observe d_{ij} events in the j^{th} sample out of Y_{ij} individuals at risk, $j = 1, 2, \dots, K$, $i = 1, 2, \dots, D$, $d_i = \sum_{j=1}^K d_{ij}$ and $Y_i = \sum_{j=1}^K Y_{ij}$ are the number of events and number of objects at risk in the combined sample at time t_i , $i = 1, 2, \dots, D$. The weight function $W_j(t)$ is positive. The terms d_{ij}/Y_{ij} and d_i/Y_i are the Nelson-Aalen estimator (Altshuler, 1970; Nelson, 1972; and Aalen, 1978) of the hazard rate in the j^{th} group and the combined sample, respectively.

In practice, all of the commonly used tests have a weight function $W_j(t_i) = Y_{ij}W(t_i)$. Here, $W(t_i)$ is a common weight shared by each group. With this choice of weight function Eq. (2.2) becomes

$$Z_j(\tau) = \sum_{i=1}^D W(t_i) \left[d_{ij} - Y_{ij} \left(\frac{d_i}{Y_i} \right) \right], \quad j = 1, \dots, K. \quad (2.3)$$

The variance and covariance of $Z_j(\tau)$ is given by

$$\sigma_{jj}(\tau) = \sum_{i=1}^D [W(t_i)]^2 \frac{Y_{ij}}{Y_i} \left(1 - \frac{Y_{ij}}{Y_i}\right) \left(\frac{Y_i - d_i}{Y_i - 1}\right) d_i, \quad j = 1, \dots, K. \quad (2.4)$$

$$\sigma_{jg}(\tau) = -\sum_{i=1}^D [W(t_i)]^2 \frac{Y_{ij}}{Y_i} \frac{Y_{ig}}{Y_i} \left(\frac{Y_i - d_i}{Y_i - 1}\right) d_i, \quad j \neq g. \quad (2.5)$$

If the variance-covariance matrix of any selected $K - 1$ $Z_j(\tau)$'s is given by a $K - 1$ by $K - 1$ matrix Σ , the test statistic is given by the quadratic form

$$\chi^2 = [Z_1(\tau), Z_2(\tau), \dots, Z_{K-1}(\tau)]^T \Sigma^{-1} [Z_1(\tau), Z_2(\tau), \dots, Z_{K-1}(\tau)]. \quad (2.6)$$

When the null hypothesis is true, this statistic has a chi-squared distribution, for large samples with $K - 1$ degrees of freedom. An α level test of null hypothesis rejects when χ^2 is larger than α th upper percentage point of a chi-squared distribution with $K - 1$ degrees of freedom. Specially, when $K = 2$ the test statistic Eq. (2.6) can be reduced as

$$Z = \frac{\sum_{i=1}^D W(t_i) [d_{i1} - Y_{i1} (d_i / Y_i)]}{\sqrt{\sum_{i=1}^D [W(t_i)]^2 (Y_{i1} / Y_i) (1 - Y_{i1} / Y_i) [(Y_i - d_i) / (Y_i - 1)] d_i}}, \quad (2.7)$$

which has a standard normal distribution for large samples when the null hypothesis is true.

The derivation shown above (Klein and Moeschberger, 2003) is in fact consistent with the score test from the partial likelihood under the Cox model (Cox, 1975)

$$L = \prod_{i \in D} \frac{e^{\beta x_{ii}}}{\sum_{k \in R_i} e^{\beta x_{ki}}}, \quad (2.8)$$

where D represents the total number of events and R represents the total number of objects at risk at time of the i^{th} event. The log-likelihood is given by

$$LL(\beta) = \sum_{i=1}^D \beta x_{ii} - \sum_{i=1}^D \ln \left[\sum_{k \in R} e^{\beta x_{ki}} \right]. \quad (2.9)$$

The partial maximum likelihood estimates are found by maximizing Eq. (2.9). The score equations $U(\beta)$ are found by taking partial derivatives of Eq. (2.9) with respect to β . The information matrix $I(\beta)$ can be obtained by finding the second derivatives of the log partial likelihood. If the maximum likelihood estimate of β is $\hat{\beta}$, then the scores test given by

$$\chi^2 = U(\hat{\beta})^t I^{-1}(\hat{\beta}) U(\hat{\beta}), \quad (2.10)$$

which corresponds to the test statistic given by Eq. (2.6).

For two sample comparison, various test methods can be obtained when various weight functions are employed in test Eq. (2.7). A common weight function, leading to a test available in most statistical packages, is $W(t_i) = 1$. This choice of weight function leads to the simple log-rank test. It has the optimal power to detect the difference between two groups of survival distributions when the proportional hazards assumption holds. A second choice of weights is $W(t_i) = Y_i$. This weight function yields Gehan's (Gehan, 1965) generalization of the two-sample test. Another weighting function is given by

$$W(t) = \tilde{S}(t) = \prod_{t_i < t} \left(1 - \frac{d_i}{Y_i + 1} \right) \quad (2.11)$$

(Peto and Peto, 1972; Kalbfleisch and Prentice, 1980). Andersen et al. (Andersen et al., 1982) suggested that this weight function should be modified slightly as

$$W(t) = \tilde{S}(t) \frac{Y_i}{Y_i + 1}. \quad (2.12)$$

The Gehan's weight function $W(t_i) = Y_i$ depends on the event times and censoring distributions. Andersen's weight functions depend on the survival function of the combined data. Therefore, these Wilcoxon type tests can have misleading results when the censoring patterns are different in each sample (Prentice and Marek, 1979).

Fleming and Harrington (Fleming and Harrington, 1981) proposed a very general class of test that includes, as special cases, the log-rank test and a version of Wilcoxon test which is very close to that suggested by Peto and Peto (Peto and Peto, 1972). Here, we denote $\hat{S}(t)$ as the Kaplan-Meier's product-limit estimator [Eq. (1.11)] of the survival function based on the combined data. The weight function of Harrington-Fleming's test (Klein and Moeschberger, 2003) is given by

$$W_{p,q}(t_i) = [\hat{S}(t_{i-1})]^p [1 - \hat{S}(t_{i-1})]^q \quad (2.13)$$

with $p \geq 0, q \geq 0$. In Eq. (2.13), the survival function at the previous death time $\hat{S}(t_{i-1})$ is used as a weight to ensure that these weights are known just prior to the time at which the comparison is to be made. The simple log-rank test is obtained when $p = q = 0$. When $p = 1$ and $q = 0$ a version of the Wilcoxon test is obtained. When $p > 0$ and $q = 0$, the weight function gives the most weight to early departures between the hazard rates. When $p = 0$ and $q > 0$ the weight function gives the most weight to late departures between the hazard rates. By tuning the value of p and q appropriately, one can construct tests which have the most power against alternatives which have the hazard rates differing over any desired region.

Gill and Schumacher (Gill and Schumacher, 1987) proposed a test method for proportional hazards assumption for two-sample censored data based on comparison of generalized rank estimators of the relative risk. They also proposed a related graphical method and gave recommendations for the choice of appropriate weight functions. Sengupta et al. (Sengupta et al., 1998) proposed a graphical method on testing for the proportionality of hazards in two samples against the increasing cumulative hazard ratio alternative.

Oakes and Jeong (Oakes and Jeong, 1998) proposed a new class of weighted log-rank test for nonproportional hazards data by assuming various distributions for an omitted covariate from the proportional hazards model, which introduces nonproportionality. Their work established the connections between the theory of weighted log-rank test and of frailty models by assuming popular distributions in frailty theory for an exponentiated covariate in the Cox model. Although the method was derived from fully parametric models, the results extend to nonparametric tests.

In this work, we applied the weighted log-rank test by Oakes and Jeong to test the difference of two groups of survival data with various types of departures. We show that the nonparametric test based on one of the weighted log-rank test by Oakes and Jeong performs better than the currently available log-rank tests such as simple, Wilcoxon, or G-rho tests by Harrington and Fleming for the early difference cases.

2.2. Proportional Odds Modeling

Several models for analyzing proportional odds survival time have been developed (Bennett, 1983a, 1983b; Pettitt, 1984; Cheng et al., 1995; Murphy et al., 1997; Lam et al., 2002). McCullagh (McCullagh, 1980) first generalized the idea of constant odds ratio to more than two samples and fitted the model by maximum likelihood. But his model was mainly for social sciences and the censorship was not included in it. Bennett extended McCullagh's work for treating medical survival data by including the censorships (Bennett, 1984a). If the baseline survival function is $S_0(t)$, the survival odds ratio of object i to the baseline is defined as

$$\psi_i = \frac{\phi_i}{\phi_0} = \frac{S(t_i)[1 - S_0(t)]}{[1 - S(t_i)]S_0(t)}. \quad (2.14)$$

Let $\psi_i = e^{\beta x_i}$, then we can express the survival function of object i , S_i , as a function of parameter β and survival time t_i . The likelihood of a set of U of uncensored observations and a set C of right-censored observation for parametric inference is

$$L(\beta) = \prod_{i \in U} f(t_i; \beta) \prod_{i \in C} S(t_i; \beta), \quad (2.15)$$

where f is the appropriate probability density function of uncensored survival time T .

To fit the proportional odds model to data it is necessary to estimate the parameter β and the underlying distribution function f for parametric inference. This is similar to the proportional hazards model. In proportional hazards model, the hazard function is cancelled out in the partial likelihood method. For proportional odds model, however, the full unconditional likelihood must be used. Bennett (1984a) used the proportional odds model for nonparametric estimation of the survival function in the regression fitting. Bennett (1984b) also extended his nonparametric method to a parametric analysis of survival data following the log-logistic and other distributions.

Pettitt (Pettitt, 1984) used Bennett's proportional odds model but replaced the survival times in the model by their ranks. By comparing his method with that of Bennett, he pointed out several advantages of the rank method over Bennett's general method: (1) the rank method is more computationally efficient; (2) the rank method is less sensitively affected by extreme observations; and (3) the statistics used to test the null hypothesis of $\beta = 0$ is distribution free; thus, the rank test for $\beta = 0$ is valid whether or not the proportional odds model is true.

Murphy et al. (1997) demonstrated that a semiparametric maximum likelihood estimation method can be used to estimate the regression coefficients of the proportional odds model

efficiently. They proposed a profile likelihood function based on the generalization of Bennett (1984a)

$$L(H_0, \beta) = \prod_{i=1}^n \left[\frac{dH_0(t) / dt}{e^{-\beta x_i} + H_0(t)} \right]^{\delta_i} \frac{e^{-\beta x_i}}{e^{-\beta x_i} + H_0(t)}, \quad (2.16)$$

where δ_i is the censoring indicator, parameter $H_0(t) = \frac{1 - S_0(t)}{S_0(t)} = \frac{1}{\phi_0(t)}$ is the baseline failure odds (inverse of survival odds). Murphy et al. (1997) proved that the maximum likelihood estimator of parameter β is consistent, asymptotically normal, and efficient. Differentiation of the derived profile likelihood function yields consistent estimators of the efficient information matrix.

Lam et al. (2002) proposed a random effect semiparametric proportional odds model. They used a unified estimation procedure to estimate the regression and dependence parameters simultaneously by means of a marginal rank likelihood approach which is approximated by Monte Carlo method. Unlike the fully parametric method, their regression parameter estimate is not sensitive to the choice of correlation structure of the random effects. With the assumption of a baseline log-odds function, Lam et al. demonstrated that the proposed class of semiparametric proportional odds models serves as a good alternative to the random effect proportional hazards model.

Dauxois and Kirmani (Dauxois and Kirmani, 2003) developed a nonparametric statistic for testing the proportional odds assumption over survival times under random censoring. It is a formal analytical test of the proportional odds assumption. The test they proposed has the form of

$$\gamma(W) = \iint_{\tau_1 < s < t < \tau_2} W(s, t) [\phi_1(t)\phi_2(s) - \phi_2(t)\phi_1(s)] ds dt, \quad (2.17)$$

where $W(s, t)$ is a weight function, which is dependent on the survival function of samples to be compared. Some of the weight functions recommended include (Gill and Schumacher, 1987; Sengupta et al. 1998)

$$\begin{aligned}
 W_a(t) &= Y_1(t)Y_2(t) \\
 W_b(t) &= \frac{Y_1(t)Y_2(t)}{Y_1(t) + Y_2(t)} \\
 W_c(t) &= \frac{Y_1(t)Y_2(t)}{Y_1(t) + Y_2(t)} \hat{S}(t) \\
 W_d(t) &= \frac{Y_1(t)Y_2(t)}{Y_1(t) + Y_2(t)} [\hat{S}(t)]^{1/2}
 \end{aligned}$$

Weight functions $W_a(t)$ and $W_b(t)$ are adopted by Dauxois and Kirmani (Dauxois and Kirmani, 2003) in their test for nonproportional odds assumption.

It has been proved that the test statistics in Eq. (2.17) is asymptotically normal under the null hypothesis of proportional odds. In the numerical calculations, the survival functions can be approximated by their Kaplan-Meier estimators. Monte Carlo simulations in the context of the log-logistic survival data indicate that the proposed model works well in testing the proportionality of odds for survival data with random censoring.

In this work, we extend the model by Dauxois and Kirmani to parametric cases by supplementing the distribution information of the survival data into the model. We have performed Monte Carlo simulations in the frame of survival data following the log-logistic distribution, and compared the modified model with the nonparametric model by Dauxois and Kirmani. Our results indicate that the parametric method developed is more sensitive in testing the validity of the proportional odds model assumption than the existing method.

3. TEST FOR PROPORTIONAL HAZARDS

We have studied the test statistic for detecting differences in nonproportional hazards survival data with random censorship using log-rank test supplemented with a newly developed weight function. In this chapter we present the results obtained from this study. We used the modified weighted log-rank test by using a new weight function to account for the random effect that disrupts the proportionality of the hazards functions. We used Monte Carlo simulations to generate survival data with an exponential distribution and found that the modified method is superior in detecting the early difference of data than the conventional log-rank test and the G-rho tests of Harrington and Fleming.

3.1. Weighted Log-rank Test

In the analysis of survival data, we often encounter the situation where the survival times of a group of individuals are not independent. Such correlations among survival times may arise when different individuals share some feature in common. For example, the survival data from the same clinic may be more similar than those from another clinic. This could be due to different treat teams in different clinics. Such random effects that can cause dependence in survival data are often referred to as frailties.

Frailty in survival data may complicate survival analysis. The efficiency of a test statistic for survival data may decrease if the frailty factor is not considered, due to the nonproportionality

caused by frailty (Oakes and Jeong, 1998). In addition, failure to include frailty in a test may result in the misspecification of the proportional hazards model (Oakes and Jeong, 1998). Some methods have been proposed to attack this problem. One of widely used methods is to include a frailty model explicitly into the proportional hazards method (Aalen, 1998) to account for the nonproportionality in hazards function caused by random effects. In other words, an extra term corresponding to frailty is introduced into the standard proportional hazards model. For example, if we denote an unobserved random effect for object i by a covariate z_i , then Eq. (2.1) becomes

$$h_i(t) = e^{\beta x_i + \gamma z_i} b_0(t), \quad (3.1)$$

where $b_0(t)$ is an unknown baseline hazard function. Comparing Eq. (2.1) and Eq. (3.1), we can see that we actually introduced a prefactor function $e^{\gamma z_i}$ to the proportional hazards model in order to incorporate the frailty term. An optimal weighting function can be derived if a distribution is assumed for the frailty (Oakes and Jeong, 1998).

Using a weighted log-rank test is important in order to account for the nonproportionality in the data. This is due to the fact that the loss of the efficiency of the test from omitting a covariate is generally more important than the additional loss of the efficiency due to the resulting misspecification of the proportional hazards model (Oakes and Jeong, 1998).

3.2. Derivation of the New Weighting Function

The proportional hazards model incorporated with a randomization variable z_i is written as

$$h_i(t | x_i, z_i) = \exp(\beta x_i + \gamma z_i) b_0(t), \quad (3.2)$$

where $b_0(t)$ is a unknown baseline hazard function. If the values of z_i are observed, then the score test (simple log-rank test) from Cox's model with γ estimated as a nuisance parameter is optimal for testing $\beta = 0$ with γ unspecified. If the values of z are not observed, we need to develop a parametric analog (weighted log-rank test) to account for the omitted covariate z .

We write the unobserved factor

$$w_i = \exp(\gamma z_i)$$

and

$$\theta_i = \exp(\beta x_i),$$

so the null hypothesis $\beta = 0$ is equivalent to $\theta_i = 1$. Eq. (3.2) becomes

$$h_i(t | x_i, w_i) = w_i \theta_i b_0(t). \quad (3.3)$$

In terms of the corresponding survival functions we get

$$S_i(t | x_i, w_i) = \exp\left[-\int_0^t h_i(u) du\right], \quad (3.4)$$

and the baseline survival function is

$$A_0(t) = \exp\left[-\int_0^t b_0(u) du\right]. \quad (3.5)$$

So from Eq. (3.3) we obtain

$$S_i(t | x_i, z_i) = [A_0(t)]^{w_i \theta_i}. \quad (3.6)$$

We assume that w_i are realizations of identical independently distributed positive random variables W_i , the Laplace transform of W is defined as

$$p(s) = \int_0^\infty f(W) e^{-sW} dW = E[e^{-sW}]. \quad (3.7)$$

Removing the conditioning on the variable W_i by integrating over W , Eq. (3.6) becomes

$$\begin{aligned}
S_i(t) &= \int_0^\infty f(W)[A_0(t)]^{W\theta_i} dW \\
&= \int_0^\infty f(W) \exp[\log A_0(t)^{W\theta_i}] dW \\
&= \int_0^\infty f(W) \exp[W\theta_i \log A_0(t)] dW \\
&= \int_0^\infty f(W) e^{-W\theta_i \{-\log A_0(t)\}} dW
\end{aligned} \tag{3.8}$$

Let

$$B = B(t) = -\log A_0(t) = \int_0^t b_0(u) du, \tag{3.9}$$

Then, by comparing Eq. (3.8) with Eq. (3.7) we obtain

$$S_i(t) = E[e^{-W\theta_i \{-\log A_0(t)\}}] = E[e^{-W\theta_i B}] = p(\theta_i B). \tag{3.10}$$

The nonparametric simple log-rank test statistic, $U_3^{(np)}$, can be calculated by setting $\beta = 0$ in the score statistic of Cox's partial likelihood (Cox, 1972, 1975),

$$U_3^{(np)}(\beta) = \sum \int \left[x_i - \frac{\sum x_j \theta_j Y_j(t)}{\sum \theta_j Y_j(t)} \right] dN_i(t), \tag{3.11}$$

which gives

$$U_3^{(np)}(0) = \sum \int \left[x_i - \frac{\sum x_j Y_j(t)}{\sum Y_j(t)} \right] dN_i(t), \tag{3.12}$$

since $\beta = 0$ gives $\theta_i = 1$. When the covariate z_i are not observed, a wider class of test statistics can be defined by introducing a weight function $\hat{l}(t)$ into the log-rank statistic, giving

$$U_3^{(np)}(0) = \sum \int \hat{l}(t) \left[x_i - \frac{\sum x_j Y_j(t)}{\sum Y_j(t)} \right] dN_i(t). \tag{3.13}$$

Gill (Gill, 1980) showed that a weighted log-rank test of form given in Eq. (3.13) is optimal provided that the weight function $\hat{l}(t)$ converges to a deterministic limit $l(t)$ proportional to

$$\lambda(t) = \left. \frac{\partial \log h_\theta(t)}{\partial \theta} \right|_{\theta=1}. \quad (3.14)$$

We know that (Collett, 2003)

$$h_\theta(t) = -\frac{\partial \log S_\theta(t)}{\partial t}. \quad (3.15)$$

Combining Eqs. (3.15) and (3.10) we have

$$h_\theta(t) = -\frac{\partial \log p(\theta B)}{\partial t}. \quad (3.16)$$

Let $\theta B = \Delta$, note that B is only dependent on t , so we have

$$\frac{\partial \Delta}{\partial t} = \frac{\partial(\theta B)}{\partial t} = \theta \frac{\partial B}{\partial t} = \theta B'_t, \text{ and } \frac{\partial \Delta}{\partial \theta} = \frac{\partial(\theta B)}{\partial \theta} = B.$$

And let $p(\Delta) = p$, $\frac{\partial p(\Delta)}{\partial \Delta} = p'$, $\frac{\partial p'}{\partial \Delta} = p''$, then Eq. (3.16) becomes

$$h_\theta(t) = -\frac{\partial \log p(\Delta)}{\partial t} = -\frac{1}{p(\Delta)} \frac{\partial p}{\partial \Delta} \frac{\partial \Delta}{\partial t} = -\frac{\theta B'_t p'}{p}. \quad (3.17)$$

So

$$\begin{aligned} \frac{\partial \log h_\theta(t)}{\partial \theta} &= \frac{\partial}{\partial \theta} \left[\log \left(-\frac{\theta B'_t p'}{p} \right) \right] = \frac{\partial}{\partial \theta} [\log(-\theta B'_t p') - \log p] \\ &= \frac{1}{-\theta B'_t p'} \frac{\partial(-\theta B'_t p')}{\partial \theta} - \frac{1}{p} \frac{\partial p}{\partial \theta} \\ &= \frac{B'_t}{\theta B'_t p'} \frac{\partial(\theta p')}{\partial \theta} - \frac{1}{p} \frac{\partial p}{\partial \Delta} \frac{\partial \Delta}{\partial \theta} \\ &= \frac{1}{\theta p'} \left(\frac{\theta \partial p'}{\partial \theta} + p' \frac{\partial \theta}{\partial \theta} \right) - \frac{p'}{p} B \\ &= \frac{1}{p'} \frac{\partial p'}{\partial \Delta} \frac{\partial \Delta}{\partial \theta} + \frac{1}{\theta} - \frac{B p'}{p} \\ &= \frac{B p''}{p'} + \frac{1}{\theta} - \frac{B p'}{p} \end{aligned} \quad (3.18)$$

At $\theta = 1$, $\Delta = B\theta = B$, $p(\Delta) = p(B)$, $\frac{\partial p(\Delta)}{\partial \Delta} = \frac{\partial p(B)}{\partial B} = p'(B)$, $\frac{\partial p'}{\partial \Delta} = p''(B)$, so that Eq.

(3.18) is

$$\lambda(t) = \left. \frac{\partial \log h_\theta(t)}{\partial \theta} \right|_{\theta=1} = \frac{Bp''(B)}{p'(B)} + 1 - \frac{Bp'(B)}{p(B)}. \quad (3.19)$$

If the frailty, W , follows a gamma distribution with index κ , then we know the Laplace transformation of it is given by (Oakes and Jeong, 1998) as

$$p(B) = (1 + B)^{-\kappa}, \quad (3.20)$$

We have

$$p'(B) = -\kappa(1 + B)^{-\kappa-1} = -\frac{\kappa}{1 + B}p(B), \quad (3.21)$$

$$p''(B) = \kappa(\kappa + 1)(1 + B)^{-\kappa-2} = -\frac{\kappa + 1}{1 + B}p'(B). \quad (3.22)$$

So

$$\lambda(t) = 1 + \frac{Bp''(B)}{p'(B)} - \frac{Bp'(B)}{p(B)} = 1 - B\frac{\kappa + 1}{1 + B} + B\frac{\kappa}{1 + B} = \frac{1}{1 + B}. \quad (3.23)$$

From Eq. (3.20) we have

$$B = \left[\frac{1}{p} \right]^{\frac{1}{\kappa}} - 1, \quad (3.24)$$

So

$$\lambda(p) = \frac{1}{1 + (1/p)^{1/\kappa} - 1} = p^{1/\kappa}. \quad (3.25)$$

At $\theta = 1$ from Eq. (3.10) we know

$$S(t) = p(B) = p. \quad (3.26)$$

So the optimal weight function for gamma frailty is

$$\lambda[S(t)] = [S(t)]^{1/\kappa} = [S(t)]^\rho. \quad (3.27)$$

If the frailty W follows an inverse Gaussian distribution with unit mean and variance $(2\psi)^{-1}$, then we know the Laplace transformation is (Oakes and Jeong, 1998)

$$p(B) = e^{-2[\psi(\psi+B)]^{1/2}+2\psi}, \quad (3.28)$$

We have

$$p'(B) = e^{-2[\psi(\psi+B)]^{1/2}+2\psi}[-\psi^{1/2}(\psi+B)^{-1/2}] = -p(B)\left[\frac{\psi}{\psi+B}\right]^{1/2}, \quad (3.29)$$

$$p''(B) = -p'(B)\left[\frac{\psi}{\psi+B}\right]^{1/2} + p(B)\left[\frac{1}{2}\psi^{1/2}(\psi+B)^{-3/2}\right]. \quad (3.30)$$

Eq. (3.29) also gives

$$p(B) = -p'(B)\left[\frac{\psi}{\psi+B}\right]^{-1/2},$$

So Eq. (3.30) becomes

$$\begin{aligned} p''(B) &= -p'(B)\left[\frac{\psi}{\psi+B}\right]^{1/2} + p'(B)\left[\frac{\psi}{\psi+B}\right]^{-1/2}\left[\frac{1}{2}\psi^{1/2}(\psi+B)^{-3/2}\right] \\ &= p'(B)\left[-\left(\frac{\psi}{\psi+B}\right)^{1/2} - \frac{1}{2}\left(\frac{1}{\psi+B}\right)\right]. \end{aligned} \quad (3.31)$$

So the optimal weight function is given by

$$\begin{aligned} \lambda(t) &= 1 + \frac{Bp''(B)}{p'(B)} - \frac{Bp'(B)}{p(B)} \\ &= 1 + B\left[-\left(\frac{\psi}{\psi+B}\right)^{1/2} - \frac{1}{2}\left(\frac{1}{\psi+B}\right)\right] + B\left[\frac{\psi}{\psi+B}\right]^{1/2}. \\ &= 1 - \frac{B}{2(\psi+B)} \end{aligned} \quad (3.32)$$

From Eq. (3.28) we obtain

$$B + \psi = \frac{\left(\psi - \frac{1}{2} \log p\right)^2}{\psi}, \quad (3.33)$$

$$B = \frac{\left(\psi - \frac{1}{2} \log p\right)^2}{\psi} - \psi, \quad (3.34)$$

and

$$\frac{B}{(\psi + B)} = \frac{\frac{\left(\psi - \frac{1}{2} \log p\right)^2}{\psi} - \psi}{\frac{\left(\psi - \frac{1}{2} \log p\right)^2}{\psi}} = 1 - \frac{\psi^2}{\left(\psi - \frac{1}{2} \log p\right)^2}$$

So Eq. (3.32) becomes

$$\lambda(p) = 1 - \frac{1}{2} + \frac{\psi^2}{2\left(\psi - \frac{1}{2} \log p\right)^2} = \frac{1}{2} + \frac{2\psi^2}{(2\psi - \log p)^2} = \frac{1}{2} + \frac{2\psi^2}{[2\psi - \log S(t)]^2}. \quad (3.35)$$

For survival distributions with nonproportional hazards, a weighted log-rank test with an optimal weight function is expected to be more sensitive than the simple log-rank test (Oakes and Jeong, 1998). For example, when the frailty has a gamma distribution with an index parameter κ , from Eq. (3.27) we obtain the optimal weight function

$$W(t) = S(t)^{\frac{1}{\kappa}}, \quad (3.36)$$

which corresponds to the ‘‘G-rho’’ tests of Harrington and Fleming (Fleming and Harrington, 1991) with $\text{rho}=1/\kappa$. When $\text{rho}=0$, the G-rho test reduces to the simple log-rank test. When $\text{rho}=1$, the G-rho test reduces to Wilcoxon test (Collett, 2003).

However, when the frailty distribution affects the proportionality of the hazards, the simple log-rank test is no longer the optimal test, so a weighted log-rank test of G-rho type or different

weighting function must be used. For example, when the frailty follows an inverse Gaussian distribution, an optimal weighting function is given as Eq. (3.35)

$$W(t) = \frac{1}{2} + \frac{2\psi^2}{[2\psi - \log \hat{S}(t)]^2}, \quad (3.37)$$

where ψ is a controlling parameter which can take the value of 0 to $+\infty$, $\hat{S}(t)$ is the estimated common survival function based on the combined sample up to t (Oakes and Jeong, 1998).

The Gamma and inverse Gaussian are two most commonly seen frailty models in modeling survival data using proportional hazards methods. It is known that the inverse Gaussian frailty makes the population homogeneous with time, whereas for the gamma frailty the relative heterogeneity is constant (Hougaard, 1984). In this study we are particularly interested in testing nonproportional hazards survival data that become more and more homogeneous as time proceeds. Therefore we chose the weighting function corresponding to the inverse Gaussian frailty [Eq. (3.37)] in our simulations. For observed survival data, the test statistic is given by

$$\Delta = \sum_{i \in D} W(t_i) \left[d_{i1} - Y_{i1} \left(\frac{d_{i1} + d_{i2}}{Y_{i1} + Y_{i2}} \right) \right], \quad (3.38)$$

where $W(t_i)$ is a common weighting function shared by each group, Y_{i1} and Y_{i2} are the number of objects at risk in group I and II at time t_i , d_{i1} and d_{i2} are the number of events occurred in each group at time t_i , respectively. The summation is over D , which includes a subset of survival times that are observed as event of interest. The variance of W can be estimated by

$$V = \sum_{i \in D} [W(t_i)]^2 \frac{Y_{i1}}{Y_i} \left(1 - \frac{Y_{i1}}{Y_i} \right) \left(\frac{Y_i - d_i}{Y_i - 1} \right) d_i. \quad (3.39)$$

It can be proved that $Z = \frac{\Delta}{\sqrt{V}}$ has an asymptotically standard normal distribution (Harrington and Fleming, 1982).

The common survival function estimator $\hat{S}(t)$ in Eq. (3.37) is given by

$$\hat{S}(t) = \prod_{t \leq t_i} \left(1 - \frac{d_i}{Y_i} \right).$$

A frailty distribution is usually unobservable, thus we do not know if the frailty itself will affect proportionality of the survival data at hand. So we must test if the observed survival data still follows proportional hazards assumption before we decide what type of weighting function should be used (Therneau and Grambsch, 2000). Testing the proportionality in survival data can be performed by using the `cox.zph` procedure provided in S-Plus. If the `cox.zph` test indicates proportionality in the data, the log-rank test statistics like simple log-rank test and Wilcoxon log-rank test can be chosen. However, if the `cox.zph` test shows that the dataset does not satisfy the proportional hazards assumption, we should use a log-rank test with a different type of weighting function, such as the one for the survival data with inverse Gaussian frailty or the Harrington-Fleming test.

3.3. Test of Frailty Distributions

A specific frailty distribution has to be assumed in order to select the appropriate weight function used in the proposed weighted log-rank test. In this work, we assume that the frailty follows either the Gamma (Vanpel et al., 1979) or Inverse Gaussian (Hougaard, 1984) distributions, which are two most commonly used distributions in proportional hazards frailty models.

Economou and Caroni (2005) proposed a graphical method to test the assumption of Gamma or Inverse Gaussian frailty distributions in survival data. The method is based on the non-

parametric Kaplan-Meier estimate of the survivor function. If the baseline hazard function is assumed to be Weibull with scale parameter ξ and shape parameter β , conditional on frailty z , the random variable T of time-to-event has survival function in the form of

$$S(t | z) = e^{-z(t/\xi)^\beta}.$$

If Z is a random variable with distribution function G on $(0, \infty)$, then the unconditional survival function is given by

$$S(t) = \int_0^\infty e^{-z(t/\xi)^\beta} dG(z). \quad (3.40)$$

In the proportional hazards frailty model, if G is chosen as the Gamma distribution, both the shape parameter and scale parameter are commonly chosen as equal to ν (and hence with mean equal to unity) in order to remove one parameter to avoid a problem of identifiability (Economou and Caroni, 2005). Then the probability density function of Gamma distribution is given by

$$f(z; \nu, \nu) = \frac{\nu^\nu}{\Gamma(\nu)} z^{\nu-1} e^{-\nu z}.$$

Integration of Eq. (3.40) we have

$$S(t) = \left[1 + \frac{(t/\xi)^\beta}{\nu} \right]^{-\nu}. \quad (3.41)$$

Eq. (3.41) corresponds to the survival function of the univariate Burr distribution (Burr, 1942).

Similarly, if G is chosen as the Inverse Gaussian distribution with shape parameter λ and scale parameter 1, then Eq. (3.40) becomes

$$S(t) = \exp \left[\lambda \left(1 - \sqrt{\frac{2(t/\xi)^\beta}{\lambda} + 1} \right) \right]. \quad (3.42)$$

Taking logarithms of the relation Eq. (3.41) we obtain

$$\log S(t) = -\nu \log \left[1 + \frac{(t/\xi)^\beta}{\nu} \right]. \quad (3.43)$$

From the approximation $\log(1+x) \approx \log x$, it follows that

$$\log S(t) \approx -\nu\beta \log t + \nu \log \nu \xi^\beta,$$

when $\frac{(t/\xi)^\beta}{\nu}$ is large. Consequently, a plot of $-\log \hat{S}(t)$ vs. $\log t$ should be a straight line,

where $\hat{S}(t)$ is a nonparametric estimator of the survival function, such as the Kaplan-Meier. The same plot can be derived for survival function following the exponential distribution since the exponential is a special case of the Weibull.

Unfortunately, there are some observations (early failures) which result in the nonlinear behavior of $-\log \hat{S}(t)$ vs. $\log t$ (when $\frac{(t/\xi)^\beta}{\nu}$ is not large enough). Then these early failure data give the plot a characteristic horizontal section for small t (see Figure 1 for examples). The length of this horizontal section depends on the parameter ν . As $\nu \rightarrow 0$, that is, there is a high degree of heterogeneity, it almost disappear since $\frac{(t/\xi)^\beta}{\nu}$ is large for all t .

Likewise, taking logarithms of Eq. (3.42) we obtain

$$\log S(t) = \lambda \left(1 - \sqrt{\frac{2(t/\xi)^\beta}{\lambda} + 1} \right). \quad (3.44)$$

For large λ , $\sqrt{\frac{2(t/\xi)^\beta}{\lambda} + 1} \approx \frac{(t/\xi)^\beta}{\lambda} + 1$, we have $\log S(t) \approx -(t/\xi)^\beta$. Plot of $\log[-\log \hat{S}(t)]$

vs. $\log t$ giving a straight line with slope of β . Then the the Weibull-Inverse Gaussian model reduces to the initial Weibull model with negligible frailty, since there is only a small degree of heterogeneity when λ is large.

For very small λ , which corresponds to high degree of heterogeneity, we have

$$1 - \sqrt{\frac{2(t/\xi)^\beta}{\lambda} + 1} \approx -\sqrt{\frac{2(t/\xi)^\beta}{\lambda}},$$

and Eq. (3.44) is $\log S(t) \approx -\sqrt{2\lambda(t/\xi)^\beta} = -[(2\lambda)^{1/\beta} t/\xi]^{\beta/2}$. Plot of $\log[-\log \hat{S}(t)]$ vs. $\log t$ giving a straight line with slope of $\beta/2$. This corresponds to the Weibull model with scale parameter $\xi/(2\lambda)^{1/\beta}$ and shape parameter equal to $\beta/2$.

For intermediate values of λ , the initial section of the plot would behave like the no-frailty case with slope β , since the term $\frac{(t/\xi)^\beta}{\lambda}$ is still small for small t . For large t , the behavior should be like the case of small λ , with slope $\beta/2$. At the transition from slope β to $\beta/2$ as time increases, the plot will show some curvature. Examples are shown in Figure 2.

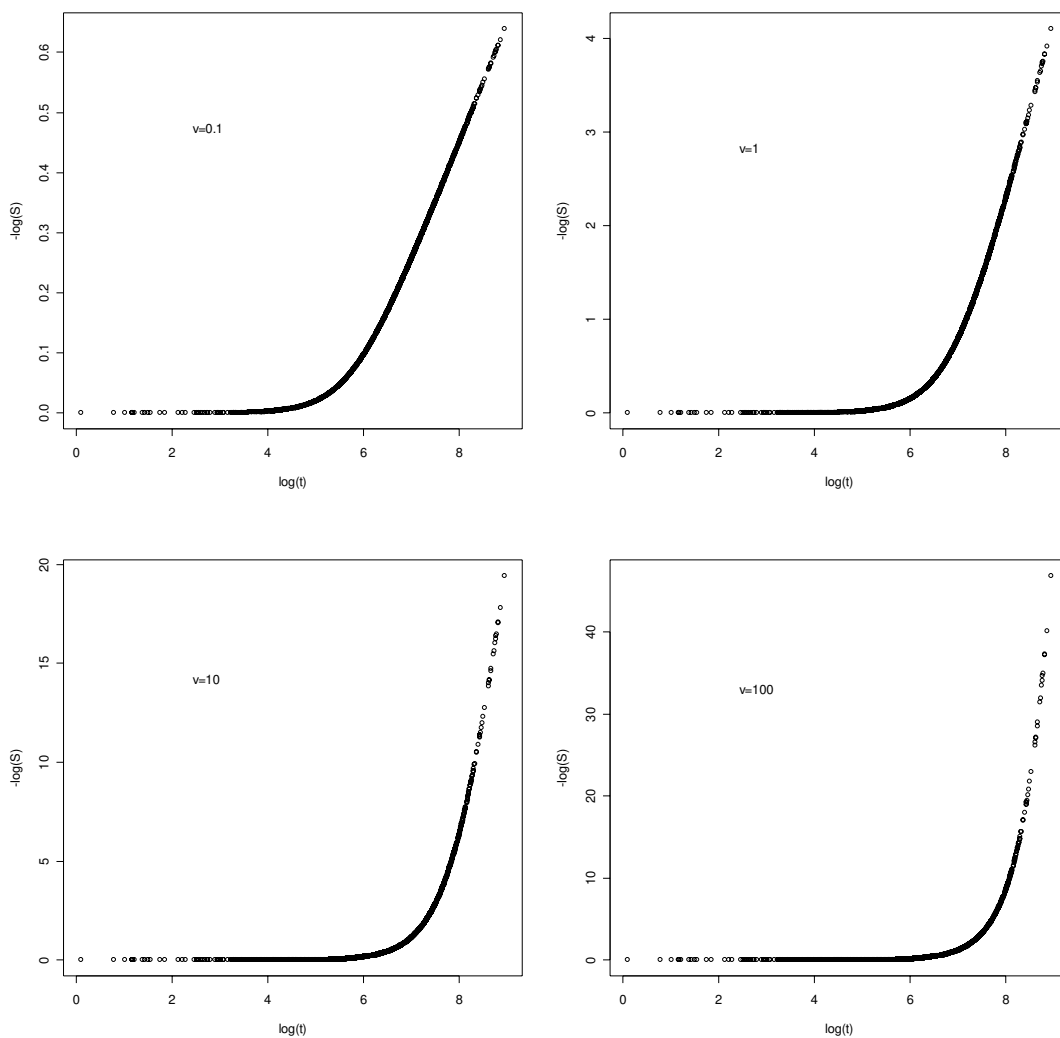


Figure 1. Diagnostic plots of the Weibull survival function with the Gamma frailty.

$-\log S(t)$ vs. $\log t$, for various values of the shape parameter ν for the Gamma frailty. The sample size is 2000, with $\xi = 1000$, and $\beta = 2$. The curve shows a concave shape when the frailty is a Gamma distribution. In particular, when ν is small, the non constant part of the curve is a straight line.

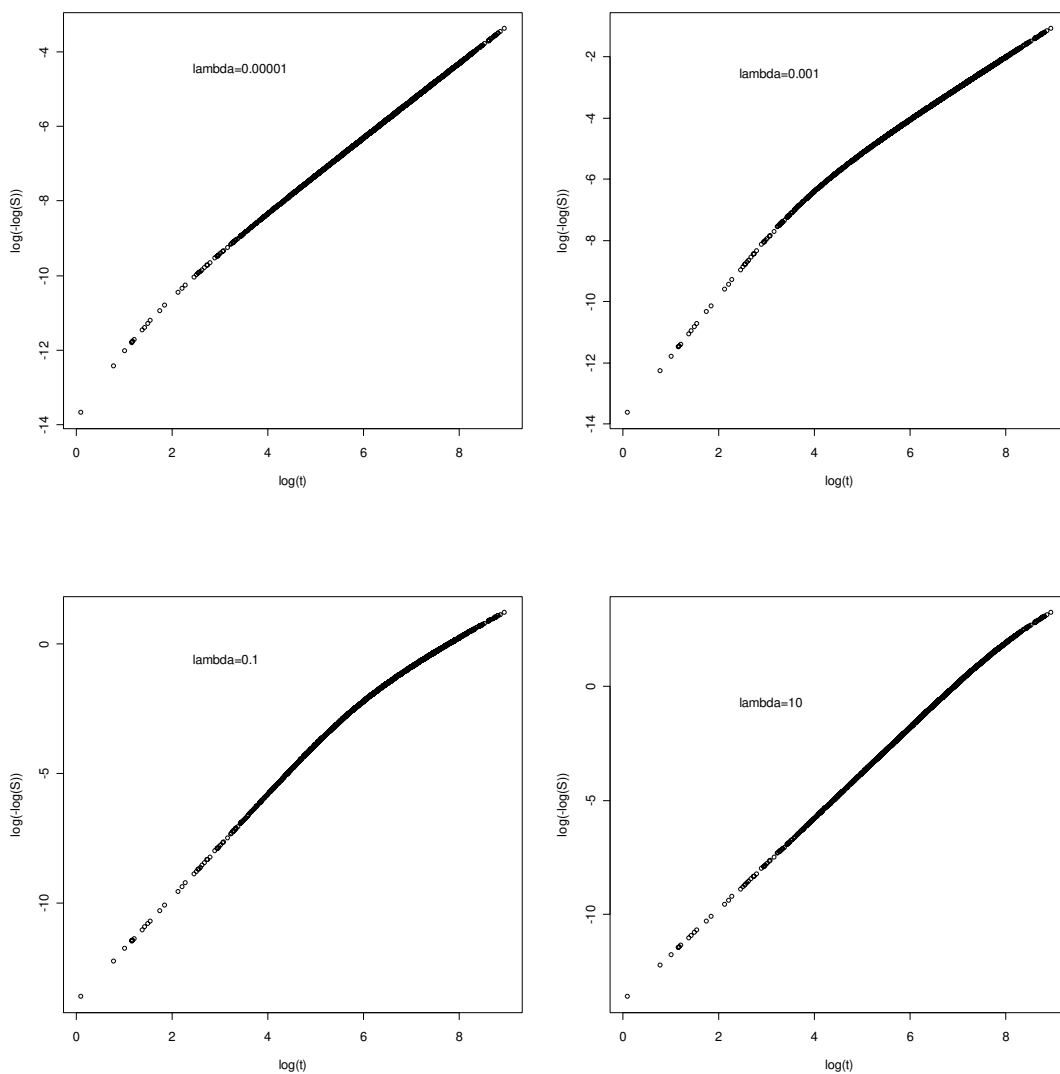


Figure 2. Diagnostic plots for the Weibull survival function with the Inverse Gaussian frailty. $\log[-\log S(t)]$ vs. $\log t$, for various values of the shape parameter λ for the Inverse Gaussian frailty. The sample size is 2000, with $\xi = 1000$, and $\beta = 2$. The curve shows a shape of straight line, especially when λ is big.

3.4. Simulation Methodology

Suppose there are two groups of survival data with corresponding hazard functions $h_2(t)$ and $h_1(t)$. Survival data can be generated by Monte Carlo method according to the characteristics of h_1 and h_2 . Then we can use log-rank tests with different weight functions to determine the power of each test method in differentiating these two groups of data. An estimate of the statistical power of the test is provided by

$$power = \frac{m}{n_s}, \quad (3.45)$$

where m is the number of simulations in which the test can differentiate the data with significance, and n_s is the total number of simulations.

In our simulation, we take $h_1(t)$ as the baseline hazard function and set it to be a constant, ρ . Therefore h_2 becomes

$$h_2(t | z) = e^{\beta z} h_1(t) = e^{\beta z} \rho. \quad (3.46)$$

Here z is a covariate. The null hypothesis that h_1 and h_2 are identical corresponds to $\beta = 0$.

The survival functions are

$$S_1(t) = e^{-\rho t}, \quad (3.47)$$

and

$$S_2(t | z) = S_1(t)^{e^{\beta z}}. \quad (3.48)$$

Then for an object in group I, the probability that its survival time is less than value t is

$$F_1(t) = 1 - S_1(t) = 1 - e^{-\rho t}, \quad (3.49)$$

and likewise, for an object in group II,

$$F_2(t | z) = 1 - S_2(t) = 1 - e^{-(\rho t)e^{\beta z}}. \quad (3.50)$$

Since that $F_1(t)$ and $F_2(t | z)$ conform to a uniform distribution in the range of $[0,1]$, we have

$$F_1(t) = 1 - e^{-\rho t} = u, \quad (3.51)$$

and

$$F_2(t) = 1 - e^{-(\rho t)e^{\beta}} = u, \quad (3.52)$$

as $z = 1$ for group two data.

For survival data in group one, we obtain

$$t_1 = -\frac{\ln(1-u)}{\rho}. \quad (3.53)$$

For survival data in group two, we obtain

$$t_2 = -\frac{\ln(1-u)}{\rho e^{\beta}} = e^{-\beta} t_1. \quad (3.54)$$

Therefore, we can generate two groups of survival data conforming to h_1 and h_2 in Eq. (3.46) by starting from a uniform distribution u , and using the relationships given in Eqs. (3.53) and (3.54).

The simulation procedures are as follows:

- i. Generate N^* observations from uniform distribution $u(0,1)$, designate them as u_i , $i = 1, 2, \dots, N^*$.
- ii. Generate the survival times for group I, t_{1i} , $i = 1, 2, \dots, N^*$, base on Eq. (3.53). The parameter ρ in Eq. (3.53) is set arbitrarily as 0.001, 0.1, and 0.3, respectively. The data are sorted in ascending order.

$$t_{1i} = -\frac{\ln(1 - u_i)}{\rho}.$$

- iii. Generate the survival times for group II, t_{2i} . The random effect (frailty) of the survival times is planted into the t_{2i} by multiplying the factor of $e^{-\beta}$ by t_{1i} , as shown in Eq. (3.54). In our simulations, we let the premultiplier $e^{-\beta}$ be in the range of (0, 1). The values of $e^{-\beta}$ were chosen according to the shapes of hazards ratios of interest. For example, if we are interested in two groups of data with early difference, we let the $e^{-\beta}$ take values of piece-wise proportionality reflecting early departures.
- iv. Generate the censored data in two groups from a uniform distribution randomly and choose the minimum. In this study we let the censoring occur randomly.

In testing the proportionality of the simulated data, we used the `cox.zph` function in the S-Plus/R package. The null hypothesis for this test is that the data obey the assumption of proportional hazards.

We found it was very difficult to incorporate the new weighting function in Eq. (3.37) into the survival procedure in the R/S-Plus package. Therefore, we wrote our own program in R/S-Plus to evaluate Eq. (3.38) and (3.39), see attached programs in the Appendix. The survival data obtained were transformed accordingly in order to calculate the quantities in Eq. (3.38) and (3.39) numerically. The p-values were calculated corresponding to the observed statistic,

$Z = \frac{\Delta}{\sqrt{V}}$, which follow the standard normal distribution. The test results based on the statistics

in Eq. (3.38) and (3.39) with the new weight function Eq. (3.37) are compared with a simple log-rank test and Harrington-Fleming's weighted log-rank test.

3.5. Results and Discussion

3.5.1. Test for proportional hazard assumptions

The purpose of this work is to investigate how the simple log-rank test and the weighted log-rank test of Harrington-Fleming perform for the nonproportional hazards data, compared with the test with the new weight function in Eq. (3.37). Thus, as the first step it is worthwhile to evaluate how significantly the simulated data violate the proportional hazards assumption.

We used the `cox.zph` function in the S-Plus package to test the statistical significance of violation of the proportional hazards assumption in the simulated data. We tested simulated data with early, middle, and late departure. The datasets were generated by the procedures described in section 3.4 with baseline hazard function $\rho = 0.3$. Our results show that `cox.zph` tests do identify that the nonproportionality exists in our simulated data. We examined simulated datasets with a data size of 1500 and found that a large fraction of datasets to be nonproportional at the significance level of 0.05 detected by the `cox.zph` procedure. However, the level of nonproportionality varies from dataset to dataset. The most significant nonproportionality in hazards occurs in data with early departures. For example, for late difference datasets, 32% of 100 simulations are identified as nonproportional in hazards using the `cox.zph` procedure. For early difference datasets, about 94% of the datasets were identified as nonproportional in hazards. For middle difference, 57% of datasets were shown to be nonproportional.

In Figures 3 to 5 we show some examples for the estimated patterns of change of the hazard ratios from the `cox.zph` function (smoothed scaled Schoenfeld residual plots), together with the corresponding Kaplan-Meier plots. The shape of the Kaplan-Meier estimation of the survival

distributions clearly indicates the nonproportionality hazards nature of the simulated data. For example, Figure 3 shows the `cox.zph` regression for two survival datasets that have later difference. The data size is 100. It can be seen from the residual plots (graph in the top) that the residual curve drifts away from value of zero as time progresses, which indicates significant late difference in the two groups of survival time. The test of proportional hazards hypothesis of the data using `cox.zph` gives a p-value of 0.04. Similar procedures were applied to simulated samples with early and middle differences, two examples of which are shown in Figure 4 and 5. Tests of proportional hazards using the function `cox.zph` of these two sets of survival data give p-value of 0.03 and 0.03, respectively. Therefore, it is evident that significant nonproportionality exists in the hazards function of the datasets generated in our simulations. These datasets would be appropriate samples that can be used for the evaluation of our method.

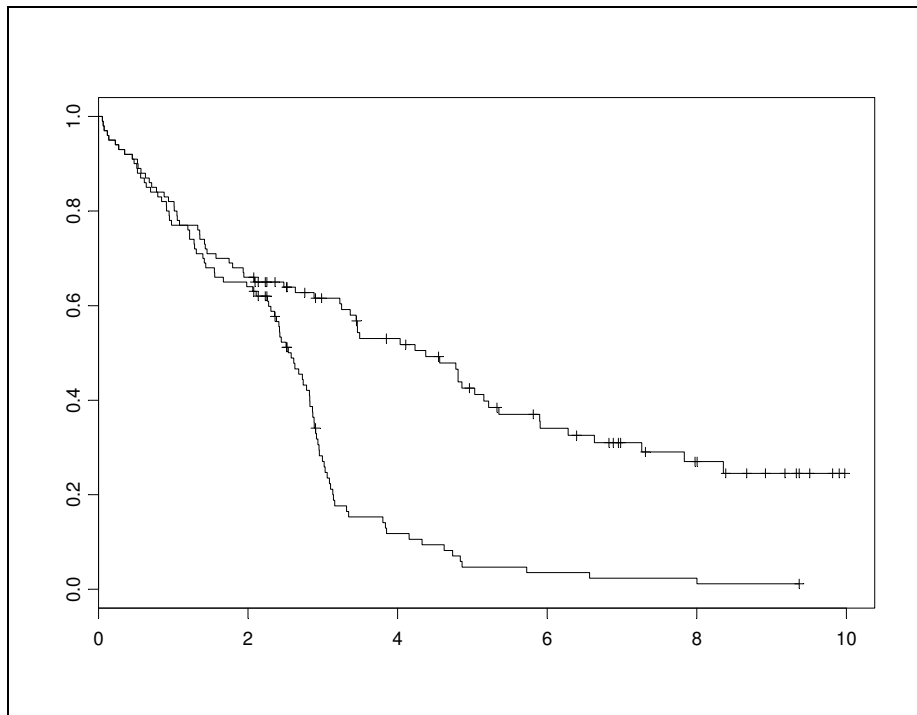
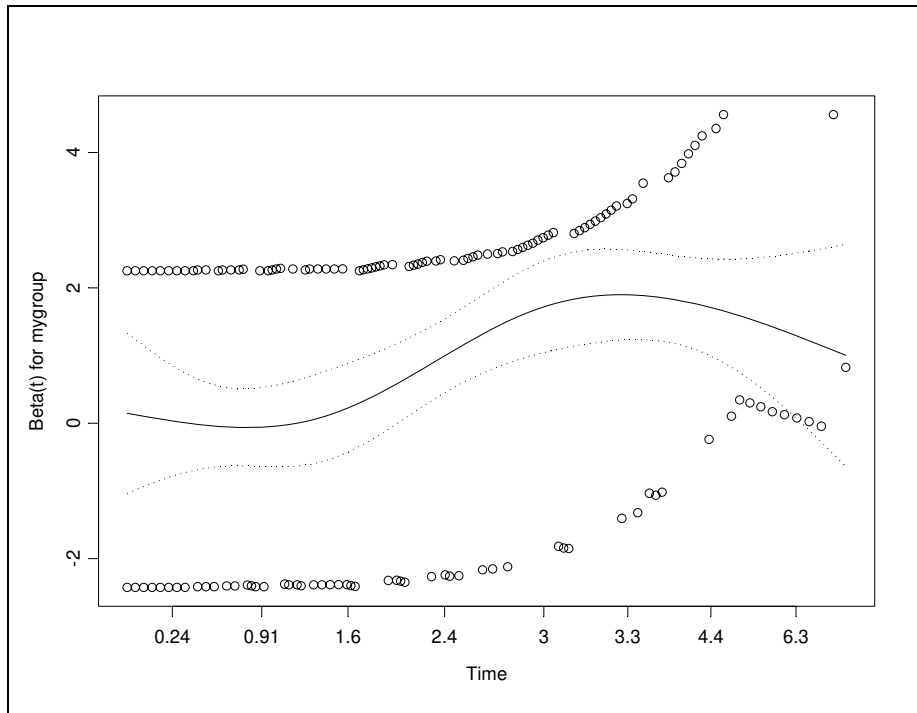


Figure 3. Example of a single simulated data set with late difference.

The size of datasets $n_1 = n_2 = N^* = 100$.

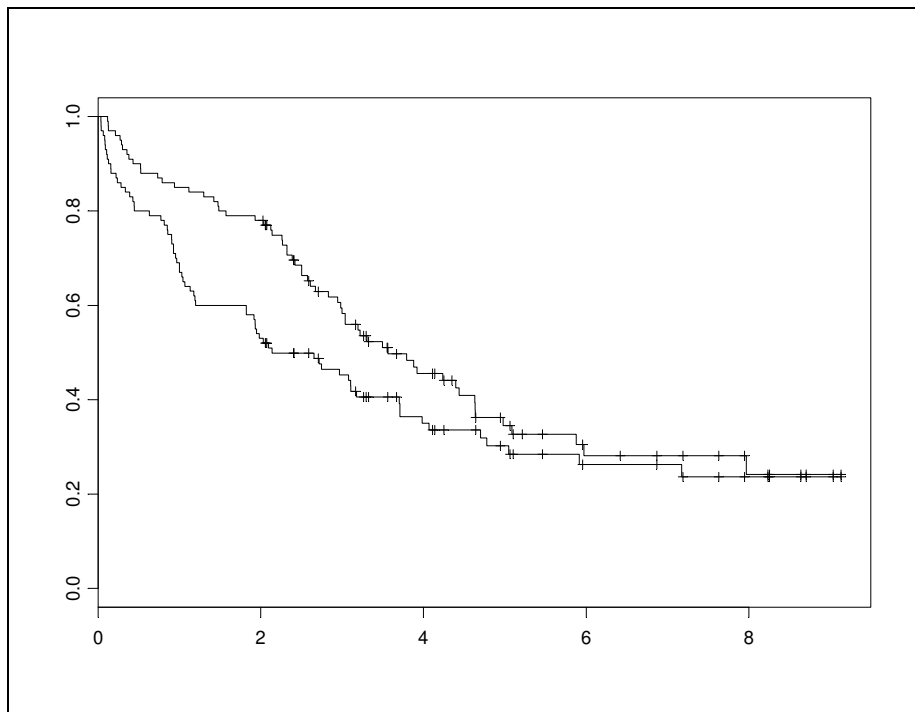
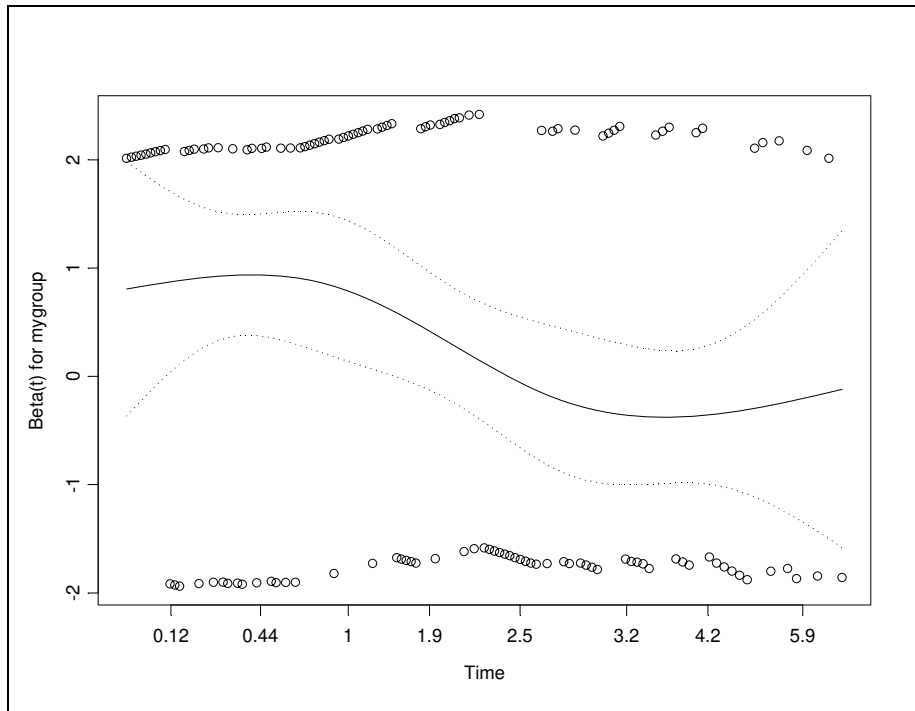


Figure 4. Example of a single simulated data set with early difference.

The size of datasets $n_1 = n_2 = N^* = 100$.

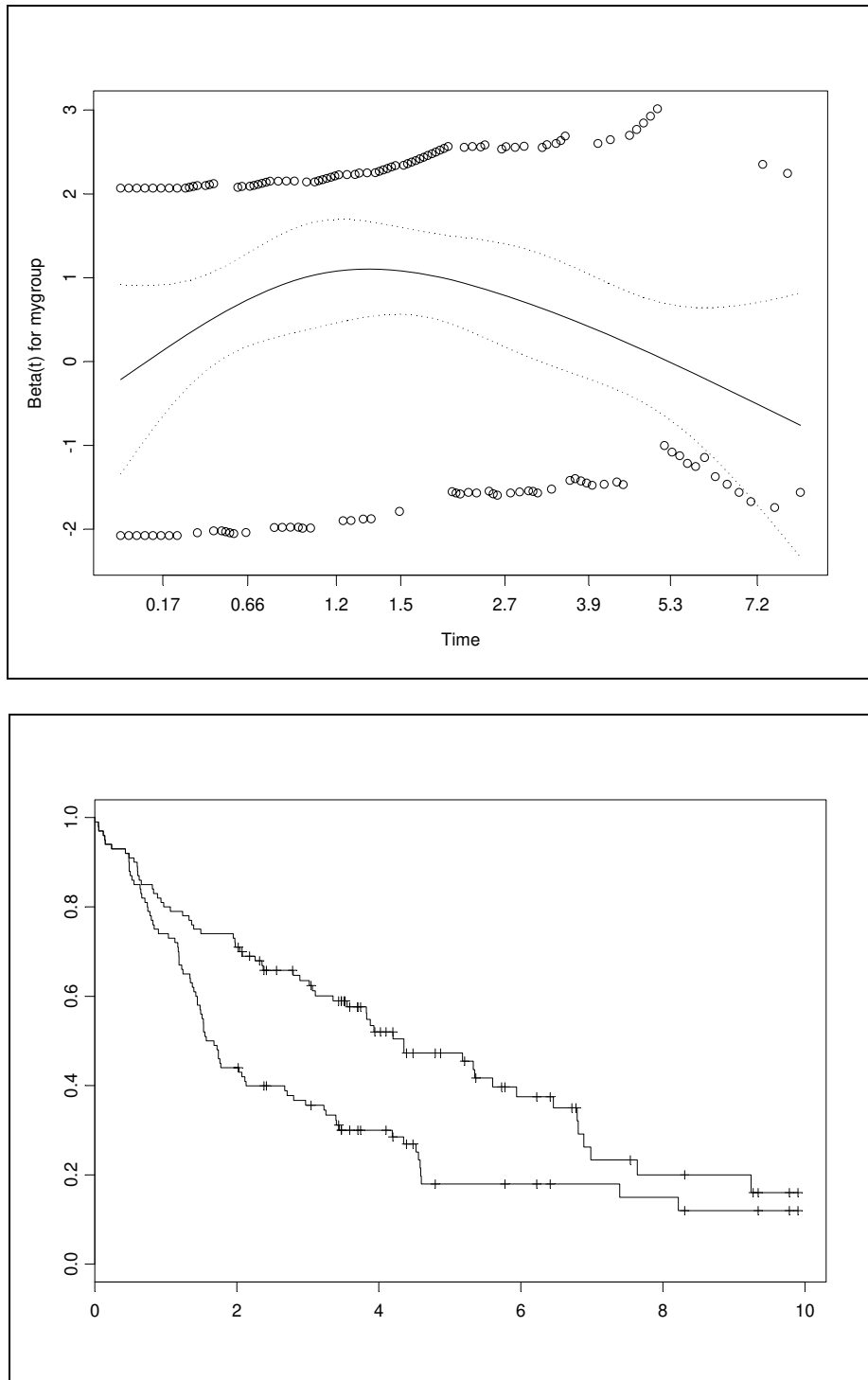


Figure 5. Example of a single simulated data set with middle difference.

The size of datasets $n_1 = n_2 = N^* = 100$.

3.5.2. Choosing parameters

The simple log-rank and Harrington-Fleming's weighted log-rank test can be formulated by properly setting the weight function $W(t_i)$ in Eqs (3.38) and (3.39). For the simple log-rank test, $W(t_i)$ simply equals to unity for all t , which means all the survival times are treated with equal weight. For Harrington-Fleming's weighted log-rank test, the weight function $W(t_i)$ in Eqs. (3.38) and (3.39) is defined by

$$W_{p,q}(t_i) = \hat{S}(t_{i-1})^p [1 - \hat{S}(t_{i-1})]^q, \quad (3.55)$$

where $p \geq 0$, $q \geq 0$ (Klein and Moeschberger, 1997). Slightly different from the $\hat{S}(t_i)$ in Eq. (3.40), the $\hat{S}(t_{i-1})$ in Eq. (3.55) is the survival function at the previous failure time. When $p = q = 0$ Eq. (3.55) reduces to the weigh function for the simple log-rank test. When $p > 0$ and $q = 0$ we have the G-rho test. By choosing the values of p and q properly, we assign different weights to the data points. For example, when $q = 0$ and $p > 0$, the tests with this weight function put more weights on early difference. When $p = 0$ and $q > 0$, the tests put more weights on late difference. In our simulation tests, we set the values of p and q in the Harrington-Fleming test as follows:

Early difference data: $p = 2$, $q = 0$;

Late difference data: $p = 0$, $q = 2$;

Middle difference data: $p = q = 1$.

For the log-rank test with the new weight function, we need to set the parameter ψ for the new weight function in Eq. (3.3). In this study we used three values for ψ , which are 0.01, 1.0, and 5.0. For the baseline hazard function, we used the values of 0.001, 0.1 and 0.3. The optimal values of ψ will be discussed later.

3.5.3. Simulation test for data difference

We chose the sample size of the simulation to be 1500, with group I and group II having the same size. That is, $n_1 = n_2 = N^*$. The data points are sorted ascendingly according to t and divided into 10 subgroups. For each subgroup a factor is multiplied by t to create the early, middle, or late difference between data group I and II. For example, when we model early difference data, we set a factor array of (0.85, 0.80, 0.75, 0.80, 0.85, 0.90, 0.93, 0.95, 0.97, 0.99). The first subgroup of 150 survival times in group II equal to the product of 0.80 and the first 150 data from group I. The second subgroup of 150 survival times in group II equal to the product of 0.825 and the second 150 data from group I, and so on. The last subgroup of 150 survival time in group I and II are essentially the same. For each choice of factor array, $n_s = 1000$ simulations with randomly generated survival times were performed, with level of test equal to 0.05. The power of each test was computed according to Eq. (3.45).

One set of results for comparing the simple log-rank, Harrington-Fleming's weighted log-rank, and the new weight function tests are shown in Table 1. This set of data has early difference. As can be seen from the simulation results, the Harrington-Fleming's weighted log-rank test (with $p = 2$ and $q = 0$) performs poorly in capturing the difference in the two groups. The simple log-rank test has shown a higher sensitivity than the Harrington-Fleming's test,

giving an average power of about 0.22. In contrast, the test with new weight function has about twice the power of a simple log-rank when the parameter ψ equals to 1.0. We should point out that all three test methods show a low power (less than 0.5) in differentiating the data group, mainly due to the very small difference we implanted in the simulated datasets. The power of Harrington-Fleming's test is relatively low. However, this does not indicate that Harrington-Fleming fails. If we increase the data difference by changing the premultiplier factor array, the testing powers for all three tests increase rapidly, but the power of the new weight function test remains the highest before they reach unity. Also we found that variation in the constant baseline hazards function does not change the relative sensitivity of these three test methods in differentiating the data groups with early departure.

For the datasets with middle difference, the test with the new weight function shows a higher power than the Harrington-Fleming test. However, its powers are in the same range as that of the simple log-rank test. From Table 2, we can see the new test has a slightly higher power than simple log-rank test when we choose ψ to be 1.0. This observation is similar to that we have seen for early difference test. It seems that $\psi = 1$ is good choice for testing the early and middle difference survival data using the new weight function. Again, the testing results have only a minor change when we vary the baseline function values. This indicates that the value of the constant baseline hazards function has negligible effect on the sensitivity of three testing methods in the light of the simulation fluctuations.

Simulation results from the late difference data are shown in Table 3. We can see that the tests with all three methods have a power in the same range.

We note that the factor arrays for the simulated data with different departure pattern (early, middle, or late difference) are in the same magnitude, ranging from 0.80 to 0.99. However, the

significance of nonproportionality in these datasets identified by `cox.zph` varies. Early difference data was identified by `cox.zph` as having the most significant nonproportionality. The fact that the new weigh function is more powerful than the simple log-rank and Harrington-Fleming's weighted log-rank tests in differentiating the early difference data, indicates that it is suitable for discriminating nonproportional hazards survival data.

Summarizing the results shown in Table 1 to 3, we can see that the advantage of the new test is obvious. For early difference data, the new method shows better performance than either simple log-rank or Harrington-Fleming's method. For middle difference, it is better than the Harrington-Fleming's test. Even for the later difference, which has the least nonproportionality, the new weight function method has a similar sensitivity as the other two methods. This indicates that the new weighting function is successful in properly accounting for the nonproportional frailty effect of the simulated survival data.

We also studied the dependence of the proposed new method on the censoring level, sample size, and the baseline distribution. We performed simulations with censoring level 5%, 10%, and 30%. We found that increasing the censoring level from 0% to 30% would decrease the sensitivity of all three testing methods, including the new method proposed in this work. However, the relative powers of these methods are consistent with the results obtained above. Simulations with various sample size ($N^* = 500, 1000, 1500$) indicates that decreasing the sample size would cause the power of the three methods to decrease. But similar to the impact of censoring level, the relative sensitivity of three methods was not changed. Simulations were also performed using a Weibull distribution as the baseline instead of the exponential distribution. The results from the Weibull distribution simulations are qualitatively consistent with that from

the exponential baseline simulations. The results for testing the censoring level, sample size, and the baseline distribution are summarized in Tables 4 to 6.

Table 1. Simulation results for samples with early difference using the simple log-rank, Harrington-Fleming's weighted log-rank test, and the new weighted logrank test.

Hypothesis: $\beta = 0$; 1000 simulations for each N^* ; $n_1 = n_2 = N^*$, level of test is 0.05. The piecewise nonproportional hazard array is (0.85, 0.80, 0.75, 0.80, 0.85, 0.90, 0.93, 0.95, 0.97, 0.99).

baseline hazards function	ψ	Test with new weight function, P_1	Simple log-rank, P_2	Harrington-Fleming weighted log-rank, P_3	P_2 / P_1	P_3 / P_1
0.001	5.0	0.285	0.226	0.15	0.793	0.526
	1.0	0.445			0.508	0.337
	0.01	0.229			0.987	0.655
0.1	5.0	0.297	0.237	0.163	0.798	0.548
	1.0	0.446			0.531	0.365
	0.01	0.239			0.992	0.682
0.3	5.0	0.312	0.228	0.157	0.731	0.503
	1.0	0.439			0.519	0.358
	0.01	0.214			1.065	0.733

Table 2. Simulation results for samples with middle difference using the simple log-rank, Harrington-Fleming's weighted log-rank test, and the new weighted logrank test.

Hypothesis: $\beta = 0$; 1000 simulations for each N^* ; $n_1 = n_2 = N^*$, level of test is 0.05. The imposed nonproportional hazard array is (0.99, 0.95, 0.90, 0.85, 0.80, 0.80, 0.85, 0.90, 0.95, 0.99).

baseline hazards function	ψ	Test with new weight function, P_1	Simple log-rank, P_2	Harrington-Fleming weighted log-rank, P_3	P_2 / P_1	P_3 / P_1
0.001	5.0	0.890	0.834	0.640	0.937	0.719
	1.0	0.945			0.883	0.677
	0.01	0.849			0.982	0.754
0.1	5.0	0.915	0.832	0.571	0.909	0.624
	1.0	0.947			0.879	0.603
	0.01	0.818			1.017	0.698
0.3	5.0	0.887	0.834	0.555	0.940	0.626
	1.0	0.957			0.871	0.580
	0.01	0.835			0.999	0.665

Table 3. Simulation results for samples with late difference using the simple log-rank, Harrington-Fleming's weighted log-rank test, and the new weighted logrank test.

Hypothesis: $\beta = 0$; 1000 simulations for each N^* ; $n_1 = n_2 = N^*$, level of test=0.05. The imposed nonproportional hazard array is (1.00, 0.99, 0.98, 0.97, 0.95, 0.93, 0.91, 0.89, 0.87, 0.85).

baseline hazards function	ψ	Test with new weight function, P_1	Simple log-rank, P_2	Harrington-Fleming weighted log-rank, P_3	P_2 / P_1	P_3 / P_1
0.001	5.0	0.977	0.986	0.988	1.009	1.011
	1.0	0.977			1.009	1.011
	0.01	0.981			1.005	1.007
0.1	5.0	0.978	0.985	0.984	1.007	1.006
	1.0	0.967			1.019	1.018
	0.01	0.987			0.998	0.997
0.3	5.0	0.983	0.986	0.981	1.003	0.998
	1.0	0.972			1.014	1.009
	0.01	0.983			1.003	0.998

Table 4. Impact of censoring level on the sensitivity of the methods (early difference).

Hypothesis: $\beta = 0$; 1000 simulations for each N^* ; $n_1 = n_2 = N^* = 1500$, level of test is 0.05. The piecewise nonproportional hazard array is (0.85, 0.80, 0.75, 0.80, 0.85, 0.90, 0.93, 0.95, 0.97, 0.99).

Censoring level	baseline hazards function	ψ	Test with new weight function	Simple log-rank	Harrington-Fleming weighted log-rank
5%	0.1	5.0	0.227	0.232	0.143
		1.0	0.430		
		0.01	0.248		
10%	0.1	5.0	0.215	0.243	0.123
		1.0	0.413		
		0.01	0.250		
30%	0.1	5.0	0.198	0.227	0.111
		1.0	0.389		
		0.01	0.233		

Table 5. Impact of sample size on the sensitivity of the methods (early difference).

Hypothesis: $\beta = 0$; 1000 simulations for each various N^* ; $n_1 = n_2 = N^*$, level of test is 0.05. The piecewise nonproportional hazard array is (0.85, 0.80, 0.75, 0.80, 0.85, 0.90, 0.93, 0.95, 0.97, 0.99).

Sample Size N^*	baseline hazards function	ψ	Test with new weight function	Simple log-rank	Harrington-Fleming weighted log-rank
500	0.1	5.0	0.127	0.108	0.04
		1.0	0.220		
		0.01	0.110		
1000	0.1	5.0	0.189	0.144	0.07
		1.0	0.345		
		0.01	0.156		
1500	0.1	5.0	0.297	0.228	0.157
		1.0	0.446		
		0.01	0.239		

Table 6. Simulation results for samples with Weibull distribution (τ, ρ) , with parameter $\tau = 1.5$.

Hypothesis: $\beta = 0$; 1000 simulations for each N^* ; $n_1 = n_2 = N^* = 1500$, level of test is 0.05. The piecewise nonproportional hazard array is (0.85, 0.80, 0.75, 0.80, 0.85, 0.90, 0.93, 0.95, 0.97, 0.99).

baseline hazards function	ψ	Test with new weight function	Simple log-rank	Harrington-Fleming weighted log-rank
0.001	5.0	0.295	0.235	0.138
	1.0	0.421		
	0.01	0.196		
0.1	5.0	0.311	0.247	0.171
	1.0	0.507		
	0.01	0.182		
0.3	5.0	0.281	0.213	0.153
	1.0	0.429		
	0.01	0.252		

3.5.4. Optimal value of parameter ψ

Theoretically, the value of ψ can be any arbitrary positive number between 0 and infinity (Oakes and Jeong, 1998). As long as a positive ψ is chosen, the weighting function will be always between 0 and 1. However, the change in ψ value will change the distribution of the weighting function. Therefore, further careful work needs to be done before we can give a reasonable rule in choosing the optimal ψ value.

From the simulation results we recognize that an optimal value exists for parameter ψ in terms of testing the nonproportional hazards data using the newly developed log-rank method. This finding can be interpreted from the dependence of the weight function on ψ and the survival distribution. For example, for survival data following exponential distribution, we can plot the weight function calculated from Eq. (3.37) versus survival time with three choices of ψ values, namely, $\psi = 0.1$, $\psi = 1$, and $\psi = 5$, as shown in Figure 6. As can be seen from the plots, increase in the value of ψ puts more weight on the data with late difference. So a small value of ψ is desired if we want to test the early difference of survival distribution. On the other hand, if the value of ψ is too small, only the data with minimum survival time will be given weight, which is apparently disadvantageous for testing any data difference. Therefore, an optimal value of ψ has to be chosen in order to obtain the maximum sensitivity of the method.

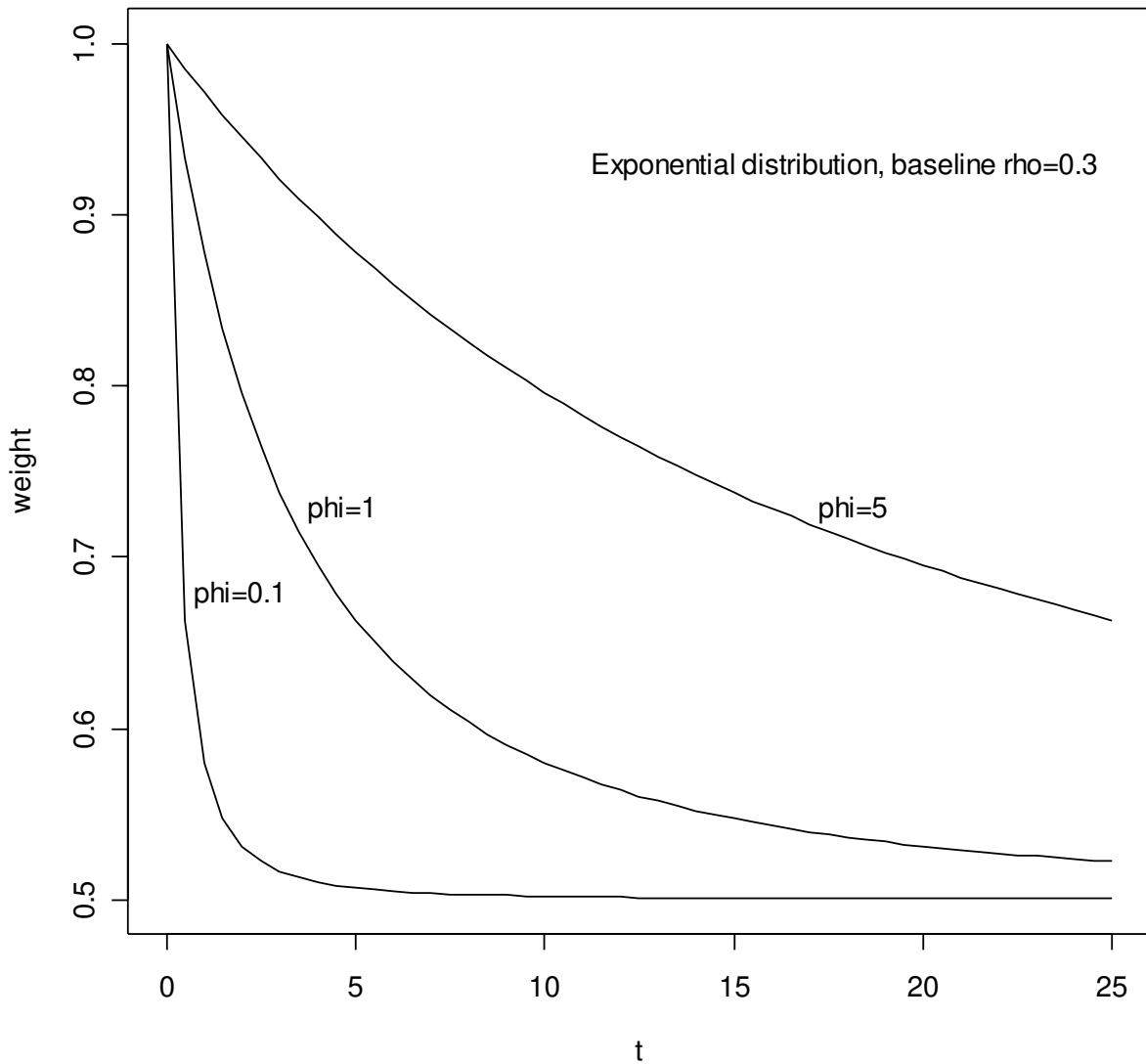


Figure 6. New weight function as a function of survival time and parameter ψ .

The survival time follows exponential distribution, with baseline parameter $\rho = 0.3$. The parameter $\psi = 5, 1, 0.1$, respectively, for the curves from top to bottom.

Theoretically, an optimal value for the parameters used in the newly developed weight functions can be calculated mathematically from the survival data, if a distribution is assumed

about the random effects. For example, for Gamma frailty, we have derived a weight function which is dependent on the parameter ρ as given by Eq. (3.27),

$$\lambda[S(t)] = [S(t)]^{1/\kappa} = [S(t)]^\rho. \quad (3.27)$$

Once the value of ρ can be estimated, the weight function in Eq. (3.27) with $\rho = \hat{\rho}$ will be optimal, where $\hat{\rho}$ is a numerical estimate of ρ . Likewise, the optimal weight function for inverse Gaussian frailty can also be calculated by estimating the value of ψ in Eq. (3.37). More discussions on estimating ρ and ψ can be found in Chapter 5.

3.5.5. Comparison with supreme weighted log-rank test

The supremum version of weighted log-rank test is given by

$$\sup_{t \in (0, \tau)} |Z(t)|,$$

where $Z(t) = \frac{\Delta(t)}{\sqrt{V(\tau)}}$, and $\Delta(t)$ and $V(\tau)$ are calculated from Eqs. (3.38) and (3.39),

$$\Delta(t) = \sum_{t_i \in (0, t)} W(t_i) \left[d_{i1} - Y_{i1} \left(\frac{d_{i1} + d_{i2}}{Y_{i1} + Y_{i2}} \right) \right],$$

$$V(\tau) = \sum_{t_i \in (0, \tau)} [W(t_i)]^2 \frac{Y_{i1}}{Y_i} \left(1 - \frac{Y_{i1}}{Y_i} \right) \left(\frac{Y_i - d_i}{Y_i - 1} \right) d_i.$$

The simulations were extended to compare the power of supreme versions of the log-rank and weighted log-rank test proposed in this work. We found that the power of the supreme versions of the log rank and weighted log-rank tests are comparable with their native version. Taking the log-rank method using the new weight function as an example, we found that the supreme version has almost the same power as the native one does. Such results are consistent

with the previous findings that supremum statistics are nearly as powerful as the traditional statistics under the proportional hazards assumption (Fleming et al., 1987) or more powerful than the traditional statistics in certain nonproportional hazards settings (Kosorok and Lin, 1999). One typical set of results comparing the supreme and traditional weight log-rank test using the new weight function is show in Table 4.

Table 7. Comparison of supreme and traditional version of the new weighted log-rank test.

Hypothesis: $\beta = 0$; 1000 simulations for each N^* ; $n_1 = n_2 = N^* = 1500$, level of test=0.05. The imposed nonproportional hazard arrays are same as those used in Table 1-3.

baseline hazards function	ψ	Early difference		Middle difference		Late difference	
		Traditional version	Supremum version	Traditional version	Supremum version	Traditional version	Supremum version
0.001	5.0	0.285	0.291	0.890	0.891	0.977	0.979
	1.0	0.445	0.446	0.945	0.947	0.977	0.973
	0.01	0.229	0.229	0.849	0.837	0.981	0.982
0.1	5.0	0.297	0.298	0.915	0.886	0.978	0.980
	1.0	0.446	0.436	0.947	0.949	0.967	0.964
	0.01	0.239	0.237	0.818	0.831	0.987	0.984
0.3	5.0	0.312	0.315	0.887	0.899	0.983	0.980
	1.0	0.439	0.442	0.957	0.934	0.972	0.975
	0.01	0.214	0.224	0.835	0.834	0.983	0.985

3.5.6. Application in real data

We tested our newly developed method with the real data from an NSABP trial, Protocol B-14. On this phase III trial, patients with primary breast cancer, negative axillary nodes, and oestrogen receptor positive tumors were randomized to receive either tamoxifen (a hormonal therapy) or placebo following surgery. The trial itself is described in details in literature (Fisher

et al., 1989, 1996). In this dataset, 1450 patients were randomized to placebo and 1435 patients were randomized to tamoxifen over a period of seven years and have been followed for an additional ten years. The endpoint of interest was disease-free survival (DFS). Only 2817 eligible patients (1413 for placebo group, 1404 for treatment group) were used for our analysis. About 47% (700 for placebo, 562 for tamoxifen group) of the patients have had a DFS event, and 53% have censored data.

Figure 7 shows the survival for the failure distributions from the Kaplan-Meier estimates. The upper line is for treatment group and lower line is for placebo group. From the plot we readily see the two groups are nonproportional in hazards. The `cox.zph` test on the proportional hazards assumption give a p-value of 0.00004. We performed test of data difference using the simple log-rank, Harrington-Fleming's weighted log-rank test with $p = q = 2.0$, and the new weighted logrank test with $\psi = 1$. The p-values obtained from these three different method are 5×10^{-8} , 4×10^{-5} , and 8×10^{-9} , respectively. This indicates that the log-rank test with the new weight function is the most sensitive one among the three methods. The results are consistent with the simulations results in section 3.5.3. In this case, all three methods give the same conclusion, mainly due to the degree of extremeness of the treatment effect in the selected real data.

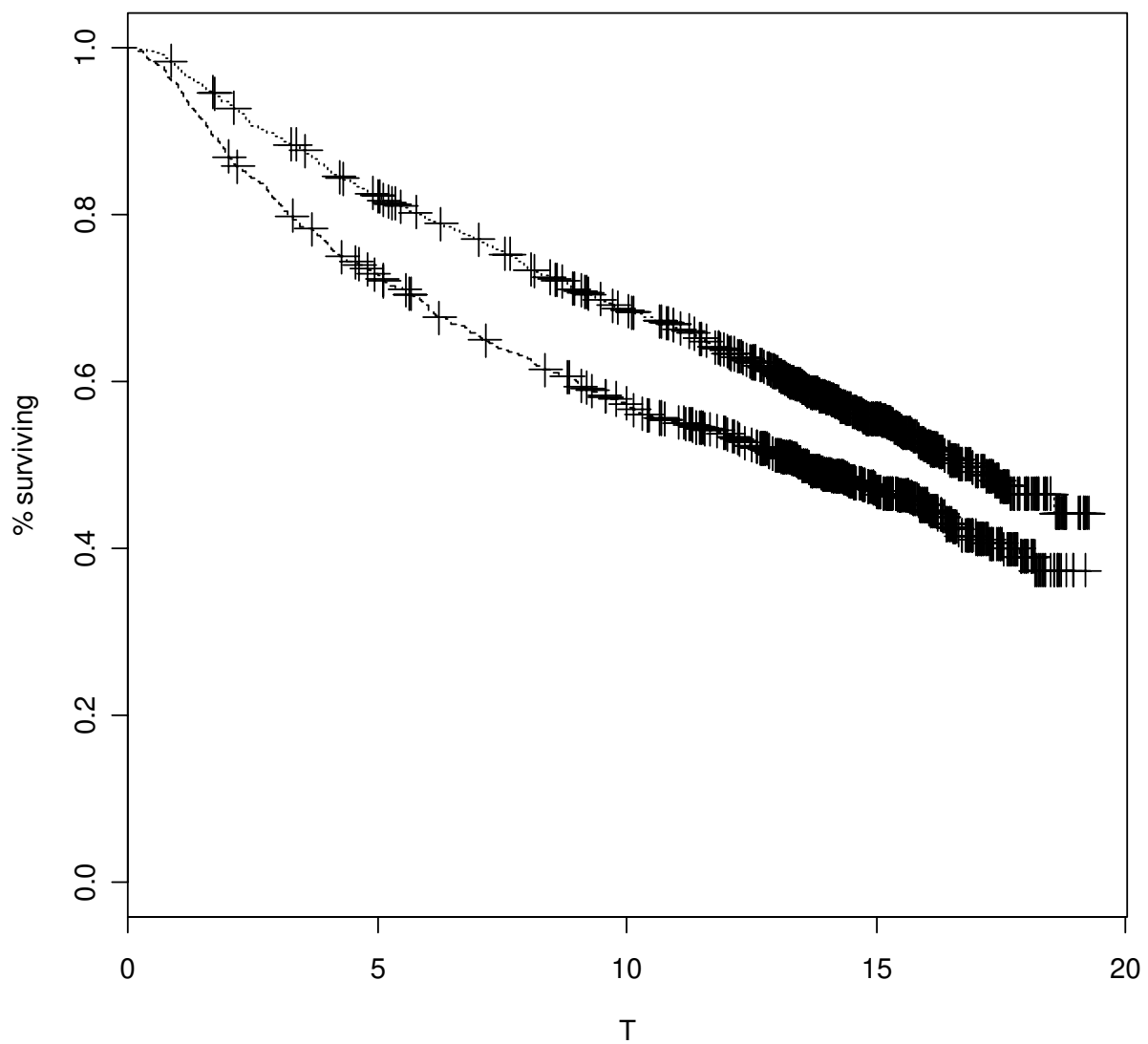


Figure 7. Survival function in NSABP B-14 data estimated from Kaplan-Meier method. The upper line is the treatment group, and the lower line is the placebo group.

3.6. Conclusion

We studied the sensitivity of a newly developed weighted log-rank test, and compared it with the simple log-rank test and Harrington-Fleming's weighted log-rank test, in testing treatment with nonproportional survival data using Monte Carlo simulations. We found that the new test shows a better sensitivity in capturing the difference between the data group when the survival data has significant nonproportionality (here the data with early difference). For the datasets with less nonproportionality (here the data with middle difference), the test with the new weight function has better sensitivity than that of Harrington-Fleming's weighted log-rank test, similar to that of the simple-log rank test. For late difference which has least nonproportionality, all three methods have similar sensitivity.

4. TEST FOR PROPORTIONAL ODDS

4.1. Introduction to the Dauxois-Kirmani Test

The objective of the second part of this thesis is to develop a parametric method for testing of the proportional odds assumption over two groups of survival times. The first part of this work focused on the method to model survival data with nonproportional hazards by using the extended Cox's proportional hazards model by including time-dependent explanatory variables or by introducing a frailty parameter (Oakes and Jeong, 1998). An alternative approach is through the proportional odds model. In this chapter we present a new test based on a nonparametric method introduced by Dauxois and Kirmani (Dauxois and Kirmani, 2003). This method is suitable for testing the proportionality of odds of survival data with random censoring. We will review briefly about this method.

In the proportional odds model, the odds of an individual in group i surviving beyond some time t are defined as

$$\phi_i(t) = \frac{S_i(t)}{1 - S_i(t)}, \quad (4.1)$$

where S_i is the survival function for group i . We say that the two groups satisfy a proportional odds model if $\phi_2(t) = \alpha\phi_1(t)$ for all $t > 0$ and some constant $\alpha > 0$. If h_i is the hazard function corresponding to S_i , then under the proportional odds model

$$\frac{h_2(t)}{h_1(t)} = \frac{1}{1 - (\alpha - 1)S_1(t)}, \quad (4.2)$$

so that, for $\alpha > 1$ ($\alpha < 1$) the hazard ratio h_2/h_1 is an increasing (a decreasing) function converging to unity.

A simple method can be used to explore the suitability of using the proportional odds model to study two survival distributions. One can calculate the odds ratio by

$$\hat{\theta} = \frac{\hat{\phi}_2(t)}{\hat{\phi}_1(t)} = \frac{\hat{S}_2(t)[1 - \hat{S}_1(t)]}{\hat{S}_1(t)[1 - \hat{S}_2(t)]}, \quad (4.3)$$

where \hat{S}_i is Kaplan-Meier estimator of survival function,

$$\hat{S}_i(t) = \begin{cases} 1 & \text{if } t < t_1 \\ \prod_{t_i \leq t} \left[1 - \frac{d_i}{Y_i}\right], & \text{if } t_i \leq t \end{cases} \quad (4.4)$$

If the plot of $\hat{\theta}$ versus t shows that $\hat{\theta}$ is independent on t , proportional odds about the samples can be justified, and vice versa.

Recently, Dauxois and Kirmani developed a procedure for testing the proportional odds assumption of two independent random samples with randomly right-censored lifetimes (Dauxois and Kirmani, 2003). The test statistic is given by

$$Z = (n_1 + n_2)^{\frac{1}{2}} \Gamma / \sigma, \quad (4.5)$$

where Z is asymptotically normal, $Z \sim N(0,1)$ for testing H_0 versus H_1 , where

$$H_0 : \phi_2(t) = \alpha \phi_1(t), \text{ for all } t > 0 \text{ and some } \alpha > 0,$$

$$H_1 : \phi_2(t) \text{ and } \phi_1(t) \text{ are not proportional.}$$

In Eq. (4.5), n_i is the sample size of group i , Γ and σ are given by

$$\Gamma = \Gamma(K_1, K_2) = \psi_{11}\psi_{22} - \psi_{12}\psi_{21}, \quad (4.6)$$

and

$$\sigma^2 = \psi_{22}\psi_{21}v_{11} - \psi_{22}\psi_{11}v_{12} - \psi_{12}\psi_{21}v_{21} + \psi_{12}\psi_{11}v_{22}. \quad (4.7)$$

Value of ψ_{ij} is given by

$$\psi_{ij} = \int_{m_1 \vee \tau_1}^{m_2 \wedge \tau_2} K_i(t)\phi_j(t)dt, \quad (4.8)$$

where K_i ($i = 1, 2$) are weight functions. And v_{ij} is given by

$$v_{ij} = \int_{\tau_1}^{\tau_2} \int_{\tau_1}^{\tau_2} K_i(t)K_j(s)\phi_1(s)\phi_2(t) \left\{ \frac{n_1}{F_1(s)F_1(t)} \int_0^{s \wedge t} \frac{dN_1(u)}{Y_1^2(u)} + \frac{n_2}{F_2(s)F_2(t)} \int_0^{s \wedge t} \frac{dN_2(u)}{Y_2^2(u)} \right\} dsdt. \quad (4.9)$$

In Eqs. (4.8) and (4.9),

$$m_1 = \max \left\{ \min_{1 \leq j \leq n_1; \delta_{1j}=1} T_{1j}, \min_{1 \leq j \leq n_2; \delta_{2j}=1} T_{2j} \right\},$$

$$m_2 = \min \left\{ \max_{1 \leq j \leq n_1} T_{1j}, \max_{1 \leq j \leq n_2} T_{2j} \right\},$$

$$m_2 \wedge \tau_2 = \min(m_2, \tau_2), \quad m_1 \vee \tau_1 = \max(m_1, \tau_1) \quad (4.10)$$

where T_{ij} is the j^{th} observed survival time in group i , and $\delta_{ij} = 0$ if T_{ij} is censored. Integration

interval $[\tau_1, \tau_2]$ can be chosen as the end points of the data. $Y_i(t) = \sum_{j=1}^{n_i} I\{T_{ij} \geq t\}$ for $t > 0$

and $i = 1, 2$, is the number of objects in group i at risk at time t . $F_i(t) = 1 - S_i(t)$, is the

cumulative distribution function. $N_i(t) = \sum_{j=1}^{n_i} I\{T_{ij} \leq t, \delta_{ij} = 1\}$ is the number of events in

group i before or at time t . The number of events in group i at time t is denoted by d_i .

The values of ϕ_i and S_i can be estimated from the Kaplan-Meier estimator of survival function, using Eqs. (4.1) and (4.4). One significant choice has to be made about the weight functions K_1 and K_2 . Dauxois and Kirmani chose arbitrarily the weight function corresponding

to the log-rank test for proportional hazards and the normalized form of Gehan's weight function,

$$K_1 = \left(\frac{n_1 + n_2}{n_1 n_2} \right) \frac{Y_1(t)Y_2(t)}{Y_1(t) + Y_2(t)}, \quad (4.11)$$

$$K_2 = \frac{Y_1(t)Y_2(t)}{n_1 n_2}. \quad (4.12)$$

The numerical estimate of ψ_{ij} based on Kaplan-Meier estimators is given by

$$\hat{\psi}_{ij} = \int_{m_1 \vee \tau_1}^{m_2 \wedge \tau_2} K_i(t) \hat{\phi}_j(t) dt = \int_{m_1 \vee \tau_1}^{m_2 \wedge \tau_2} K_i(t) d\hat{\Phi}_j(t) \approx \sum_{t=m_1 \vee \tau_1}^{m_2 \wedge \tau_2} K_i(t) \Delta \hat{\Phi}_i(t), \quad (4.13)$$

where $\hat{\Phi}_i(t)$ denotes the odds of group i . Similarly, the numerical estimate of v_{ij} based on Kaplan-Meier estimators is

$$\hat{v}_{ij} = \sum_{t=\tau_1}^{\tau_2} \sum_{s=\tau_1}^{\tau_2} K_i(t) K_j(s) \Delta \hat{\Phi}_1(s) \Delta \hat{\Phi}_2(t) \left[\frac{n_1}{F_1(s)F_1(t)} \sum_{u=0}^{s \wedge t} \frac{\Delta N_1(u)}{Y_1^2(u)} + \frac{n_2}{F_2(s)F_2(t)} \sum_{u=0}^{s \wedge t} \frac{\Delta N_2(u)}{Y_2^2(u)} \right] \quad (4.14)$$

4.2. Extending the Method with Parametric Assumptions

One of the keys to the Dauxois and Kirmani's method is the choice of the weight functions, K_1 and K_2 . They did not attempt to develop new weight functions, but simply followed Gill and Schumacher (1987) and Sengupta et al. (1998), chose arbitrarily the weight functions K_1 and K_2 as two commonly employed format. The weight functions chosen by Dauxois and Kirmani (Dauxois and Kirmani, 2003) correspond to the log-rank test for proportional hazards and the normalized form of Gehan's weight function. In theory, any cadlag functions, not necessarily

predictable, can be used if they lead to increasing K_1/K_2 . The purpose is to build a positive weight function

$$w(s, t) = K_1(t)K_2(s) - K_1(s)K_2(t) \quad (4.15)$$

to be used in the expression of

$$\gamma(w) = \iint_{\tau_1 < s < t < \tau_2} w(s, t) [\phi_1(t)\phi_2(s) - \phi_2(t)\phi_1(s)] ds dt, \quad (4.16)$$

where $\gamma(w)$ can be considered as a measure of nonproportionality of the odd function $\phi_1(t)$ and $\phi_2(t)$ on the time interval $[\tau_1, \tau_2]$.

Besides the weight functions used by Dauxois and Kirmani, other suitable choices include

$$K = \left(\frac{n_1 + n_2}{n_1 n_2} \right) \frac{Y_1(t)Y_2(t)}{Y_1(t) + Y_2(t)} [\hat{S}(t)]^p, \quad (4.17)$$

where $\hat{S}(t)$ is the Kaplan-Meier estimator computed from the combined survival data sample up to time t . The power parameter p is in $[0, 1]$.

In addition, one may also explore the application of certain weight functions specific to data with particular distributions. That is, K_1 and K_2 can be derived based on the assumption of survival data distributions. In the following example, we show how to derive K_1 and K_2 for survival data that follow the log-logistic distribution.

From Dauxios and Kirmani's paper, we have

$$\begin{aligned}
K_1 &= \frac{n_1 + n_2}{n_1 n_2} \frac{Y_1(t+)Y_2(t+)}{Y_1(t+) + Y_2(t+)} = \frac{[Y_1(t+)/n_1][Y_2(t+)/n_2]}{\frac{Y_1(t+) + Y_2(t+)}{n_1 + n_2}} \\
&= \frac{[Y_1(t+)/n_1][Y_2(t+)/n_2]}{n_1[Y_1(t+)/n_1] + n_2[Y_2(t+)/n_2]}, \\
&= \frac{[Y_1(t+)/n_1][Y_2(t+)/n_2]}{\frac{n_1}{n_1 + n_2}[Y_1(t+)/n_1] + \frac{n_2}{n_1 + n_2}[Y_2(t+)/n_2]}
\end{aligned} \tag{4.18}$$

and

$$K_2 = \frac{Y_1(t+)Y_2(t+)}{n_1 n_2} = \frac{Y_1(t+)}{n_1} \frac{Y_2(t+)}{n_2}. \tag{4.19}$$

As $n_1 \rightarrow \infty, n_2 \rightarrow \infty$, by the Glivenko-Cantelli Theorem (Shorack and Wellner, 1986), it is easy

to prove that,

$$\begin{aligned}
\frac{Y_1(t+)}{n_1} &\rightarrow S_1(t); & \frac{Y_1(t+)}{n_1} &\rightarrow P(T_1 > t) \\
\frac{Y_2(t+)}{n_2} &\rightarrow S_2(t); & \frac{Y_2(t+)}{n_2} &\rightarrow P(T_2 > t)
\end{aligned} ,$$

and we define

$$\rho = \frac{n_1}{n_1 + n_2},$$

with

$$\frac{n_2}{n_1 + n_2} = 1 - \rho.$$

So we get

$$K_1 = \frac{[S_1(t)][S_2(t)]}{\rho[S_1(t)] + (1 - \rho)[S_2(t)]} = \frac{[P(T_1 > t)][P(T_2 > t)]}{\rho[P(T_1 > t)] + (1 - \rho)[P(T_2 > t)]}, \tag{4.20}$$

$$K_2 = \frac{Y_1(t+)Y_2(t+)}{n_1 n_2} = \frac{Y_1(t+)}{n_1} \frac{Y_2(t+)}{n_2} = [S_1(t)][S_2(t)] = [P(T_1 > t)][P(T_2 > t)]. \tag{4.21}$$

Weight functions based on Eqs. (4.20) and (4.21) utilized the parametric information about the distribution of the survival time t . For continuous distributions, $S(t)$ can be used in Eqs. (4.20) and (4.21), while for discrete distributions, $P(T > t)$ can be used.

Substituting the survival function of the log-logistic distribution into (4.18) and (4.19), we have

$$K_1 = \frac{1}{\rho \left[1 + (t / \xi_2)^{\beta_2} \right] + (1 - \rho) \left[1 + (t / \xi_1)^{\beta_1} \right]}, \quad (4.22)$$

$$K_2 = \frac{1}{\left[1 + (t / \xi_1)^{\beta_1} \right] \left[1 + (t / \xi_2)^{\beta_2} \right]}. \quad (4.23)$$

These are the weight functions for the log-logistic survival data. The test statistics can still be computed by

$$(n_1 + n_2)^{\frac{1}{2}} \Gamma(K_1, K_2) = (n_1 + n_2)^{\frac{1}{2}} (\psi_{11}\psi_{22} - \psi_{12}\psi_{21}),$$

with

$$\begin{aligned} \psi_{ij} &= \int_{\tau_1}^{\tau_2} K_i(t) \phi_j(t) dt; \\ \phi_j(t) &= \frac{1}{e^{(t/\xi)^{\beta_j}} - 1}, \end{aligned} \quad (4.24)$$

which can be calculated numerically.

The variance of the statistic is calculated by

$$\sigma^2 = \psi_{22}\psi_{21}v_{11} - \psi_{22}\psi_{11}v_{12} - \psi_{12}\psi_{21}v_{21} + \psi_{12}\psi_{11}v_{22}.$$

with

$$v_{ij} = \int_{\tau_1}^{\tau_2} \int_{\tau_1}^{\tau_2} K_i(t) K_j(s) \phi_1(s) \phi_2(t) \left\{ \frac{1}{F_1(s)F_1(t)} \int_0^{s \wedge t} \frac{d\Lambda_1(u)}{y_1(u)} + \frac{1}{F_2(s)F_2(t)} \int_0^{s \wedge t} \frac{d\Lambda_2(u)}{y_2(u)} \right\} ds dt.$$

Here

$$\Lambda(t) = -\ln S(t).$$

And

$$y_i(t) = \frac{Y_i(t)}{n_i} \rightarrow S_i(t)$$

as $n_1 \rightarrow \infty, n_2 \rightarrow \infty$. So we have

$$\frac{d\Lambda_i(t)}{y_i(t)} = -\frac{dS_i(t)}{[S_i(t)]^2}.$$

Then,

$$\int_0^{s \wedge t} \frac{d\Lambda_i(u)}{y_i(u)} = \int_0^{s \wedge t} -\frac{dS_i(u)}{[S_i(u)]^2} = \frac{1}{S_i(u)} \Big|_0^{s \wedge t} = \frac{1}{S_i(s \wedge t)} - 1.$$

Substituting it into the variance equation, we obtain

$$v_{ij} = \int_{\tau_1}^{\tau_2} \int_{\tau_1}^{\tau_2} K_i(t)K_j(s)\phi_1(s)\phi_2(t) \left\{ \frac{1}{F_1(s)F_1(t)} \left[\frac{1}{S_1(s \wedge t)} - 1 \right] + \frac{1}{F_2(s)F_2(t)} \left[\frac{1}{S_1(s \wedge t)} - 1 \right] \right\} ds dt,$$

where

$$S_\beta(t) = \frac{1}{1 + (t/\xi)^\beta}, \quad F_\beta(t) = \frac{(t/\xi)^\beta}{1 + (t/\xi)^\beta}. \quad (4.25)$$

The odds ratio between $\phi_1(t)$ and $\phi_2(t)$ is

$$\frac{\phi_2(t)}{\phi_1(t)} = \frac{(\xi_2/t)^{\beta_2}}{(\xi_1/t)^{\beta_1}}.$$

Let group I be the baseline survival time with $\beta_1 = 1$. The odds ratio $\frac{\phi_2(t)}{\phi_1(t)} = 1$ if $\xi_1 = \xi_2$ and

$\beta_2 = 1$, which means the survival data from T_1 and T_2 are proportional in odds. Otherwise, if

$\beta_2 > 1$, the survival data from T_1 and T_2 have nonproportional odds. Now let $\beta_1 = 1$, $\xi_1 = \xi_2 = \xi$, and we denote β_2 simply as β so that we can drop out all the subscripts in the formula. We obtain

$$S_1(t) = \frac{1}{1 + (t/\xi)}, S_2(t) = \frac{1}{1 + (t/\xi)^\beta}$$

$$F_1(t) = \frac{(t/\xi)}{1 + (t/\xi)}, F_2(t) = \frac{(t/\xi)^\beta}{1 + (t/\xi)^\beta}$$

$$\phi_1(t) = (\xi/t), \phi_2(t) = (\xi/t)^\beta$$

$$K_1 = \frac{1}{\rho[1 + (t/\xi)^\beta] + (1 - \rho)[1 + (t/\xi)]}, K_2 = \frac{1}{[1 + (t/\xi)][1 + (t/\xi)^\beta]}$$

4.3. Estimation of Parameters in Assumed Distributions

In order to use the parametric weight functions, one needs to estimate the model parameters in the assumed distribution about the survival data. For example, if we assume that the survival data follow the log-logistic distribution, we need to estimate parameters (ξ, β) . One commonly used method for estimating the distribution parameters is the maximum likelihood estimate. In maximum likelihood, if the probability density function (pdf) of the survival time is $f(t)$, the likelihood of the n observations t_1, t_2, \dots, t_n is the product of the probability of an event occurring at time t_i given by $\prod_{i=1}^n f(t_i)$, which is dependent on the parameters in the pdf assumed.

The maximum likelihood estimates of these parameters are those values for which the likelihood function is maximized.

If there are censored survival time in the sample, suppose that r of the n survival time t_1, t_2, \dots, t_r are observed, and the remaining $n - r$ times $t_{r+1}^*, t_{r+2}^*, \dots, t_n^*$ are right-censored, then the contribution of r observed times to the likelihood function is $\prod_{i=1}^r f(t_i)$. For $n - r$ censored time, if a survival time is censored at time t^* , we know that the lifetime of the individual is at least t^* , and the probability of this event is $P(T \geq t^*) = S(t^*)$. Then the contribution of each censored time to the likelihood function is $\prod_{j=1}^{n-r} S(t_j^*)$. So the total likelihood function is

$$\prod_{i=1}^r f(t_i) \prod_{j=1}^{n-r} S(t_j^*). \quad (4.26)$$

If we assign an indicator variable δ_i to each survival time,

$$\begin{cases} \delta_i = 1, & \text{observed} \\ \delta_i = 0, & \text{censored} \end{cases},$$

then we can rewrite the likelihood function as

$$\prod_{i=1}^n [f(t_i)]^{\delta_i} [S(t_i)]^{1-\delta_i}, \quad (4.27)$$

without discriminating if a survival time t_i is censored or not in Eq. (4.27) by incorporating the indicator variable δ_i .

Furthermore, Eq. (4.27) can be rewritten as

$$\prod_{i=1}^n \left[\frac{f(t_i)}{S(t_i)} \right]^{\delta_i} S(t_i). \quad (4.28)$$

From Eq. (1.7) we know $h(t_i) = \frac{f(t_i)}{S(t_i)}$, so Eq. (4.28) becomes

$$\prod_{i=1}^n [h(t_i)]^{\delta_i} S(t_i). \quad (4.29)$$

Eqs. (4.26), (4.27), (4.28), (4.29) are equivalent. Eq. (4.29) is most often used because usually the pdf of a survival time could be complicated but the corresponding hazard function could be simple.

For survival time data following the log-logistic distribution, we have

$$f(t) = \frac{\beta t^{\beta-1} \xi^{-\beta}}{(1 + t^{\beta} \xi^{-\beta})^2}, \text{ and } S(t) = \frac{1}{1 + (t/\xi)^{\beta}}.$$

So the hazard function is given by

$$h(t) = \frac{f(t)}{S(t)} = \frac{\frac{\beta t^{\beta-1} \xi^{-\beta}}{[1 + (t/\xi)^{\beta}]^2}}{\frac{1}{1 + (t/\xi)^{\beta}}} = \frac{\beta t^{\beta-1} \xi^{-\beta}}{1 + (t/\xi)^{\beta}}.$$

The likelihood function is

$$L(\xi, \beta) = \prod_{i=1}^n \left(\frac{\beta t_i^{\beta-1} \xi^{-\beta}}{1 + (t_i/\xi)^{\beta}} \right)^{\delta_i} \frac{1}{1 + (t_i/\xi)^{\beta}}. \quad (4.30)$$

The log-likelihood function is

$$\begin{aligned} \ln L(\xi, \beta) &= \sum_{i=1}^n \ln \left[\left(\frac{\beta t_i^{\beta-1} \xi^{-\beta}}{1 + (t_i/\xi)^{\beta}} \right)^{\delta_i} \frac{1}{1 + (t_i/\xi)^{\beta}} \right] \\ &= \sum_{i=1}^n \left[\delta_i \ln \beta + \delta_i \ln t_i^{\beta-1} - \delta_i \beta \ln \xi - (\delta_i + 1) \ln \{1 + (t_i/\xi)^{\beta}\} \right] \end{aligned} \quad (4.31)$$

Since there are r observed events, correspondingly we have

$$\sum_{i=1}^n \delta_i = r.$$

So the log-likelihood function is

$$\ln L(\xi, \beta) = r \ln \beta - r\beta \ln \xi + \sum_{i=1}^n \delta_i \ln t_i^{\beta-1} - \sum_{i=1}^n (\delta_i + 1) \ln[1 + (t_i / \xi)^\beta]. \quad (4.32)$$

Differentiation with respect to parameters (ξ, β) and let the derivatives equal to zero, we have

$$\frac{\partial}{\partial \xi} \ln L(\xi, \beta) = -\frac{r\beta}{\xi} - \sum_{i=1}^n (\delta_i + 1) \frac{\beta(t_i / \xi)^{\beta-1}(-t_i / \xi^2)}{1 + (t_i / \xi)^\beta} = 0, \quad (4.33)$$

$$\frac{\partial}{\partial \beta} \ln L(\xi, \beta) = \frac{r}{\beta} - r \ln \xi + \sum_{i=1}^n \delta_i \frac{t_i^{\beta-1} \ln t_i}{t_i^{\beta-1}} - \sum_{i=1}^n (\delta_i + 1) \frac{(t_i / \xi)^\beta \ln(t_i / \xi)}{1 + (t_i / \xi)^\beta} = 0. \quad (4.34)$$

Notice that $\frac{\partial t_i^x}{\partial x} = t_i^x \ln t_i$.

After some algebra, Eqs. (4.33) and (4.34) becomes

$$r = \sum_{i=1}^n \frac{(\delta_i + 1)t_i^\beta}{\xi^\beta + t_i^\beta}, \quad (4.35)$$

$$\frac{r}{\beta} + \sum_{i=1}^n \delta_i \ln t_i = \sum_{i=1}^n \frac{(\delta_i + 1)t_i^\beta}{\xi^\beta + t_i^\beta} \ln t_i. \quad (4.36)$$

Solving Eqs. (4.35) and (4.36) simultaneously using nonlinear method such as Newton-Raphson method, we can find the estimate for parameters $(\hat{\xi}, \hat{\beta})$. The estimated $(\hat{\xi}, \hat{\beta})$ then can be used in Eqs. (4.22) and (4.23) to calculate the parametric weight functions used in the extended testing method.

The asymptotic variance-covariance matrix of $(\hat{\xi}, \hat{\beta})$ is given by

$$\text{var}(\hat{\xi}, \hat{\beta}) = \left\{ -E \left[\begin{array}{cc} \frac{\partial^2}{\partial \xi^2} \ln L(\xi, \beta) & \frac{\partial^2}{\partial \xi \partial \beta} \ln L(\xi, \beta) \\ \frac{\partial^2}{\partial \beta \partial \xi} \ln L(\xi, \beta) & \frac{\partial^2}{\partial \beta^2} \ln L(\xi, \beta) \end{array} \right] \right\}^{-1}. \quad (4.37)$$

4.4. Simulation Methodology

The log-logistic distribution is a distribution that owns both the accelerated failure time property and the proportional odds property (Collett, 2003). Therefore, it is natural to use it in conjunction with the proportional odds model. For the log-logistic distribution, the survival function is given by

$$S_{\beta}(t) = \frac{1}{1 + (t / \xi)^{\beta}} \quad (t \geq 0, \beta > 0, \xi > 0). \quad (4.38)$$

The cumulative distribution function is

$$F_{\beta}(t) = 1 - S_{\beta}(t) = 1 - \frac{1}{1 + (t / \xi)^{\beta}} = \frac{(t / \xi)^{\beta}}{1 + (t / \xi)^{\beta}}. \quad (4.39)$$

The odds function is expressed as

$$\phi_{\beta}(t) = \frac{S_{\beta}(t)}{F_{\beta}(t)} = (t / \xi)^{-\beta} = (\xi / t)^{\beta}. \quad (4.40)$$

If the baseline distribution X_1 has a log-logistic distribution with $\beta = 1$, then its odds function is

$$\phi_1(t) = \frac{S_1(t)}{F_1(t)} = \xi / t, \quad (4.41)$$

and

$$\phi_{\beta}(t) = (\xi / t)^{\beta}, \quad (4.42)$$

so the odds ratio between $\phi_1(t)$ and $\phi_{\beta}(t)$ is

$$\frac{\phi_1(t)}{\phi_\beta(t)} = \frac{\xi/t}{(\xi/t)^\beta} = \xi' t^{\beta-1}, \quad (4.43)$$

where $\xi' = \xi^{1-\beta}$.

When $\beta = 1$, the odds ratio $\frac{\phi_1(t)}{\phi_\beta(t)} = \text{constant}$, the survival data X_1 and X_β are proportional

in odds. Otherwise, if $\beta > 1$, the survival data from X_1 and X_β are nonproportional in odds.

First we try to validate if the Dauxios-Kirmanian method can tell nonproportional odds data. To do that, we create two sets of data with nonproportional odds. The first group of data is from X_1 , the second is from X_β with $\beta > 1$. In our simulations we set the constants ξ' (hereafter ξ) to be 1.0.

To generate survival data X_1 and X_β following the log-logistic distributions, we use the facts that the cumulative distribution functions F_1 and F_β conform to uniform distributions in the range of [0,1]. That is,

$$F_1(t) = \frac{t/\xi}{1+t/\xi} = u \sim \text{UNIF}(0,1),$$

$$F_\beta(t) = \frac{(t/\xi)^\beta}{1+(t/\xi)^\beta} = u \sim \text{UNIF}(0,1). \quad (4.44)$$

Then, we obtain

$$t_1 = \frac{\xi}{1/u - 1}; \quad t_2 = \xi \left(\frac{1}{1/u - 1} \right)^\beta = \xi^{1-\frac{1}{\beta}} \left(\frac{\xi}{1/u - 1} \right)^\beta = \xi^{1-\frac{1}{\beta}} t_1^{\frac{1}{\beta}}. \quad (4.45)$$

The numerical simulation procedure is

1. Generate survival data with appropriate distributions. For comparison purpose, we generate data from the log-logistic distributions.

$$u \sim \text{UNIF}(0,1);$$

$$t_1 = \frac{\xi}{1/u - 1} \text{ and } t_2 = \xi^{1-\frac{1}{\beta}} t_1^{\frac{1}{\beta}}$$

Please note two independent sets of u are used in calculate t_1 and t_2 so that they are not correlated.

2. Set random censoring,

$$c_1 \sim \text{UNIF}(5,25); c_2 \sim \text{UNIF}(5,25)$$

$$t_1 = \min(t_1, c_1) ; t_2 = \min(t_2, c_2)$$

The value of c is set according to appropriate censoring levels.

3. Build data table according to event (or censoring). No tie is allowed.
4. Count $Y_i(t), d_i(t), N_i(t), \Delta N_i(t)$, where $t \in [\tau_1, \tau_2], \tau_1 = \min(t_{ij}), \tau_2 = \max(t_{ij})$ for $i = 1, 2; j = 1, n_j$.
5. Estimate $S_i(t), F_i(t), \phi_i(t), t \in [\tau_1, \tau_2], i = 1, 2$ from Eqs. (4.4) and (4.1), and estimate $K_i(t)$ from either Eqs. (4.11) and (4.12) for Dauxois-Kirmanian method, or Eqs. (4.22) and (4.23) for the new test method developed in this work.
6. Calculate $m_1, m_2, \hat{\psi}_{ij}, \hat{v}_{ij}, i = 1, 2; j = 1, 2$ from Eqs. (4.10), (4.8), and (4.9).
7. Calculate $\Gamma, \hat{\sigma}, Z$ from Eqs. (4.6), (4.5), and (4.7).
8. Calculate the p-value based on $Z \sim N(0,1)$ to see if the test is significant (p-value being less than the nominal significance level).
9. Repeat 1-8 for 1000 times and calculate the percentage of significant test, which is the power of the method.

4.5. Results and Discussions

4.5.1. Examination of the existing method

First we tried to reproduce the simulation results reported by Dauxois and Kirmani by performing simulations with same conditions as that in the literature (Dauxois and Kirmani, 2003). We modeled survival data in the context of the log-logistic distribution. The pair of survival data have common median $\xi = 150$. The hypothesis of interest are $H_0: \beta = 1$ and $H_1: \beta > 1$. The null hypothesis $H_0: \beta = 1$ is rejected in favor of $H_1: \beta > 1$ if and only if the test statistic $Z = (n_1 + n_2)^{\frac{1}{2}} \Gamma / \sigma$ exceeds the $100(1 - \alpha)$ th percentile of the standard normal distribution, where α is the nominal significance level. The assessment of power against various alternative values of β is obtained by carrying out Monte Carlo simulations.

For simulation purpose, two pairs of sample sizes (n_1, n_2) were selected, $(n_1, n_2) = (50, 60)$ and $(n_1, n_2) = (100, 120)$. Random censoring was imposed on the data with same percentage censoring level for the two samples. One typical example of survival data is shown in Figure 6, the Kaplan-Meier estimate of survival functions with $\beta = 2$, $\xi = 1$.

We reproduced the Dauxois and Kirmani's approach by performing Monte Carlo simulations. Our simulation results at selected β values for nominal significance levels $\alpha = 0.02$ and $\alpha = 0.05$ are shown in Table 4. They agree excellently with the results reported in the original Dauxois and Kirmani paper.

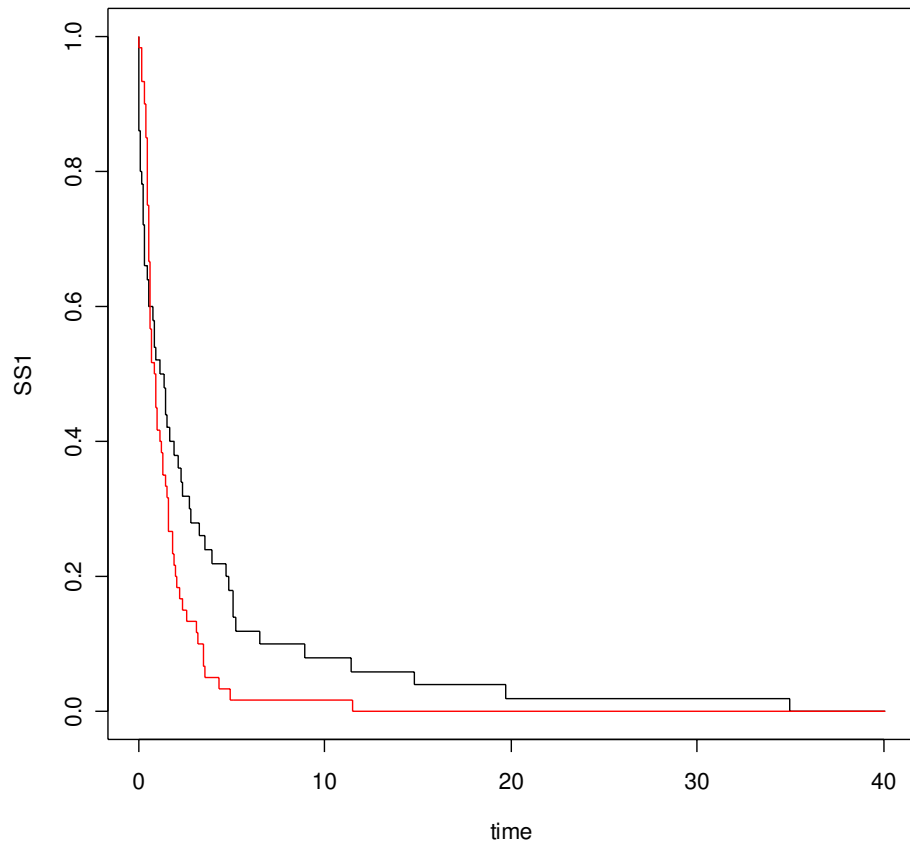


Figure 8. Kaplan-Meier estimators of survival distribution of the log-logistic data.

The two groups of survival data have common median $\xi = 1.0$, the lower line is the baseline with $\beta = 1$, the upper line is the comparison group with $\beta = 2$.

Table 8. Reproduction of the Monte Carlo simulation results of the original paper using the Dauxois-Kirmani approach.

Each point is simulated for 20000 simulations, $\xi = 150$, significance level $\alpha = 0.02, 0.05$.

α	Sample size, (n_1, n_2)	Censoring level, %	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 6$
0.02	(50, 60)	0	0.0058	0.5422	0.8790	0.9771	0.9987
		10	0.0054	0.4943	0.8458	0.9436	0.9772
		30	0.0051	0.4244	0.7594	0.8836	0.9308
	(100, 120)	0	0.0078	0.7888	0.9783	0.9981	1.0000
		10	0.0075	0.7227	0.9689	0.9947	0.9986
		30	0.0066	0.7025	0.9069	0.9484	0.9775
0.05	(50, 60)	0	0.0193	0.7039	0.9310	0.9891	0.9998
		10	0.0200	0.6681	0.9033	0.9658	0.9853
		30	0.0213	0.5540	0.8734	0.9326	0.9481
	(100, 120)	0	0.0303	0.8163	0.9887	0.9991	1.0000
		10	0.0301	0.8104	0.9836	0.9968	0.9992
		30	0.0310	0.7912	0.9350	0.9237	0.9406

4.5.2. Simulation analysis of the extended method

From simulations we can see that the original method proposed by Dauxois and Kirmani is rather conservative in the sense that the power is much smaller than the nominal α when the null hypothesis is true (i.e. when $\beta = 1$, two survival distributions are proportional in odds). The ideal value of power for $\beta = 1$ should be close to the value of α in order for the test method to be most efficient. It was noticed that the standard deviation of Dauxoi-Kirmina statistic $Z = (n_1 + n_2)^{1/2} \Gamma / \hat{\sigma}$ is smaller than the asymptotic expected value of unity. Apparently, the plug-in estimator of $\hat{\sigma}$ overestimates the asymptotic standard deviation of Γ resulting in underdispersion of Z .

We performed Monte Carlo simulation of survival data following the log-logistic distribution, with proportional or non proportional odds by varying the values of β , using the extended testing method with weight functions derived with the parametric assumption. One typical set of simulation results using the new method is shown in Table 5. The simulation results show that the extended method has an appreciably higher power in testing the validity of the proportional odds assumption than that of the original Dauxois-Kirmani method for the log-logistic survival data. This indicates that the inclusion of parametric information in the existing nonparametric approach can significantly enhance its sensitivity. It also suggests that the original method by Dauxois and Kirmani is too conservative.

Table 9. Monte Carlo simulation results of the log-logistic distribution data using the extended method.

Each point is simulated for 1000 simulations, $\xi = 1.0$, significance level $\alpha = 0.02, 0.05$.

α	Sample size, n_1, n_2	Censoring level, %	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 6$	
0.02	50, 60	0	0.015	0.621	0.989	0.993	1.000	
		10	0.017	0.600	0.948	0.969	0.999	
		30	0.018	0.567	0.927	0.948	0.998	
	100, 120	0	0.020	0.996	0.991	0.999	1.000	
		10	0.022	0.957	0.945	0.988	0.999	
		30	0.022	0.944	0.917	0.968	0.999	
	0.05	50, 60	0	0.040	0.817	0.998	1.000	1.000
			10	0.040	0.803	0.958	0.998	1.000
			30	0.042	0.779	0.935	0.989	0.999
100, 120		0	0.048	0.981	0.998	1.000	1.000	
		10	0.049	0.945	0.965	1.000	1.000	
		30	0.052	0.924	0.944	0.998	1.000	

4.5.3. Application in real data

We applied the extended method in the NSABP trial data, Protocol B-14, to comparing its sensitivity with that of the existing Dauxois-Kirmani method. The real data have been used as test of the new method in the first part of this proposal (see section 3.4.5). Previous work by Jeong et al. (Jeong et al., 2003) has shown that the parametric test for the proportional odds assumption of the Protocol B-14 data gives a small p-value of 0.000097, indicating that the data do not satisfy the proportional odds assumption. Therefore the dataset is a good example for testing the extended Dauxois-Kirmani method developed in this work.

Application of the new method in real data requires one to estimate the distribution parameters using the maximum likelihood estimation method as described in section 4.3. The parameters (β, ξ) were estimated using the procedure outlined in section 4.3. By solving the nonlinear equations (4.35) and (4.36) simultaneous we obtain the MLE estimate of (β, ξ) as

$$\begin{aligned}\hat{\beta} &= 1.2927 \\ \hat{\xi} &= 6.1214\end{aligned}$$

These two values are used in using the Dauxois-Kirmani method and the extended method to calculate the p-value upon the null hypothesis that data are proportional in survival odds. Test for the proportional odds assumption of the real data using Dauxois-Kirmani's nonparametric method gives a p-value of 0.0078, and the test using our extended parametric method gives a p-value of 0.00081, which again indicates that the extended parametric method is more sensitive than the existing one in testing the assumption of nonproportional odds.

4.6. Conclusion

We studied the sensitivity of a newly developed parametric method in detecting the validity of the proportional odds model assumption between two groups of survival data. The extended method is based on the premise that the test developed with distribution information would have better sensitivity than a nonparametric test method. We evaluated type I error and power probabilities of the new method by using the simulated survival data following the log-logistic distribution. The error probabilities are compared with ones obtained from the existing method. The results indicate that the extended method performs with a higher sensitivity than the existing nonparametric method.

5. FUTURE WORK

5.1. Determination of Optimal Values for ψ and ρ

There are several open questions in our new method developed for testing the data difference in survival distributions with nonproportional hazards. One further task is to apply the new method in nonproportional data with Gamma frailty. In Chapter 3, our studies were focused on the case where the frailty is assumed to follow inverse Gaussian distribution. In reality, the Gamma frailty is also extensively encountered in modeling nonproportional hazards data. Under the Gamma frailty model, the weight function given by Eq. (3.27) should be used (Oakes and Jeong, 1998). However, the value of ρ has to be determined for best efficiency of the new test method developed. The extension of the current work to Gamma frailty would be quite straightforward.

Another primary concern is to determine the optimal value for the parameters ψ used in the weight function (3.37). In section 3.5.4 we explain qualitatively the existence of optimal value of ψ . However, it would be desirable to develop a rigorous mathematical procedure for the determination of optimal ψ given that parametric information about the survival data is available or can be assumed. Accordingly, we describe briefly a procedure to determine the optimal value of ψ via the maximum likelihood estimation (MLE) method.

The proportional hazards model incorporated with a randomization variable z_i is

$$h_1(t | x, z) = \exp(\beta z + \gamma x)h_0(t), \quad (5.1)$$

where $h_0(t)$ is an unknown baseline hazard function. Correspondingly, the survival function is

$$S_1(t | x, z) = S_0(t)^{\exp(\beta z + \gamma x)}. \quad (5.2)$$

Let

$$\theta = \exp(\beta z + \gamma x), \quad (5.3)$$

and

$$H_0 = -\ln S_0(t). \quad (5.4)$$

Thus

$$S_0 = e^{-H_0}, \quad S_1(t | x, z) = S_0(t)^{\exp(\beta z + \gamma x)} = S_0(t)^\theta = e^{-H_0\theta}, \quad (5.5)$$

and we also have

$$F_1(t) = 1 - S_1(t) = 1 - e^{-H_0\theta} = u. \quad (5.6)$$

Similarly, if T follows Weibull distribution with parameters (τ, ρ) , then the survival function is given by

$$S_0(t) = \exp(-\rho t^\tau). \quad (5.7)$$

Substituting Eq. (5.7) into Eq. (5.4) we get

$$H_0 = \rho t^\tau. \quad (5.8)$$

Eq. (5.6) becomes

$$1 - e^{-\rho t^\tau \theta} = u. \quad (5.9)$$

and

$$\left(\frac{-\ln(1-u)}{\rho\theta} \right)^{\frac{1}{\tau}} = t. \quad (5.10)$$

Survival time can be generated using Eq. (5.10)

$$t_i = \left(\frac{-\ln(1 - u_i)}{\rho\theta_i} \right)^{\frac{1}{\tau}}, \quad (5.11)$$

where u_i is from a uniform distribution $u(0,1)$; and

$$\theta_i = \exp(\beta z_i + \gamma x_i),$$

where z_i is the group indicator generated from a Bernoulli distribution $Bernoulli(0.5)$ with $\beta = 0.5$; $e^{\gamma x_i} = w_i$ is the frailty. We assume that w follows either Gamma or inverse Gaussian distribution. In real applications, the distribution for w can be determined by using the graphical method described in section 3.3. If w follows gamma distribution $gamma(\kappa, 1/\kappa)$ with unity mean and variance of $\rho = 1/\kappa$, the survival function of t_i with gamma frailty $gamma(\kappa, 1/\kappa)$ unit mean and variance $1/\kappa$ is given by the Laplace transform

$$S(t_i; \kappa) = p[\xi_i H_0(t_i)], \quad (12)$$

where $p()$ is the Laplace transform of gamma distribution

$$p(s) = \left(\frac{1}{1 + s/\kappa} \right)^\kappa. \quad (13)$$

So Eq. (5.12) becomes

$$S(t_i; \kappa) = \left(\frac{1}{1 + \xi_i H_0(t_i)/\kappa} \right)^\kappa = \left(\frac{\kappa}{\kappa + \xi_i H_0(t_i)} \right)^\kappa, \quad (5.14)$$

where $\xi_i = e^{\beta z_i}$ is the group indicator.

The hazard function is

$$h(t; \kappa) = -\frac{\partial \ln S(t; \kappa)}{\partial t} = -\frac{\partial [\kappa \ln \kappa - \kappa \ln(\kappa + \xi H_0(t))]}{\partial t} = \kappa \frac{\xi \frac{\partial H_0(t)}{\partial t}}{\kappa + \xi H_0(t)}. \quad (5.15)$$

From Eq. (5.8) we know $H_0(t) = \rho t^\tau$, so

$$h(t; \kappa) = \kappa \frac{\xi \rho \tau t^{\tau-1}}{\kappa + \xi H_0(t)} = \frac{\kappa \xi \tau H_0(t)}{[\kappa + \xi H_0(t)]t}. \quad (5.16)$$

The likelihood function of κ is given by

$$L(\kappa) = \prod_{i=1}^n [h(t_i; \kappa)]^{\delta_i} S(t_i; \kappa), \quad (5.17)$$

where

$$\begin{cases} \delta_i = 1, & \text{observed} \\ \delta_i = 0, & \text{censored} \end{cases}$$

The log-likelihood function is then

$$\ln L(\kappa) = \sum_{i=1}^r \ln h(t_i; \kappa) + \sum_{i=1}^N \ln S(t_i; \kappa), \quad (5.18)$$

where $\sum_{i=1}^n \delta_i = r$, are number of observed events. Substituting Eqs. (5.14) and (5.16) into Eq.

(5.18) we obtain

$$\begin{aligned} \ln L(\kappa) &= \sum_{i=1}^r \ln \frac{\kappa \xi_i \tau H_0(t_i)}{[\kappa + \xi_i H_0(t_i)]t_i} + \sum_{i=1}^N \ln \left(\frac{\kappa}{\kappa + \xi_i H_0(t_i)} \right) \\ &= \sum_{i=1}^r \{ \ln \kappa \tau + \ln \xi_i + \ln H_0(t_i) - \ln t_i - \ln[\kappa + \xi_i H_0(t_i)] \}, \\ &\quad + \sum_{i=1}^N \{ \kappa \ln \kappa - \kappa \ln[\kappa + \xi_i H_0(t_i)] \} \end{aligned} \quad (5.19)$$

Taking derivative in terms of κ gives

$$\frac{\partial \ln L(\kappa)}{\partial \kappa} = \frac{r}{\kappa} - \sum_{i=1}^r \frac{1}{\kappa + \xi_i H_0(t_i)} + N(\ln \kappa + 1) - \sum_{i=1}^N \left\{ \ln[\kappa + \xi_i H_0(t_i)] + \frac{\kappa}{\kappa + \xi_i H_0(t_i)} \right\}.$$

So κ can be estimated by solving the equation

$$\frac{r}{\kappa} - \sum_{i=1}^r \frac{1}{\kappa + \xi_i H_0(t_i)} + N(\ln \kappa + 1) - \sum_{i=1}^N \left\{ \ln[\kappa + \xi_i H_0(t_i)] + \frac{\kappa}{\kappa + \xi_i H_0(t_i)} \right\} = 0. \quad (5.20)$$

Similar procedure can be applied to survival data with inverse Gaussian frailty. If the frailty follows an inverse Gaussian distribution $invGauss(1, 2\psi)$ with unit mean and variance $1/2\psi$, the survival function of t_i is given by the Laplace transform

$$S(t_i; \psi) = p[\xi_i H_0(t_i)], \quad (5.21)$$

where $p(\cdot)$ is the Laplace transform of inverse Gaussian distribution

$$p(s) = \exp\{-2[\psi(\psi + s)]^{1/2} + 2\psi\}. \quad (5.22)$$

So Eq. (5.21) is

$$S(t_i; \psi) = \exp\{-2[\psi(\psi + \xi_i H_0(t_i))]^{1/2} + 2\psi\}, \quad (5.23)$$

where $\xi_i = e^{\beta z_i}$ is the group indicator.

The hazard function is

$$\begin{aligned} h(t; \psi) &= -\frac{\partial \ln S(t; \psi)}{\partial t} = -\frac{\partial \{-2[\psi(\psi + \xi H_0(t))]^{1/2} + 2\psi\}}{\partial t} \\ &= \sqrt{\psi} \frac{\xi \frac{\partial H_0(t)}{\partial t}}{\sqrt{\psi + \xi H_0(t)}} = \frac{\sqrt{\psi} \xi \tau H_0(t)}{t \sqrt{\psi + \xi H_0(t)}}. \end{aligned} \quad (5.24)$$

The log-likelihood function is then

$$\begin{aligned} \ln L(\psi) &= \sum_{i=1}^r \ln \frac{\sqrt{\psi} \xi_i \tau H_0(t_i)}{t_i \sqrt{\psi + \xi_i H_0(t_i)}} + \sum_{i=1}^N \{-2[\psi(\psi + \xi_i H_0(t_i))]^{1/2} + 2\psi\} \\ &= \sum_{i=1}^r \left\{ \frac{1}{2} \ln \psi + \ln \xi_i + \ln \tau + \ln H_0(t_i) - \ln t_i - \frac{1}{2} \ln[\psi + \xi_i H_0(t_i)] \right\}, \quad (5.25) \\ &\quad -2 \sum_{i=1}^N [\psi(\psi + \xi_i H_0(t_i))]^{1/2} + 2N\psi \end{aligned}$$

Taking derivative in terms of ψ gives

$$\frac{\partial \ln L(\psi)}{\partial \psi} = \frac{r}{2\psi} - \frac{1}{2} \sum_{i=1}^r \frac{1}{\psi + \xi_i H_0(t_i)} + 2N - \sum_{i=1}^N \frac{2\psi + \xi_i H_0(t_i)}{\sqrt{\psi[\psi + \xi_i H_0(t_i)]}}.$$

So ψ can be estimated by solving the equation

$$\frac{r}{2\psi} - \frac{1}{2} \sum_{i=1}^r \frac{1}{\psi + \xi_i H_0(t_i)} + 2N - \sum_{i=1}^N \frac{2\psi + \xi_i H_0(t_i)}{\sqrt{\psi[\psi + \xi_i H_0(t_i)]}} = 0. \quad (5.26)$$

Eqs. (5.20) and (5.26) determine the optimal values of ρ and ψ from MLE. Simulation procedures similar to those described in section 3.4 can be developed to model the process derived above.

For real data, we have to estimate the optimal values of ρ or ψ together with the parameters τ and ρ simultaneously under some assumption about the survival data t_1 and t_2 (for example, Weibull or exponential), and about the frailty w (for example, inverse Gaussian or gamma).

5.2. Linear Regression Test of Nonproportional Odds

As discussed in section 4.1, the proportional odds assumption of survival data can also be tested using a simple method by examination of the relationship between the odds ratio

$$\hat{\theta} = \frac{\hat{S}_1(t)[1 - \hat{S}_0(t)]}{\hat{S}_0(t)[1 - \hat{S}_1(t)]},$$

versus time. The survival functions $\hat{S}_i(t)$ can be estimated by the Kaplan-Meier method. The linear regression method is simple in mathematics, and easy to use. However, it is limited in one aspect that the plot of $\log \hat{\theta}$ vs. $\log t$ could be ambiguous when the data size is small where one can not obtain a clean curve to show the dependence of the odds ratio and time, due to data fluctuations. The method can be improved if a test statistic for the linear model regression method proposed can be developed. The `lm` package included in R or Splus assumes that the slope of the linear model follows a student's t distribution. Our preliminary simulation test using

the ℓ_m function in Splus indicates that rejection rate of null hypotheses at $\beta = 1$ is significantly higher than the nominal significance level. This implies that the statistic employed in the ℓ_m function in Splus may not be suitable for testing the null hypothesis in section 4.6. It would be desirable to derive an appropriate test statistic for the linear regression method proposed.

Linear model regression can be used to fit the relationship between the odds ratio and time if after manipulations the relationship between a function of odds ratio and time satisfies the following assumptions (1) linearity of the relationship between dependent and independent variables; (2) independence of the errors; (3) constant variance of the errors versus time and the predictions; and (4) normality of the error distribution.

Taking the log-logistic distribution as an example, the odd ratio

$$\theta(t) = \frac{\phi_0(t)}{\phi_1(t)} = \frac{(\xi/t)}{(\xi/t)^\beta} = (\xi/t)^{1-\beta},$$

which is

$$\ln \hat{\theta}(t) = (\beta - 1) \ln t + (1 - \beta) \ln \xi. \quad (5.27)$$

The proportional odds assumption corresponds to that $\beta = 1$.

Therefore, by examining the estimated value of β through regression of $\ln \hat{\theta}$ as a function of $\ln t$, one can readily tell if the survival distributions are proportional in odds. Let

$$\ln \hat{\theta} = a + b \ln t,$$

then we have

$$\begin{array}{ll} H_0 : b = 0 & \text{proportional odds} \\ H_a : b \neq 0 & \text{nonproportional odds} \end{array}$$

If the survival data is proportional in odds, the slope of line (5.27) should be close to zero when $\beta = 1$. Otherwise, the slope would not be close to zero. If an appropriate statistical

distribution can be postulated for the slope, a test statistic can be derived, given that the dependence of $\ln \hat{\theta}$ and time satisfies the four principal assumptions for using linear regression model.

APPENDIX A

Simulation Programs

Here we attached all the computing programs. Program 1.1~1.10 are for the algorithms used in the first part of this proposal, namely, simulations and real data calculations on using the new method to test the survival distribution difference for nonproportional hazards data.

Program 2.1~2.7 are the programs for the testing of proportional odds assumption using Dauxois and Kirmani's method and the new method. For simulations with large samples and significant number of replications, R program runs very slow. Therefore, we developed FORTRAN 77 programs to do the same task, which is very fast. The FORTRAN program is also attached.

Programs for new method on testing the survival distribution difference

Program 1.1: Testing the proportional hazards assumption using cox.zph.

```
# Simulation project with Cox model, Qing Xu
#
### divide random data into 10 groups for non proportional hazard ratio
nt<-200; n1<-nt/10; n2<-n1*2; n3<-n1*3;
n4<-n1*4; n5<-n1*5; n6<-n1*6; n7<-n1*7;
n8<-n1*8; n9<-n1*9; n10<-n1*10

### each group is characterized by a factor
# late difference
#factor 1<-1.0; factor 2<-0.99; factor 3<-0.98;
#factor 4<-0.97; factor 5<-0.95; factor 6<-0.93;
#factor 7<-0.91; factor 8<-0.89; factor 9<-0.87;
#factor 10<-0.85;

#early difference
#factor 1<-0.85; factor 2<-0.8; factor 3<-0.75;
#factor 4<-0.8; factor 5<-0.85; factor 6<-0.9;
#factor 7<-0.93; factor 8<-0.95; factor 9<-0.97;
#factor 10<-0.99;

#middle difference
#factor 1<-0.99; factor 2<-0.95; factor 3<-0.9;
#factor 4<-0.85; factor 5<-0.8; factor 6<-0.8;
#factor 7<-0.85; factor 8<-0.9; factor 9<-0.95;
#factor 10<-0.99;

### set baseline hazard function h_0(t)=rho
myrho<-0.3;
### set data set size
ni<-200;
### set loop index
```

```

i i <- 0;
### counter
i p1 <- 0;
repeat {
  i i <- i i + 1;
  ### generate proportional data myt 1
  myu <- sort(runif(nt));
  myt 1 <- -log(1 - myu) / myrho * (- 1);
  ### generate nonproportional data myt 2
  myt 2 <- myt 1;
  for (i in 1:n1) { myt 2[i] <- factor 1 * myt 1[i] };
  for (i in (n1+1):n2) { myt 2[i] <- factor 2 * myt 1[i] };
  for (i in (n2+1):n3) { myt 2[i] <- factor 3 * myt 1[i] };
  for (i in (n3+1):n4) { myt 2[i] <- factor 4 * myt 1[i] };
  for (i in (n4+1):n5) { myt 2[i] <- factor 5 * myt 1[i] };
  for (i in (n5+1):n6) { myt 2[i] <- factor 6 * myt 1[i] };
  for (i in (n6+1):n7) { myt 2[i] <- factor 7 * myt 1[i] };
  for (i in (n7+1):n8) { myt 2[i] <- factor 8 * myt 1[i] };
  for (i in (n8+1):n9) { myt 2[i] <- factor 9 * myt 1[i] };
  for (i in (n9+1):n10) { myt 2[i] <- factor 10 * myt 1[i] };

  ### generate censoring range
  tmax <- max(myt 1, myt 2);
  tmin <- min(myt 1, myt 2);
  myc1 <- runif(nt, tmin, tmax);
  myc2 <- runif(nt, tmin, tmax);
  ### set censored data randomly, censored=Ci < ti (???)
  myT1st <- rep(1, nt);
  myT2st <- rep(1, nt);
  for (i in 1:nt) { if (myt 1[i] > myc1[i]) myT1st[i] <- 0 };
  for (i in 1:nt) { if (myt 2[i] > myc2[i]) myT2st[i] <- 0 };

  ### set group
  mygpT1 <- rep(0, nt);
  mygpT2 <- rep(1, nt);

  ### generate data frame (t, status, group)
  futime <- c(myt 1, myt 2);
  fustat <- c(myT1st, myT2st);
  mygroup <- c(mygpT1, mygpT2);
  mydata.df <- data.frame(futime, fustat, mygroup);
  ### use Cox model to fit the data
  myfit <- coxph(Surv(futime, fustat) ~ mygroup, data=mydata.df, x=T) ;
  mytemp <- cox.zph(myfit) ;

  ### see how many p-value are smaller than 0.05
  p1 <- mytemp$table[1,][3];
  if (p1 < 0.05) i p1 <- i p1 + 1;
  if (i i >= n i i) break;
}

i p1;
### below are only use for single step run
mydata.surv <- survfit(Surv(futime, fustat) ~ mygroup, mydata.df);
plot(mydata.surv, xlab="T", ylab="% survival", cex=2, lty=2:3);
title("Simulation result");
print(mytemp);
plot(mytemp);

```

Program 1.2: Simple log-rank test, Harrington-Fleming's weighted log-rank test, and test with the new weight function

```

### Simulation with nonproportional data, with 10 factors, from 0.1 to 1.0
### by Qing Xu, September 2004
#
### divide random data into three subgroups for frailty implantation

```

```

nt <- 1500; n1 <- nt/10; n2 <- n1*2; n3 <- n1*3;
n4 <- n1*4; n5 <- n1*5; n6 <- n1*6; n7 <- n1*7;
n8 <- n1*8; n9 <- n1*9; n10 <- n1*10

### each group is characterized by a factor, later, early, middle difference for each

factor1s <- c (1.00, 0.99, 0.85);
factor2s <- c (0.99, 0.95, 0.80);
factor3s <- c (0.98, 0.90, 0.75);
factor4s <- c (0.97, 0.85, 0.80);
factor5s <- c (0.95, 0.80, 0.85);
factor6s <- c (0.93, 0.80, 0.90);
factor7s <- c (0.91, 0.85, 0.93);
factor8s <- c (0.89, 0.90, 0.95);
factor9s <- c (0.87, 0.95, 0.97);
factor10s <- c (0.85, 0.99, 0.99);

### set how many set of data you want to simulate, here is 3 (late, early, middle)
nd <- 3;

### set baseline hazards function
rhos <- c (0.001, 0.1, 0.3);
### set how many rho values you want to simulate, here is 3
nr <- 3;

### parameter for new weighting function
etas <- c (5, 1, 0.01);
### set how many etas you want to simulate, here is 3
ne <- 3;

### number of simulations
n.iter <- 1000;

### parameters for H-F weighting function
ps <- c (0, 1, 1);
qs <- c (1, 0, 1);

### set a control number, nctrl=nd x nr x ne = 3 x 3 x 3=27
ictrl <- 0;
nctrl <- nd*nr*ne;

### set initial values for the recording numbers
hfgtnew <- rep(0, nctrl);
logrankgtnew <- rep(0, nctrl);
wilcoxgtnew <- rep(0, nctrl);
newlta <- rep(0, nctrl);
hflta <- rep(0, nctrl);
logranklta <- rep(0, nctrl);
wilcoxlta <- rep(0, nctrl);
pave.logrank <- rep(0, nctrl);
pave.wilcox <- rep(0, nctrl);
pave.hf <- rep(0, nctrl);
pave.new <- rep(0, nctrl);

### define a variable to identify the data type
datatype <- rep('aaaaa', nctrl);
### eta and rho values
etavalue <- rep(0, nctrl);
rhoval ue <- rep(0, nctrl);

### loops
for(id in 1:nd)
{
  ### use current factors
  factor1 <- factor1s[id];
  factor2 <- factor2s[id];
  factor3 <- factor3s[id];
  factor4 <- factor4s[id];
  factor5 <- factor5s[id];
  factor6 <- factor6s[id];
  factor7 <- factor7s[id];
  factor8 <- factor8s[id];

```

```

factor9 <- factor9s[id];
factor10 <- factor10s[id];

### set the right HF parameters, p, q
p <- ps[id];
q <- qs[id];

for(ir in 1:nr)
{
### use current rho
rho <- rhos[ir];
for(ie in 1:ne)
{
### use current eta
eta <- etas[ie];
### set initial values for p values
p.logrank <- rep(0,n.iter);
p.wilcox <- rep(0,n.iter);
p.hf <- rep(0,n.iter);
p.new <- rep(0,n.iter);

### increment the control number
ictrl <- ictrl + 1;

### main loop
for(it in 1:n.iter)
{
### generate survival data group one: t1
u <- sort(runif(nt));
t1 <- log(1-u)/rho*(-1);
t2 <- t1;
### generate survival data group two: t2
for(i in 1:n1) {t2[i] <- factor1*t1[i]};
for(i in (n1+1):n2) {t2[i] <- factor2*t1[i]};
for(i in (n2+1):n3) {t2[i] <- factor3*t1[i]};
for(i in (n3+1):n4) {t2[i] <- factor4*t1[i]};
for(i in (n4+1):n5) {t2[i] <- factor5*t1[i]};
for(i in (n5+1):n6) {t2[i] <- factor6*t1[i]};
for(i in (n6+1):n7) {t2[i] <- factor7*t1[i]};
for(i in (n7+1):n8) {t2[i] <- factor8*t1[i]};
for(i in (n8+1):n9) {t2[i] <- factor9*t1[i]};
for(i in (n9+1):n10) {t2[i] <- factor10*t1[i]};

### generate censoring range according to the data range
tmax <- max(t1,t2); # the upper limit of t
tmin <- min(t1,t2); # the lower limit of t
tc1 <- runif(nt,tmin,tmax);
tc2 <- runif(nt,tmin,tmax);

### set censored data randomly, c1[i], c2[i] =1 if censored
c1 <- rep(0,nt);
c2 <- rep(0,nt);
for(i in 1:nt) {if (t1[i] > tc1[i]) c1[i] <- 1};
for(i in 1:nt) {if (t2[i] > tc2[i]) c2[i] <- 1};

itmax <- nt*2;
time.temp<- sort(c(t1,t2),index.return=TRUE)
t<-time.temp$x; tindex<-time.temp$i;

### Y1(n.risk), d1(n.event), cc1(n.sensor), Y2, d2, c2
y1 <- rep(nt,itmax);
y2 <- rep(nt,itmax);
d1 <- rep(0,itmax);
d2 <- rep(0,itmax);
cc1 <- rep(0,itmax);
cc2 <- rep(0,itmax);

### calculate number at risk
for(i in 2:itmax)
{
if(tindex[i]<=nt)
{

```

```

        y2[i] <- y2[i - 1];
        if(c[i]==0) d1[i] <- 1;
        cc1[i] <- c[i];
        y1[i] <- y1[i - 1] - 1;
    }
    if(tindex[i]>nt)
    {
        y1[i] <- y1[i - 1];
        if(c[i]==0) d2[i] <- 1;
        cc2[i] <- c[i];
        y2[i] <- y2[i - 1] - 1;
    }
}

### calculate Y=Y1+Y2, d=d1+d2, Y1*(d/Y), d1-Y1*(d/Y)
y <- y1 + y2;
d <- d1 + d2;
y1dy <- y1*(d/y);
d1y1dy <- d1 - y1dy;

### big one
big <- rep(0, itmax);
for(i in 1:itmax)
{
    if(y[i] != y1[i])
        (big[i] <- (y1[i]/y[i])*(1-y1[i]/y[i])*((y[i]-d[i])/(y[i]-1))*d[i]));
};

### common survival function
ss <- (1-d/(y+1));
### common survival function during t(i-1) to t(i)
s1 <- rep(1, itmax);
### common survival function during t(i) to t(i+1)
s0 <- ss;
for (i in 2:itmax)
{
    s1[i] <- s1[i-1]*ss[i];
    s0[i] <- s0[i-1]*ss[i];
};

### simple log rank test, w(t)=1
zlogrank <- sum(d1y1dy)/((sum(big))^0.5);
if(zlogrank < 0) (p.logrank[it] <- 2*pnorm(zlogrank));
if(zlogrank > 0) (p.logrank[it] <- 2*(1-pnorm(zlogrank)));

### wilcoxon (Pet o-Pet o) logrank test, w(t)=s(t)
upper <- s0*d1y1dy;
lower <- s0*s0*big;
zwi cox <- sum(upper)/((sum(lower))^0.5);
if(zwi cox > 0) (p.wi cox[it] <- 2*(1-pnorm(zwi cox)));
if(zwi cox < 0) (p.wi cox[it] <- 2*pnorm(zwi cox));

### F-H weighting function, w(t)=s1^p*(1-s1)^q
hfw <- (s1^p)*((1-s1)^q)
upper hf <- hfw*d1y1dy;
lower hf <- hfw*hf*w*big;
zhf <- sum(upper hf)/((sum(lower hf))^0.5);
if(zhf > 0) (p.hf[it] <- 2*(1-pnorm(zhf)));
if(zhf < 0) (p.hf[it] <- 2*pnorm(zhf));

### new weight function, 1/2+(2*et a^2)/(2*et a-log[s])^2
weight <- 0.5 + (2*et a^2)/(2*et a-log(s0))^2;
upper new <- weight*d1y1dy;
lower new <- weight*weight*big;
znewweight <- sum(upper new)/((sum(lower new))^0.5);
if(znewweight < 0) (p.new[it] <- 2*pnorm(znewweight));
if(znewweight > 0) (p.new[it] <- 2*(1-pnorm(znewweight)));

### calculate how many p.hf > p.new
hfgtnew[ictrl] <- hfgtnew[ictrl] + 1*(p.hf[it] > p.new[it]);

### calculate how many p.logrank > p.new

```

```

logrankgtnew[ictrl] <- logrankgtnew[ictrl] + 1*(p.logrank[it] > p.new[it]);

### calculate how many wilcox > p.new
wilcoxgtnew[ictrl] <- wilcoxgtnew[ictrl] + 1*(p.wilcox[it] > p.new[it]);

### calculate how many pvalues are less than 0.05
newta[ictrl] <- newta[ictrl] + 1*(p.new[it] < 0.05);
hflta[ictrl] <- hflta[ictrl] + 1*(p.hf[it] < 0.05);
logranklta[ictrl] <- logranklta[ictrl] + 1*(p.logrank[it] < 0.05);
wilcoxlta[ictrl] <- wilcoxlta[ictrl] + 1*(p.wilcox[it] < 0.05);
}; # end of main loop

### record the control identification;
if(id == 1)(datatype[ictrl] <- 'late');
if(id == 3)(datatype[ictrl] <- 'early');
if(id == 2)(datatype[ictrl] <- 'middle');
rhoval ue[ictrl] <- rho;
etaval ue[ictrl] <- eta;

### average p-value for each test
pave.logrank[ictrl] <- sum(p.logrank)/n.iter;
pave.wilcox[ictrl] <- sum(p.wilcox)/n.iter;
pave.hf[ictrl] <- sum(p.hf)/n.iter;
pave.new[ictrl] <- sum(p.new)/n.iter;
}; # end of ie loop
}; # of ir loop
}; # end of id loop

### get the data and print;
result <- data.frame
(datatype, etaval ue, rhoval ue, pave.logrank, pave.wilcox, pave.hf, pave.new, logranklta, wilcoxlta, hflta, newta, hfgtnew, logrankgtnew, wilcoxgtnew);
result;

```

Program 1.3: Real data application, using cox.zph to test the proportional hazards assumption

```

# read in real data
dd <- read.table("d: xuqing/phdthesis/qingb14.txt", header=TRUE, sep=",")

# dd$group; #group 1 placebo, group 2 tamoxifen group
itmax <- 2817;
time.temp <- sort(dd$time, index.return=TRUE)
t <- time.temp$x; tindex <- time.temp$ix;
g <- rep(0, itmax); c <- g; e <- c;
for (i in 1:itmax)
{
  ii <- tindex[i];
  c[i] <- dd$ndi.event[ii];
  g[i] <- dd$group[ii];
  e[i] <- dd$event[ii];
}
newdata <- data.frame(g, e, t, c, tindex);
#newdata;
y1 <- 0; y2 <- 0; c1 <- 0; c2 <- 0;
for (i in 1:itmax)
{
  if (g[i] == 1)
  {
    y1 <- y1 + 1;
    c1 <- c1 + c[i];
  }
  if (g[i] == 2)
  {
    y2 <- y2 + 1;
    c2 <- c2 + c[i];
  }
}

```

```

y1; # number of objects in group 1
y2; # number of objects in group 2
c1; # observed events in group 1
c2; # observed events in group 2
sum(c); #total number of censored data

S1<-rep(1,itmax); S2<-S1; yy1<-rep(y1,itmax); yy2<-rep(y2,itmax);
if(g[1]==1)y1<-y1-1;
if(g[1]==2)y2<-y2-1;
for (i in 2:itmax)
{
  if (g[i]==1)
  {
    yy1[i]<-y1;
    yy2[i]<-yy2[i-1];
    S2[i]<-S2[i-1];
    if(c[i]==1) {S1[i]<-S1[i-1]*(1-1/y1);}
    if(c[i]==0) {S1[i]<-S1[i-1];}
    y1<-y1-1;
  }
  if (g[i]==2)
  {
    yy2[i]<-y2;
    yy1[i]<-yy1[i-1];
    S1[i]<-S1[i-1];
    if(c[i]==1) {S2[i]<-S2[i-1]*(1-1/y2);}
    if(c[i]==0) {S2[i]<-S2[i-1];}
    y2<-y2-1;
  }
}

dat a1<-dat a.frame(t, yy1, yy2, S1, S2);
dat a1;

#t<-t[1:1855]; SS1<-S1[1:1855]; SS2<-S2[1:1855];
#plot(tt, SS1, "S"); lines(tt, SS2, "S");
plot(t, S1, "S"); lines(t, S2, "S");

# use Cox model to fit the data
myfit<-coxph(Surv(t, c)~g, dat a=newdat a, x=T);
mytemp<-cox.zph(myfit);
print(mytemp);
plot(mytemp);

# see p-value
p1<-mytemp$table[1,][3];
p1;

mydat a.surv<-survfit(Surv(t, c)~g, newdat a);
plot(mydat a.surv, xlab="T", ylab="% survival", cex=2, lty=2:3);
title("Simulation result");
legend(c(5, 7.5), c(0.85, 0.72), c("group 1", "group 2"), lty=2:3);

```

Program 1.4: Real data application, using simple log-rank, Harrington-Fleming weight log-rank, new method to test the survival distribution difference

```

# read in real data
dd<-read.table("d:xuqing/phdthesi/qingb14.txt", header=TRUE, sep=",")

# dd$group; #group 1 placebo, group 2 tamoxifen group
itmax<-2817;
time.temp<-sort(dd$time, index.return=TRUE)
t<-time.temp$x; tindex<-time.temp$ix;
g<-rep(0,itmax); c<-g; e<-c;
for (i in 1:itmax)
{
  ii<-tindex[i];

```

```

c[i] <- dd$indi.event s[i i];
g[i] <- dd$group[i i];
e[i] <- dd$event s[i i];
}
newdat a<- dat a.f r a m e(g, e, t, c, t i n d e x);
#newdat a;
y1t <- 0; y2t <- 0; c1 <- 0; c2 <- 0;
for (i in 1:itmax)
{
  if (g[i] == 1)
  {
    y1t <- y1t + 1;
    c1 <- c1 + c[i];
  }
  if (g[i] == 2)
  {
    y2t <- y2t + 1;
    c2 <- c2 + c[i];
  }
}
y1; # number of objects in group 1
y2; # number of objects in group 2
c1; # observed events in group 1
c2; # observed events in group 2
sum(c); #total number of censored data

S1 <- rep(1, itmax); S2 <- S1; y1 <- rep(y1t, itmax); y2 <- rep(y2t, itmax);
d1 <- rep(0, itmax); d2 <- rep(0, itmax);
cc1 <- rep(0, itmax); cc2 <- rep(0, itmax);

for (i in 2:itmax)
{
  if (g[i] == 1)
  {
    y2[i] <- y2[i - 1];
    if (c[i] == 1) d1[i] <- 1;
    cc1[i] <- 1 - c[i];
    y1[i] <- y1[i - 1] - 1;
  }
  if (g[i] == 2)
  {
    y1[i] <- y1[i - 1];
    if (c[i] == 1) d2[i] <- 1;
    cc2[i] <- 1 - c[i];
    y2[i] <- y2[i - 1] - 1;
  }
}
for (i in 2:itmax)
{
  if (g[i] == 1)
  {
    S2[i] <- S2[i - 1];
    if (y1[i] != 0) S1[i] <- S1[i - 1] * (1 - d1[i] / y1[i]);
    if (y1[i] == 0) S1[i] <- S1[i - 1];
  }
  if (g[i] == 2)
  {
    S1[i] <- S1[i - 1];
    if (y2[i] != 0) S2[i] <- S2[i - 1] * (1 - d2[i] / y2[i]);
    if (y2[i] == 0) S2[i] <- S2[i - 1];
  }
}
}

plot(t, S1, "S"); lines(t, S2, "S");

### set the H-F parameters, p, q
p <- 0.5;
q <- 0.5;

### use current eta
eta <- 1;

```



```

### calculate  $Y=Y_1+Y_2$ ,  $d=d_1+d_2$ ,  $Y_1*(d/Y)$ ,  $d_1 - Y_1*(d/Y)$ 
y <- y1 + y2;
d <- d1 + d2;
cc<- cc1+cc2;
y1dy <- y1*(d/y);
d1y1dy <- d1 - y1dy;

### variance
big <- rep(0, itmax);
for(i in 1:itmax)
{
  if(y[i] != y1[i])
  (big[i] <- (y1[i]/y[i]) * (1-y1[i]/y[i]) * ((y[i]-d[i])/(y[i]-1)) * d[i]);
};

### common survival function
ss <- (1-d/(y+1));
### common survival function during t(i-1) to t(i)
s1 <- rep(1, itmax);
### common survival function during t(i) to t(i+1)
s0 <- ss;
for (i in 2:itmax)
{
  s1[i] <- s1[i-1]*ss[i];
  s0[i] <- s0[i-1]*ss[i];
};

dat a<- dat a. frame(t, c, g, d, y, cc, s0, s1, d1, y1, cc1, d2, y2, cc2, S1, S2)
dat a;

### simple log rank test,  $w(t)=1$ 
zlogrank <- sum(d1y1dy)/((sum(big))^0.5);
if(zlogrank < 0) (p.logrank <- 2*pnorm(zlogrank));
if(zlogrank > 0) (p.logrank <- 2*(1-pnorm(zlogrank)));

### wilcoxon (Pet o- Pet o) logrank test,  $w(t)=s(t)$ 
upper <- s0*d1y1dy;
lower <- s0*s0*big;
zwilcox <- sum(upper)/((sum(lower))^0.5);
if(zwilcox > 0) (p.wilcox <- 2*(1-pnorm(zwilcox)));
if(zwilcox < 0) (p.wilcox <- 2*pnorm(zwilcox));

### F- H wei ght i ng f unct i on,  $w(t)=s^2p*(1-s)^q$ 
hfw <- (s1^p)*((1-s1)^q)
upper hf <- hfw*d1y1dy;
lower hf <- hfw*hf*w*big;
zhf <- sum(upper hf)/((sum(lower hf))^0.5);
if(zhf > 0) (p.hf <- 2*(1-pnorm(zhf)));
if(zhf < 0) (p.hf[it] <- 2*pnorm(zhf));

### new wei ght f unct i on,  $1/2+(2*et a^2)/(2*et a-l og[ s ])^2$ 
weight <- 0.5 + (2*et a^2)/(2*et a-l og( s0))^2;
upper new <- weight*d1y1dy;
lower new <- weight*weight*big;
znewweight <- sum(upper new)/((sum(lower new))^0.5);
if(znewweight < 0) (p.new <- 2*pnorm(znewweight));
if(znewweight > 0) (p.new <- 2*(1-pnorm(znewweight)));

# p- values
p.new; p.hf; p.wilcox; p.logrank;

```

Program 1.5: Plot of new weight function as a function of parameter ψ

```

# set survival time [0, 25]
t <- seq(0, 25, by=0.5)
# for exponential distributions,
rho<-0.3
s<-exp(-rho*t)

```

```

# weight function
phi <- 5
w1<- 0.5+2*phi*phi/(2*phi-log(s))^2
phi <- 1
w2<- 0.5+2*phi*phi/(2*phi-log(s))^2
phi <- 0.1
w3<- 0.5+2*phi*phi/(2*phi-log(s))^2
plot(t, w3, "l", main="weight function vs. survival time")
lines(t, w2)
lines(t, w1)
legend(16, 0.75, "phi=5", bty="n")
legend(2.5, 0.75, "phi=1", bty="n")
legend(-0.5, 0.7, "phi=0.1", bty="n")
legend(10, 0.95, "Exponential distribution, baseline rho=0.3", bty="n")

```

Program 1.6: Example of diagnostic plot of survival data with inverse Gaussian frailty.

```

### set parameter xi and beta
xi <- 1000; # constant, mean
lambda <- 10; # parameter
nt <- 2000;
beta <- 2;

### generate survival data following Weibull distribution
u1 <- sort(runif(nt)); #sorted uniform random data
t <- (-1)*xi*log(1-u1); #convert into survival time
s <- (t/xi)^beta;
S <- exp(lambda*(1-sqrt(2*s/lambda+1)));

plot(log(t), log(-log(S)))
legend(2, 0, "lambda=10", bty="n")

```

Program 1.7: Example of diagnostic plot of survival data with gamma frailty.

```

### set parameter xi and beta
xi <- 1000; # constant, mean
lambda <- 0.1; # parameter
nt <- 2000;
beta <- 2;

### generate survival data following Weibull distribution
u1 <- sort(runif(nt)); #sorted uniform random data
t <- (-1)*xi*log(1-u1); #convert into survival time
s <- (t/xi)^beta;
S <- (1+s/lambda)^(-lambda);

plot(log(t), -log(S))
legend(2, 0, "lambda=10", bty="n")

```

Program 1.8: Simulations on the MLE of inverse Gaussian frailty parameters.

```

# Use MLE to estimate the IG parameters of the frailty
# for exponential survival data with explicit frailty implanted
#
### divide random data into 10 groups for non proportional hazard ratio
nt <- 200; # sample size
n <- nt/10; # subgroup size
f <- c(0.85, 0.8, 0.75, 0.8, 0.85, 0.9, 0.93, 0.95, 0.97, 0.99); #early difference
# f <- c(0.99, 0.95, 0.9, 0.85, 0.8, 0.8, 0.85, 0.9, 0.95, 0.99); #middle difference
# f <- c(1, 0.99, 0.98, 0.97, 0.95, 0.93, 0.91, 0.89, 0.87, 0.85); #late difference
theta <- c(rep(f[1], n), rep(f[2], n), rep(f[3], n), rep(f[4], n), rep(f[5], n),
rep(f[6], n), rep(f[7], n), rep(f[8], n), rep(f[9], n), rep(f[10], n));
### set baseline hazard function h_0(t) = rho
rho <- 0.3;
### set frailty distribution parameter, lambda of IG distribution

```

```

nu<- 1.0; lambda<- 1;      #nu=mean, lambda=2*phi, variance=1/lambda
### set number of simulations
nii<- 1000;
### set loop index
ii<- 0;
### censor level
censlvl <- 0;
t0<- 0; tm<- 0;

lambda.cum<- rep(0, nii);
repeat{
  ii<- ii+1;
  ### generate random uniform numbers
  u<- sort(runif(nt));
  ### generate frailty w following LG distribution
  w<- sort(rinvGauss(nt, nu, lambda), decreasing=FALSE);
  ### generate exponential times with frailty implanted
  t1<- (-1)*log(1-u)/(rho*w);
  t2<- t1*theta; #implant difference
  ### generate censoring
  if(censlvl>0){
    tc1 <- (runif(nt, min=t0, max=tm));
    tc2 <- (runif(nt, min=t0, max=tm));
    for(i in 1:nt)
      {
        if (t1[i] > tc1[i]) c1[i] <- 1;
        if (t2[i] > tc2[i]) c2[i] <- 1;
      };
  };
  ### survival time table
  tb<- sort(c(t1,t2), index.return=TRUE);
  t<- tb$x; tindex<- tb$ix;
  itmax<- nt*2; # total number of data points
  ### calculate survival function using Kaplan-Meier method
  S1<- rep(1, itmax); S2<- rep(1, itmax);
#   yy1<- rep(nt, itmax); yy2<- rep(nt, itmax);
  if(tindex[1]<=nt) {y1<- nt- 1; y2<- nt;}
  if(tindex[1]>nt) {y1<- nt; y2<- nt- 1;}
  for (i in 2:itmax)
  {
#     if(tindex[i]<=nt){
#       yy1[i]<- y1;
#       yy2[i]<- yy2[i- 1];
#       S2[i]<- S2[i- 1];
#       if(censlvl>0){
#         if(c1[i]==0) {S1[i]<- S1[i- 1]*(1- 1/y1);}
#         if(c1[i]==1) {S1[i]<- S1[i- 1];}
#       }
#       if(censlvl==0){S1[i]<- S1[i- 1]*(1- 1/y1);}
#       y1<- y1- 1;
#     }
#     if(tindex[i]>nt){
#       yy2[i]<- y2;
#       yy1[i]<- yy1[i- 1];
#       S1[i]<- S1[i- 1];
#       if(censlvl>0){
#         if(c2[i]==0) {S2[i]<- S2[i- 1]*(1- 1/y2);}
#         if(c2[i]==1) {S2[i]<- S2[i- 1];}
#       }
#       if(censlvl==0){S2[i]<- S2[i- 1]*(1- 1/y2);}
#       y2<- y2- 1;
#     }
  }

  ### calculate w[i]
  w.hat <- rep(0, itmax); ww<- rep(0, itmax)
  for(i in 1:itmax){
    theta.temp<- theta[trunc(i/2)];
    if(S1[i]<1 && S2[i]<1 && S1[i]>0 && S2[i]>0){
      w.hat[i]<- log(S2[i])/(theta.temp*log(S1[i]));
      ww[i]<- (w.hat[i]- 1)^2/w.hat[i];
    }
  }
}

```

```

    }
#   dat a1<- dat a. frame(t , S1, S2, w. hat , ww) ;
#   dat a1;

### estimate of lambda from MLE

lambda. hat <- nt / sum( ww) ;
#   lambda. hat ;
lambda. cum[ i i ] <- lambda. hat
### estimate of phi = half lambda. hat
if ( i i >= n i i ) break ;
}
lambda. cum
mean( lambda. cum) ;
var( lambda. cum) ;

```

Program 1.9: Simulations on the MLE of gamma frailty parameters.

```

# Use MLE to estimate the gamma parameters of the frailty
# for exponential survival data with explicit frailty implanted
#
### divide random data into 10 groups for non proportional hazard ratio
nt <- 200;          # sample size
n <- nt / 10;      # subgroup size
# f <- c(0.85, 0.8, 0.75, 0.8, 0.85, 0.9, 0.93, 0.95, 0.97, 0.99); #early difference
# f <- c(0.99, 0.95, 0.9, 0.85, 0.8, 0.8, 0.85, 0.9, 0.95, 0.99); #middle difference
f <- c(1, 0.99, 0.98, 0.97, 0.95, 0.93, 0.91, 0.89, 0.87, 0.85); #late difference
theta <- c(rep(f[1], n), rep(f[2], n), rep(f[3], n), rep(f[4], n), rep(f[5], n),
          rep(f[6], n), rep(f[7], n), rep(f[8], n), rep(f[9], n), rep(f[10], n));
### set baseline hazard function h_0(t)=rho
rho <- 0.001;
### set frailty distribution parameter,
kappa <- 1.0; scale <- 1/kappa; #shape=kappa, scale=rho=1/kappa
### set number of simulations
nii <- 1000;
### set loop index
ii <- 0;
### censor level
censorlvl <- 0;
t0 <- 0; tm <- 0;

kappa. cum <- rep(0, nii);
repeat {
  ii <- ii + 1;
  ### generate random uniform numbers
  u <- sort(runif(nt));
  ### generate frailty following gamma distribution
  w <- sort(rgamma(nt, kappa, scale), decreasing=FALSE);
  ### generate exponential times with frailty implanted
  t1 <- (-1)*log(1-u)/(rho*w);
  t2 <- t1*theta; #implant difference
  ### generate censoring
  if(censorlvl > 0){
    tc1 <- (runif(nt, min=t0, max=tm));
    tc2 <- (runif(nt, min=t0, max=tm));
    for(i in 1:nt)
      {
        if (t1[i] > tc1[i]) c1[i] <- 1;
        if (t2[i] > tc2[i]) c2[i] <- 1;
      };
  };
  ### survival time table
  tb <- sort(c(t1, t2), index.return=TRUE);
  t <- tb$x; tindex <- tb$ix;
  itmax <- nt*2; # total number of data points
  ### calculate survival function using Kaplan-Meier method
  S1 <- rep(1, itmax); S2 <- rep(1, itmax);
#   yy1 <- rep(nt, itmax); yy2 <- rep(nt, itmax);

```

```

if(tindex[1]<=nt) {y1<-nt-1; y2<-nt;}
if(tindex[1]>nt) {y1<-nt; y2<-nt-1;}
for (i in 2:itmax)
{
  if(tindex[i]<=nt){
#     yy1[i]<-y1;
#     yy2[i]<-yy2[i-1];
    S2[i]<-S2[i-1];
    if(censorlvl>0){
      if(c1[i]==0) {S1[i]<-S1[i-1]*(1-1/y1);}
      if(c1[i]==1) {S1[i]<-S1[i-1];}
    }
    if(censorlvl==0){S1[i]<-S1[i-1]*(1-1/y1);}
    y1<-y1-1;
  }
  if(tindex[i]>nt){
#     yy2[i]<-y2;
#     yy1[i]<-yy1[i-1];
    S1[i]<-S1[i-1];
    if(censorlvl>0){
      if(c2[i]==0) {S2[i]<-S2[i-1]*(1-1/y2);}
      if(c2[i]==1) {S2[i]<-S2[i-1];}
    }
    if(censorlvl==0){S2[i]<-S2[i-1]*(1-1/y2);}
    y2<-y2-1;
  }
}

### calculate w[i]
w.hat<-rep(0,itmax); logw<-rep(0,itmax)
for(i in 1:itmax){
  theta.temp<-theta[trunc(i/2)];
  if(S1[i]<1 && S2[i]<1 && S1[i]>0 && S2[i]>0){
    w.hat[i]<-log(S2[i])/(theta.temp*log(S1[i]));
    logw[i]<-log(w.hat[i]);
  }
}

dat a1<-data.frame(t, S1, S2, w.hat, logw);
#   dat a1;

### estimate of kappa from MLE
a<-sum(logw); b<-sum(w.hat); c<-(a-b)/itmax;
### solve the nonlinear equation for kappa using Newton type algorithm
f<-function(ka) (c+log(ka)+1-digamma(ka))^2 #target function to minimize
aa<-nlm(f, 1); #minimization using nlm
if(aa$minimum[1]<1e-5){kappa.hat<-aa$estimate[1];}
kappa.cumf[i]<-kappa.hat
if(i>=nii) break;
}
kappa.cum;
mean(kappa.cum);
var(kappa.cum);

```

Program 1.10: Simple log-rank test, Harrington-Fleming's weighted log-rank test, test with the new weight function, test with the new weight function supreme version

```

### divide random data into three subgroups for frailty implantation
nt <- 1500; n1 <- nt/10; n2 <- n1*2; n3 <- n1*3;
n4 <- n1*4; n5 <- n1*5; n6 <- n1*6; n7 <- n1*7;
n8 <- n1*8; n9 <- n1*9; n10 <- n1*10

### each group is characterized by a factor, later, early, middle difference for each
factor1s <- c(1.00, 0.99, 0.85);
factor2s <- c(0.99, 0.95, 0.80);
factor3s <- c(0.98, 0.90, 0.75);
factor4s <- c(0.97, 0.85, 0.80);
factor5s <- c(0.95, 0.80, 0.85);
factor6s <- c(0.93, 0.80, 0.90);
factor7s <- c(0.91, 0.85, 0.93);

```

```

factor8s <- c (0.89, 0.90, 0.95);
factor9s <- c (0.87, 0.95, 0.97);
factor10s<- c (0.85, 0.99, 0.99);
#factor1s <- c (0.99, 0.775, 0.99);
#factor2s <- c (0.975, 0.8, 0.975);
#factor3s <- c (0.95, 0.825, 0.95);
#factor4s <- c (0.925, 0.85, 0.925);
#factor5s <- c (0.90, 0.875, 0.9);
#factor6s <- c (0.875, 0.90, 0.9);
#factor7s <- c (0.85, 0.925, 0.925);
#factor8s <- c (0.825, 0.95, 0.95);
#factor9s <- c (0.8, 0.975, 0.975);
#factor10s<- c (0.775, 0.99, 0.99);

### set how many set of data you want to simulate, here is 3 (late, early, middle)
nd <- 3;

### set baseline hazards function
rhos <- c (0.001, 0.1, 0.3);
### set how many rho values you want to simulate, here is 3
nr <- 3;

### parameter for new weighting function
etas <- c (5, 1, 0.01);
### set how many etas you want to simulate, here is 3
ne <- 3;

### number of simulations
n.iter <- 1000;

### parameters for H-F weighting function
ps <- c (0, 2, 1);
qs <- c (2, 0, 1);

### set the groups of t
itmax <- 20;

### set a control number, nctrl=nd x nr x ne = 3 x 3 x 3=27
ictrl <- 0;
nctrl <- nd*nr*ne;

### set initial values for the recording numbers
hfgtnew <- rep(0, nctrl);
logrankgtnew <- rep(0, nctrl);
wilcoxgtnew <- rep(0, nctrl);
newlta <- rep(0, nctrl);
hflta <- rep(0, nctrl);
logranklta <- rep(0, nctrl);
wilcoxlta <- rep(0, nctrl);
pave.logrank <- rep(0, nctrl);
pave.wilcox <- rep(0, nctrl);
pave.hf <- rep(0, nctrl);
pave.new <- rep(0, nctrl);
newsuplta <- rep(0, nctrl);
pave.new.sup <- rep(0, nctrl);

### define a variable to identify the data type
#datatype <- rep('aaaaa', nctrl);
### eta and rho values
etavalue <- rep(0, nctrl);
rhoval ue <- rep(0, nctrl);

### loops
for(id in 1:nd)
{
  ### use current factors
  factor1 <- factor1s[id];
  factor2 <- factor2s[id];
  factor3 <- factor3s[id];
  factor4 <- factor4s[id];
  factor5 <- factor5s[id];
  factor6 <- factor6s[id];

```

```

factor7 <- factor7s[id];
factor8 <- factor8s[id];
factor9 <- factor9s[id];
factor10 <- factor10s[id];

### set the right HF parameters, p, q
p <- ps[id];
q <- qs[id];

for(ir in 1:nr)
{
  ### use current rho
  rho <- rhos[ir];
  for(ie in 1:ne)
  {
    ### use current eta
    eta <- etas[ie];
    ### set initial values for p values
    p.logrank <- rep(0,n.iter);
    p.wilcox <- rep(0,n.iter);
    p.hf <- rep(0,n.iter);
    p.new <- rep(0,n.iter);
    p.new.sup <- rep(0,n.iter);

    ### increment the control number
    ictrl <- ictrl + 1;

    ### main loop
    for (it in 1:n.iter)
    {
      ### generate survival data group one: t1
      u <- sort(runif(nt));
      t1 <- log(1-u)/rho*(-1);
      t2 <- t1;
      ### generate survival data group two: t2
      for (i in 1:n1) {t2[i] <- factor1*t1[i]};
      for (i in (n1+1):n2) {t2[i] <- factor2*t1[i]};
      for (i in (n2+1):n3) {t2[i] <- factor3*t1[i]};
      for (i in (n3+1):n4) {t2[i] <- factor4*t1[i]};
      for (i in (n4+1):n5) {t2[i] <- factor5*t1[i]};
      for (i in (n5+1):n6) {t2[i] <- factor6*t1[i]};
      for (i in (n6+1):n7) {t2[i] <- factor7*t1[i]};
      for (i in (n7+1):n8) {t2[i] <- factor8*t1[i]};
      for (i in (n8+1):n9) {t2[i] <- factor9*t1[i]};
      for (i in (n9+1):n10) {t2[i] <- factor10*t1[i]};

      ### generate censoring range according to the data range
      tmax <- max(t1,t2); # the upper limit of t
      tmin <- min(t1,t2); # the lower limit of t
      tc1 <- runif(nt,tmin,tmax);
      tc2 <- runif(nt,tmin,tmax);

      ### set censored data randomly, c1[i], c2[i] =1 if censored
      c1 <- rep(0,nt);
      c2 <- rep(0,nt);
      for(i in 1:nt) {if (t1[i] > tc1[i]) c1[i] <- 1};
      for(i in 1:nt) {if (t2[i] > tc2[i]) c2[i] <- 1};

      ### calculate dt according to tmin,tmax, and itmax
      dt <- (tmax-tmin)/itmax;

      ### set time increment from t=0 to t=itmax
      t <- rep(0,itmax);
      for (i in 1:itmax) { t[i] <- (i-1)*dt };

      ### initialize Y1(n.risk), d1(n.event), cc1(n.sensor), Y2, d2, c2
      y1 <- rep(nt,itmax);
      y2 <- rep(nt,itmax);
      d1 <- rep(0,itmax);
      d2 <- rep(0,itmax);
      cc1 <- rep(0,itmax);
      cc2 <- rep(0,itmax);
    }
  }
}

```

```

### convert data into two sample table (like Table 7.2 at p196)
for (i in 1:nt)
{
  iia <- trunc(t1[i]/dt)+1; ### truncate t1, t2 to integers
  iib <- trunc(t2[i]/dt)+1;
  if(c1[i] == 0){d1[iia] <- d1[iia] + 1}; ### integer values as index
  if(c2[i] == 0){d2[iib] <- d2[iib] + 1}; ### to increment n.event
  cc1[iia] <- cc1[iia] + c1[i]; ### increment n.censor
  cc2[iib] <- cc2[iib] + c2[i];
};

### calculate Y (n.risk) according to n.event and n.censor
for (i in 2:itmax)
{
  y1[i] <- y1[i-1] - d1[i-1] - cc1[i-1];
  y2[i] <- y2[i-1] - d2[i-1] - cc2[i-1];
};

### calculate Y=Y1+Y2, d=d1+d2, Y1*(d/Y), d1-Y1*(d/Y)
y <- y1 + y2;
d <- d1 + d2;
y1dy <- y1*(d/y);
d1y1dy <- d1 - y1dy;

### big one
big <- rep(0,itmax);
for(i in 1:itmax)
{
  if(y[i] != y1[i])
  (big[i] <- (y1[i]/y[i])*(1-y1[i]/y[i])*((y[i]-d[i])/(y[i]-1))*d[i]);
};

### common survival function
ss <- (1-d/(y+1));
### common survival function during t(i-1) to t(i)
s1 <- rep(1,itmax);
### common survival function during t(i) to t(i+1)
s0 <- ss;
for (i in 2:itmax)
{
  s1[i] <- s1[i-1]*ss[i];
  s0[i] <- s0[i-1]*ss[i];
};

### simple log rank test, w(t)=1
zlgrank <- sum(d1y1dy)/((sum(big))^0.5);
if(zlgrank <= 0) (p.lgrank[it] <- 2*pnorm(zlgrank));
if(zlgrank > 0) (p.lgrank[it] <- 2*(1-pnorm(zlgrank)));

### wilcoxon (Pet o- Pet o) logrank test, w(t)=s(t)
upper <- s0*d1y1dy;
lower <- s0*s0*big;
zwilcox <- sum(upper)/((sum(lower))^0.5);
if(zwilcox > 0) (p.wilcox[it] <- 2*(1-pnorm(zwilcox)));
if(zwilcox <= 0) (p.wilcox[it] <- 2*pnorm(zwilcox));

### F-H weighting function, w(t)=s1^p*(1-s1)^q
hfw <- (s1^p)*((1-s1)^q)
upperhf <- hfw*d1y1dy;
lowerhf <- hfw*hf*big;
zhf <- sum(upperhf)/((sum(lowerhf))^0.5);
if(zhf > 0)(p.hf[it] <- 2*(1-pnorm(zhf)));
if(zhf <= 0)(p.hf[it] <- 2*pnorm(zhf));

### new weight function, 1/2+(2*et a^2)/(2*et a-log[s])^2
weight <- 0.5 + (2*et a^2)/(2*et a-log(s0))^2;
uppernew <- weight*d1y1dy;
lowernew <- weight*weight*big;
znewweight <- sum(uppernew)/((sum(lowernew))^0.5);
if(znewweight <= 0)(p.new[it] <- 2*pnorm(znewweight));
if(znewweight > 0)(p.new[it] <- 2*(1-pnorm(znewweight)));

```



```

### supreme new weight function method
Tn <- rep(1, itmax);
for (i in 1:itmax)
{
  Tn[i] <- sum(upper new[1:i]) / ((sum(lower new[1:itmax]))^0.5);
};
abs.Tn <- abs(Tn);
sup.Tn <- max(abs.Tn);
if(sup.Tn <= 0)(p.new.sup[it] <- 2*pnorm(sup.Tn));
if(sup.Tn > 0)(p.new.sup[it] <- 2*(1-pnorm(sup.Tn)));

### calculate how many p.hf > p.new
hfgtnew[ictrl] <- hfgtnew[ictrl] + 1*(p.hf[it] > p.new[it]);

### calculate how many p.logrank > p.new
logrankgtnew[ictrl] <- logrankgtnew[ictrl] + 1*(p.logrank[it] > p.new[it]);

### calculate how many p.wilcox > p.new
wilcoxgtnew[ictrl] <- wilcoxgtnew[ictrl] + 1*(p.wilcox[it] > p.new[it]);

### calculate how many p values are less than 0.05
newlta[ictrl] <- newlta[ictrl] + 1*(p.new[it] < 0.05);
hflta[ictrl] <- hflta[ictrl] + 1*(p.hf[it] < 0.05);
logranklta[ictrl] <- logranklta[ictrl] + 1*(p.logrank[it] < 0.05);
wilcoxlta[ictrl] <- wilcoxlta[ictrl] + 1*(p.wilcox[it] < 0.05);
newsuplta[ictrl] <- newsuplta[ictrl] + 1*(p.new.sup[it] < 0.05);
}; # end of main loop

### record the control identification;
if(id == 1)(datatype[ictrl] <- 'late');
if(id == 2)(datatype[ictrl] <- 'middle');
if(id == 3)(datatype[ictrl] <- 'early');
rhoval[ictrl] <- rho;
etaval[ictrl] <- eta;

### average p-value for each test
pave.logrank[ictrl] <- sum(p.logrank)/n.iter;
pave.wilcox[ictrl] <- sum(p.wilcox)/n.iter;
pave.hf[ictrl] <- sum(p.hf)/n.iter;
pave.new[ictrl] <- sum(p.new)/n.iter;
pave.new.sup[ictrl] <- sum(p.new.sup)/n.iter;
}; # end of ie loop
}; # of ir loop
}; # end of id loop

### get the data and print;
result <- data.frame
(datatype, etaval, rhoval, pave.logrank, pave.wilcox, pave.hf, pave.new, logranklta, wilcoxlta, hflta, newlta, newsuplta);
#result1 <- data.frame(hfgtnew, logrankgtnew, wilcoxgtnew);
result;

```

Programs for new method on testing the proportional odds assumption

Program 2.1: Test the proportional odds assumption using Dauxois-Kirmani method

```
### set parameter xi and beta
xi <- 150;          # constant, mean
beta <- 1;         # parameter
rbeta <- 1.0/beta; # rbeta=1/beta
censlvl <- 0.1     # censoring level
if(censlvl==0.1){
  t0<-2; tm<-10;
}
if(censlvl==0.3){
  t0<-0.2; tm<-5;
}

### number of simulations, number of data point in each group
n.iter <- 200;      # number of simulations
nt1 <- 50; nt2 <- 60; # number of data points in each group
siglvl2 <- 0.02; siglvl5 <- 0.05;

### set the groups of t
itmax <- nt1+nt2;  # number of time intervals

### set initial values of the p-value and power for each simulation
p.zs<-rep(0,n.iter); # array for storing p values
p.sig2<-rep(0,n.iter); # array for storing power
p.sig5<-rep(0,n.iter);
clevel <- 0;

### main loop
for (it in 1:n.iter)
{
  ### generate survival data group one: t1
  u1 <- sort(runif(nt1)); #sorted uniform random data
  t1 <- xi/(1/u1-1);      #convert into survival time
  u2 <- sort(runif(nt2));
  t3 <- xi/(1/u2-1);
  t2 <- (xi^(1-rbeta))*(t3^rbeta);

  ### set censored data randomly, c1[i], c2[i] =1 if censored
  c1 <- rep(0,nt1); c2 <- rep(0,nt2);
  r1 <- runif(nt1); r2 <- runif(nt2);
  for(i in 1:nt1)
  {
    if (r1[i] <= censlvl) c1[i] <- 1;
  };
  for(i in 1:nt2)
  {
    if (r2[i] <= censlvl) c2[i] <- 1;
  };

  ### set censored data randomly, c1[i], c2[i] =1 if censored
  c1 <- rep(0,nt1); c2 <- rep(0,nt2);
  if(censlvl>0){
    tc1 <- (runif(nt1, min=t0, max=tm)); #if need sorting?
    tc2 <- (runif(nt2, min=t0, max=tm));
    for(i in 1:nt1)
    {
      if (t1[i] > tc1[i]) c1[i] <- 1;
    };
    for(i in 1:nt2)
    {
      if (t2[i] > tc2[i]) c2[i] <- 1;
    };
  };
};
```

```

### set t intervals
t<-sort(c(t1,t2));

### Y1(n.risk), d1(n.event), cc1(n.sensor), Y2, d2, c2
y1 <- rep(nt1,itmax); y2 <- rep(nt2,itmax);

### convert data into two sample table (like Table 7.2 at p196)
d1<-rep(0,itmax); d2<-rep(0,itmax); #initial value for d1,d2
cc1<-rep(0,itmax); cc2<-rep(0,itmax); #initial value for cc1,cc2

indext1<-rep(0,nt1); #initial value for index of t1
for (i in 1:nt1)
{
  for (j in 1:itmax)
  {
    if(t1[i]>=t[j]){indext1[i]<-j} #compare with t to get index
  };
};

for (i in 1:nt1) # get d1,cc1
{
  ### to increment n.event
  if(c1[i] == 0){d1[indext1[i]] <- d1[indext1[i]] + 1};
  cc1[indext1[i]] <- cc1[indext1[i]] + c1[i];
};

indext2<-rep(0,nt2); #initial value for index of t2
for (i in 1:nt2)
{
  for (j in 1:itmax)
  {
    #compare with t to get index
    if(t2[i]>=t[j]){indext2[i]<-j}
  };
};

for (i in 1:nt2) # get d2,cc2
{
  ### to increment n.event
  if(c2[i] == 0){d2[indext2[i]] <- d2[indext2[i]] + 1};
  cc2[indext2[i]] <- cc2[indext2[i]] + c2[i];
};

### calculate Y (n.risk) according to n.event and n.censor
for (i in 2:itmax)
{
  y1[i] <- y1[i-1] - d1[i-1] - cc1[i-1];
  y2[i] <- y2[i-1] - d2[i-1] - cc2[i-1];
};

### calculate Y=Y1+Y2, d=d1+d2, Y1*(d/Y), d1-Y1*(d/Y)
y <- y1 + y2; d <- d1 + d2; cc <- cc1 + cc2;

### K-M estimate of survival functions
ss1<-rep(1,itmax); ss2<-rep(1,itmax);
for (i in 1:itmax)
{
  if(y1[i]!=0) ss1[i]<-(1-d1[i]/y1[i]);
  if(y2[i]!=0) ss2[i]<-(1-d2[i]/y2[i]);
};
S1<-rep(1,itmax); S2<-rep(1,itmax);
S1[1]<-ss1[1]; S2[1]<-ss2[1];
phi1<-rep(0,itmax); phi2<-rep(0,itmax);
for (i in 2:itmax)
{
  S1[i]<-S1[i-1]*ss1[i]; S2[i]<-S2[i-1]*ss2[i];
}

### phi, the survival odds
for (i in 1:itmax)

```

```

{
  if (S1[i] != 1) phi 1[i] <- S1[i] / (1 - S1[i]);
  if (S2[i] != 1) phi 2[i] <- S2[i] / (1 - S2[i]);
};

### F
F1 <- 1 - S1; F2 <- 1 - S2;

### K1 K2
K1 <- ((nt 1 + nt 2) / (nt 1 * nt 2)) * y1 * y2 / y;
K2 <- y1 * y2 / nt 1 / nt 2;

### set interval (overlap region)
rr <- rep(0, itmax);
for (i in 1:itmax)
{
  if ((phi 1[i] * phi 2[i]) != 0) rr[i] <- 1;
};

psi 11 <- sum(K1 * phi 1 * rr); psi 12 <- sum(K1 * phi 2 * rr);
psi 21 <- sum(K2 * phi 1 * rr); psi 22 <- sum(K2 * phi 2 * rr);
dat aset <- data.frame(t, dt, y, d, cc, y1, d1, cc1, y2, d2, cc2, S1, S2, rr);

### gamma
GAMMA <- psi 11 * psi 22 - psi 12 * psi 21;

### vij calculation
### initial values of vij
v11 <- 0; v12 <- 0; v21 <- 0; v22 <- 0;
for (kt in 1:itmax)
{
  for (ks in 1:itmax)
  {
    nu <- min(ks, kt);
    dbyy1 <- 0; dbyy2 <- 0;
    temp1 <- 0; temp2 <- 0;
    for (ku in 1:nu)
    {
      if (y1[ku] != 0) dbyy1 <- dbyy1 + d1[ku] / (y1[ku] * y1[ku]);
      if (y2[ku] != 0) dbyy2 <- dbyy2 + d2[ku] / (y2[ku] * y2[ku]);
    };
    if ((F1[ks] != 0) & (F1[kt] != 0)) temp1 <- nt 1 * dbyy1 / (F1[ks] * F1[kt]);
    if ((F2[ks] != 0) & (F2[kt] != 0)) temp2 <- nt 2 * dbyy2 / (F2[ks] * F2[kt]);
    tmp <- phi 1[ks] * phi 2[kt] * (temp1 + temp2);
    v11 <- v11 + K1[kt] * K1[ks] * tmp;
    v12 <- v12 + K1[kt] * K2[ks] * tmp;
    v21 <- v21 + K2[kt] * K1[ks] * tmp;
    v22 <- v22 + K2[kt] * K2[ks] * tmp;
  }; # end of ks loop
}; # end of kt loop

### variance
var 2 <- psi 22 * psi 21 * v11 - psi 22 * psi 11 * v12 - psi 12 * psi 21 * v21 + psi 12 * psi 11 * v22;

### statistics
zs <- 0;
if (var 2 != 0) zs <- sqrt(nt 1 + nt 2) * GAMMA / sqrt(var 2);

### p-value
if (zs > 0) (p.zs[it] <- 2 * (1 - pnorm(zs)));
if (zs < 0) (p.zs[it] <- 2 * pnorm(zs));
if (p.zs[it] < siglvl5) (p.sig5[it] <- 1);
if (p.zs[it] < siglvl2) (p.sig2[it] <- 1);
cl level <- cl level + (sum(c1) + sum(c2)) / (nt 1 + nt 2);
}; # end of main loop

# print error if crash
if (it < n.iter) {
  print('it < n.iter'); it;
  dat aset; dat aset 1;
  GAMMA; var 2; v11; v12; v21; v22;
  psi 11; psi 12; psi 21; psi 22;
}

```

```

};

### average p-value
pave.zs<- sum(p.zs)/n.iter;

### power
power2<- sum(p.sig2)/n.iter;
power5<- sum(p.sig5)/n.iter;
pave.zs; power2; power5;

### check censoring level
cl level /n.iter;

### following is only for single step run
#i f (zs>0) p<- 2*(1-pnorm(zs));
#i f (zs<=0) p<- 2*pnorm(zs);
#zs; p;
#time<- c(0,t); SS1<- c(1,S1); SS2<- c(1,S2);
#plot(time,SS1,"S")
#lines(time,SS2,"S","red")
# censoring level
#sum(c1)/nt1;
#sum(c2)/nt2;

```

Program 2.2: Test the proportional odds assumption using extended method

```

### Simulation with nonproportional odds data
# log-logistic distribution survival data
# parametric weight function is used

### set parameter xi and beta
xi <- 1; # constant, mean
beta <- 1; # parameter
rbeta <- 1.0/beta; # rbeta=1/beta
censlvl <- 0.1 # censoring level
i f (censlvl ==0.1){
  t0<-2; tm<-10;
}
i f (censlvl ==0.3){
  t0<-0.2; tm<-5;
}

### number of simulations, number of data point in each group
n.iter <- 1000; # number of simulations
nt1 <- 50; nt2 <- 60; # number of data points in each group
siglvl2 <- 0.02; siglvl5 <- 0.05;
rho<-nt1/(nt1+nt2);

### set the groups of t
itmax <- nt1+nt2; # number of time intervals

### set initial values of the p-value and power for each simulation
p.zs<-rep(0,n.iter); # array for storing p values
p.sig2<-rep(0,n.iter); # array for storing power
p.sig5<-rep(0,n.iter);
cl level <-0;

### main loop
for (it in 1:n.iter)
{
  ### generate survival data following Weibull distn
  u1 <- sort(runif(nt1)); #sorted uniform random data
  t1 <- xi/(1/u1-1); #convert into survival time
  u2 <- sort(runif(nt2));
  t3 <- xi/(1/u2-1);
  t2 <- (xi^(1-rbeta))*(t3^rbeta);

  ### set censored data randomly,c1[i], c2[i] =1 if censored

```

```

c1 <- rep(0, nt 1); c2 <- rep(0, nt 2);
if ( censor | vl > 0 ) {
  tc1 <- ( r u n i f ( n t 1, m i n = t 0, m a x = t m ) ); #if need sorting?
  tc2 <- ( r u n i f ( n t 2, m i n = t 0, m a x = t m ) );
  for ( i i n 1 : n t 1 )
  {
    if ( t 1 [ i ] > t c 1 [ i ] ) c 1 [ i ] <- 1;
  };
  for ( i i n 1 : n t 2 )
  {
    if ( t 2 [ i ] > t c 2 [ i ] ) c 2 [ i ] <- 1;
  };
};

### survival time table
tb <- sort ( c ( t 1, t 2 ), i n d e x . r e t u r n = T R U E );
t <- t b $ x ; t i n d e x <- t b $ i x ;

### estimate of survival functions
phi 1 <- rep ( 0, i t m a x ); phi 2 <- rep ( 0, i t m a x );
S1 <- rep ( 1, i t m a x ); S2 <- rep ( 1, i t m a x );
t x i <- t / x i ;

for ( i i n 2 : i t m a x )
{
  if ( t i n d e x [ i ] <= n t 1 ) {
    i n d e x 1 <- t i n d e x [ i ];
    if ( c 1 [ i n d e x 1 ] == 0 ) { S 1 [ i ] <- 1. 0 / ( 1 + t x i [ i ] ) };
    if ( c 1 [ i n d e x 1 ] == 1 ) { S 1 [ i ] <- S 1 [ i - 1 ] };
    S 2 [ i ] <- S 2 [ i - 1 ];
  }
  if ( t i n d e x [ i ] > n t 1 ) {
    i n d e x 2 <- t i n d e x [ i ] - n t 1 ;
    if ( c 2 [ i n d e x 2 ] == 0 ) { S 2 [ i ] <- 1. 0 / ( 1 + ( t x i [ i ] ^ b e t a ) ) };
    if ( c 2 [ i n d e x 2 ] == 1 ) { S 2 [ i ] <- S 2 [ i - 1 ] };
    S 1 [ i ] <- S 1 [ i - 1 ];
  }
}

### odds, phi
for ( i i n 1 : i t m a x )
{
  if ( S 1 [ i ] <= 0. 9 9 9 ) phi 1 [ i ] <- S 1 [ i ] / ( 1 - S 1 [ i ] );
  if ( S 2 [ i ] <= 0. 9 9 9 ) phi 2 [ i ] <- S 2 [ i ] / ( 1 - S 2 [ i ] );
};
dat aset 1 1 <- dat a . f r a m e ( t , t i n d e x , S 1 , S 2 , phi 1 , phi 2 )

### F
F1 <- 1 - S1; F2 <- 1 - S2;

### K1 K2
k1 t m p <- ( 1 + ( t / x i ) );
k2 t m p <- ( 1 + ( ( t / x i ) ^ b e t a ) );
K2 <- 1. 0 / ( k 1 t m p * k 2 t m p );
K1 <- 1. 0 / ( r h o * k 2 t m p + ( 1 - r h o ) * k 1 t m p );

### set integration interval (overlap region)
r r <- rep ( 0, i t m a x );
for ( i i n 1 : i t m a x )
{
  if ( ( phi 1 [ i ] * phi 2 [ i ] ) != 0 ) r r [ i ] <- 1;
};

psi 1 1 <- sum ( K 1 * phi 1 * r r ); psi 1 2 <- sum ( K 1 * phi 2 * r r );
psi 2 1 <- sum ( K 2 * phi 1 * r r ); psi 2 2 <- sum ( K 2 * phi 2 * r r );

dat aset 1 <- dat a . f r a m e ( t , S 1 , S 2 , F 1 , F 2 , phi 1 , phi 2 , K 1 , K 2 , r r );

### gamma
GAMMA <- psi 1 1 * psi 2 2 - psi 1 2 * psi 2 1;

```

```

### v i j  c a l c u l a t i o n
### i n i t i a l  v a l u e s  o f  v i j
v11 <- 0; v12 <- 0; v21 <- 0; v22 <- 0;
for (kt in 1:itmax)
{
  for (ks in 1:itmax)
  {
    nu<- m i n(ks, kt) ;
    ss1t mp<- S1[ nu] ; ss2t mp<- S2[ nu] ;
    i f (ss1t mp<0. 000001) ss1t mp<- 0. 000001;
    i f (ss2t mp<0. 000001) ss2t mp<- 0. 000001;
    t emp1 <- 0; t emp2 <- 0;
    i f (( F1[ ks]>=0. 01) &( F1[ kt] >=0. 01)) {
      t emp1 <- ( 1. 0/ ss1t mp- 1)/( F1[ ks] * F1[ kt] )};
    i f (( F2[ ks]>=0. 01) &( F2[ kt] >=0. 01)) {
      t emp2 <- ( 1. 0/ ss2t mp- 1)/( F2[ ks] * F2[ kt] )};
    t mp<- phi 1[ ks] * phi 2[ kt] * ( t emp1+t emp2);
    v11 <- v11 + K1[ kt] * K1[ ks] * t mp;
    v12 <- v12 + K1[ kt] * K2[ ks] * t mp;
    v21 <- v21 + K2[ kt] * K1[ ks] * t mp;
    v22 <- v22 + K2[ kt] * K2[ ks] * t mp;
  }; # end of ks loop
}; # end of kt loop

### v a r i a n c e
var 2<- psi 22* psi 21* v11- psi 22* psi 11* v12- psi 12* psi 21* v21+psi 12* psi 11* v22;

### s t a t i s t i c s
zs<-0;
i f ( var 2>0) zs<- sqrt ( nt 1+nt 2) * GAMMA/ sqrt ( var 2);

### p- v a l u e
i f (zs >= 0) (p. zs[ it] <- 2*( 1- pnorm(zs)));
i f (zs < 0) (p. zs[ it] <- 2*pnorm(zs));
i f (p. zs[ it] < si gl vl 5) (p. si g5[ it] <- 1);
i f (p. zs[ it] < si gl vl 2) (p. si g2[ it] <- 1);
cl evel <- cl evel +(sum(c1)+sum(c2))/( nt 1+nt 2);
}; # end of m a i n l o o p

# p r i n t  e r r o r  i f  c r a s h
i f (it<n.iter){
  p r i n t (' i t < n. i t e r' ); i t;
  d a t a s e t 1; d a t a s e t 11;
  G A M M A; v a r 2; v 11; v 12; v 21; v 22;
  p s i 11; p s i 12; p s i 21; p s i 22;
};

### a v e r a g e  p- v a l u e
p a v e. z s<- s u m ( p. z s) / n. i t e r;

### p o w e r
p o w e r 2<- s u m ( p. s i g 2) / n. i t e r;
p o w e r 5<- s u m ( p. s i g 5) / n. i t e r;
p a v e. z s; p o w e r 2; p o w e r 5;

### c h e c k  c e n s o r i n g  l e v e l
c l e v e l / n. i t e r;

### f o l l o w i n g  i s  o n l y  f o r  s i n g l e  s t e p  r u n
# i f ( z s > 0) p <- 2 * ( 1 - p n o r m ( z s ) );
# i f ( z s < = 0) p <- 2 * p n o r m ( z s );
# z s;
# p; # p- v a l u e
# t i m e <- c ( 0, t ); S S 1 <- c ( 1, S 1 ); S S 2 <- c ( 1, S 2 );
# p l o t ( t i m e, S S 1, " S" )
# l i n e s ( t i m e, S S 2, " S", " r e d" )
# c e n s o r i n g  l e v e l
# s u m ( c 1) / n t 1;
# s u m ( c 2) / n t 2;

```

Program 2.3: Test the proportional odds assumption using linear regression method

```

### set parameter xi and beta
xi <- 1; # constant, mean
beta <- 2; # parameter
rbeta <- 1.0/beta; # rbeta=1/beta
censorlvl <- 0.0 # censoring level
if(censorlvl==0.1){
  t0<-2; tm<-20;
}
if(censorlvl==0.3){
  t0<-0.2; tm<-8;
}

### number of simulations, number of data point in each group
n.iter <- 50; # number of simulations
nt1 <- 50; nt2 <- 60; # number of data points in each group
siglvl2 <- 0.02; siglvl5 <- 0.05;

### set the groups of t
itmax <- nt1+nt2; # number of time intervals

p.zs<-rep(0,n.iter); # array for storing p values
p.sig2<-rep(0,n.iter); # array for storing power
p.sig5<-rep(0,n.iter);
clevel <- 0;
slope<-rep(0,n.iter);
slope.se<-rep(0,n.iter);
p.value<-rep(0,n.iter);

### main loop
for (it in 1:n.iter)
{
  ### generate survival data group one: t1
  u1 <- sort(runif(nt1)); #sorted uniform random data
  t1 <- xi/(1/u1-1); #convert into survival time
  u2 <- sort(runif(nt2));
  t3 <- xi/(1/u2-1);
  t2 <- (xi^(1-rbeta))*(t3^rbeta);

  c1 <- rep(0,nt1); c2 <- rep(0,nt2);
  if(censorlvl>0){
    tc1 <- (runif(nt1,min=t0,max=tm)); #if need sorting?
    tc2 <- (runif(nt2,min=t0,max=tm));
    for(i in 1:nt1)
    {
      if (t1[i] > tc1[i]) c1[i] <- 1;
    };
    for(i in 1:nt2)
    {
      if (t2[i] > tc2[i]) c2[i] <- 1;
    };
  };
  ### survival time table
  tb<-sort(c(t1,t2),index.return=TRUE);
  t<-tb$x; tindex<-tb$ix;

  phi1<-rep(0,itmax); phi2<-rep(0,itmax);
  S1<-rep(1,itmax); S2<-rep(1,itmax);
  txi <- t/xi;

  for (i in 2:itmax)
  {
    if(tindex[i]<=nt1){
      index1<-tindex[i];
      if(c1[index1]==0){S1[i] <- 1.0/(1+xtxi[i])};
      if(c1[index1]==1){S1[i] <- S1[i-1]};
      S2[i] <- S2[i-1];
    }
    if(tindex[i]>nt1) {
      index2<-tindex[i]-nt1;

```



```

        if (c2[index2]==0){ S2[i] <- 1.0/(1+(t xi [i]^bet a))};
        if (c2[index2]==1){ S2[i] <- S2[i-1]};
        S1[i] <- S1[i-1];
    }
}

### phi, and survival odds ratio (theta)
theta <- rep(0, itmax); tt <- rep(0, itmax);
jjj <- 0;
for (i in 1:itmax)
{
    if (S1[i] <= 0.999) phi 1[i] <- S1[i]/(1-S1[i]);
    if (S2[i] <= 0.999) phi 2[i] <- S2[i]/(1-S2[i]);
    tmp <- phi 1[i]*phi 2[i];
    if (tmp!=0) {
        jjj <- jjj+1;
        tt[jjj] <- t[i];
        theta[jjj] <- phi 1[i]/phi 2[i];
    }
};

cl evel <- cl evel +(sum(c1)+sum(c2))/(nt1+nt2);
t het a1 <- t het a[1:jjj]; t tt <- t t[1:jjj];
l og t het a <- l og(t het a1); l og t <- l og(t tt);
l m D9 <- l m(l og t het a ~ l og t);
t emp <- summar y(l m D9);
t emp1 <- anova(l m D9);
s l ope[i t] <- t emp$coef f i c i e n t s[2, 1];
s l ope. se[i t] <- t emp$coef f i c i e n t s[2, 2];
p. val ue[i t] <- t emp$coef f i c i e n t s[2, 4];
if (p. val ue[i t] < si gl vl 5){ p. si g5[i t] <- 1};
if (p. val ue[i t] < si gl vl 2){ p. si g2[i t] <- 1};
};
# censor level
cl evel / n. i t e r;
# average p- value
sum(p. zs) / n. i t e r;
# power for 0.05 significance
sum(p. si g5) / n. i t e r;
# power for 0.02 significance
sum(p. si g2) / n. i t e r;
# average slope and its deviation estimated from lm function
sum(s l ope) / n. i t e r;
sum(s l ope. se) / n. i t e r;

```

Program 2.4: FORTRAN programs for testing proportional odds assumption using extended method. It is about 177 times faster than the R program for the same simulation task

```

pr ogr am mai n
i m p l i c i t none
i n t e g e r n t t, n b, n n
p a r a m e t e r ( n t t = 4 0 0, n b = 5, n n = 2)
r e a l * 8 t 1( n t t), t 2( n t t), c 1( n t t), c 2( n t t), y( n t t), d( n t t),
& t( n t t), y 1( n t t), y 2( n t t), d 1( n t t), d 2( n t t), c( n t t),
& phi 1( n t t), phi 2( n t t), t c( n t t), s( n t t), f( n t t),
& s s 1( n t t), s s 2( n t t), s 1( n t t), s 2( n t t), f 1( n t t), f 2( n t t),
& k 1( n t t), k 2( n t t), b( n b), n 1( n n), n 2( n n), t 1 0( n b), t 3 0( n b),
& z s, p_ s i g 2, p_ s i g 5, b e t a, x i, r r, c t, t m a x, c e n s o r, c l v l,
& r b e t a, r a n m a r, v 1 1, v 1 2, v 2 1, v 2 2, p s i 1 1, p s i 1 2, p 2, p 5, t 0,
& p s i 2 1, p s i 2 2, v a r 2, g a m a, t m p, t e m p 1, t e m p 2, d b y y 1, d b y y 2
i n t e g e r n_ i t e r, n t 1, n t 2, i t m a x, i, j, k, i s e e d, j s e e d, k s, k t,
& k u, n u, i t e r, i d( n t t), i d n u l l( n t t), i b, j n
d a t a n 1/ 5 0, 1 0 0/, n 2/ 6 0, 1 2 0/, b/ 1, 2, 3, 4, 6/
d a t a t 3 0/ 5 1 0, 3 9 5, 3 9 5, 3 9 5, 3 9 5/, t 1 0/ 1 5 5 0, 8 2 0, 7 0 0, 6 8 0, 6 7 0/

o p e n( 1, f i l e = ' d a u x o i s. r e s' )
o p e n( 2, f i l e = ' d a u x o i s. d a t' )
w r i t e( *, *) ' n_ i t e r = '
r e a d( *, *) n_ i t e r

```

```

write(*,*)' censor level %
read(*,*) censor
xi=150
t0=5
i seed=18734
j seed=23869
c loop over (nt 1, nt 2)
do j n=1, 2
nt 1=n1(j n)
nt 2=n2(j n)
i t max=nt 1+nt 2
c loop over bet a
do i b=1, 5
bet a=b(i b)
r bet a=1./bet a
i f (censor . eq. 30) t max=t 30(i b)
i f (censor . eq. 10) t max=t 10(i b)

call r mar in(i seed, j seed)

p_si g2=0. 0
p_si g5=0. 0
ct=0. 0
c loop over si mul at i on
do i ter=1, n_i ter
do i =1, nt 1
t 1(i)=xi / ( 1. / r an mar () - 1. )
enddo
do i =1, nt 2
t 2(i)=xi / (( 1. / r an mar () - 1. ) ** r bet a)
enddo
do i =1, nt 1
t (i)=t 1(i)
i d(i)=1
enddo
do i =nt 1+1, i t max
t (i)=t 2(i - nt 1)
i d(i)=2
enddo
call shel l (i t max, t, i d)
do i =1, i t max
i f (censor . gt . 0) t hen
t c(i)=t 0+(t max-t 0)*r an mar ()
el se
t c(i)=0. 0
endi f
enddo
i f (censor . gt . 0) call shel l (i t max, t c, i dnul l)
do i =1, i t max
i f (i d(i) . eq. 1) t hen
c2(i)=0. 0
d2(i)=0. 0
i f (censor . gt . 0. and. t (i) . gt . t c(i)) t hen
c1(i)=1. 0
d1(i)=0. 0
el se
c1(i)=0. 0
d1(i)=1. 0
endi f
el se i f (i d(i) . eq. 2) t hen
c1(i)=0. 0
d1(i)=0. 0
i f (censor . gt . 0. and. t (i) . gt . t c(i)) t hen
c2(i)=1. 0
d2(i)=0. 0
el se
c2(i)=0. 0
d2(i)=1. 0
endi f
el se
write(*,*)' i d(i) NE 1 or 2'
st op

```

```

        endi f
    enddo
    y1(1)=nt 1
    y2(1)=nt 2
    do i=2, i t max
        y1(i)=y1(i-1)-d1(i-1)-c1(i-1)
        y2(i)=y2(i-1)-d2(i-1)-c2(i-1)
    enddo
    do i=1, i t max
        y(i)=y1(i)+y2(i)
        d(i)=d1(i)+d2(i)
        c(i)=c1(i)+c2(i)
        ct=ct+c(i)
        i f (y1(i). ne. 0) t hen
            ss1(i)=1.0-d1(i)/y1(i)
        el se
            ss1(i)=0.0
        endi f
        i f (y2(i). ne. 0) t hen
            ss2(i)=1.0-d2(i)/y2(i)
        el se
            ss2(i)=0.0
        endi f
    enddo
    s1(1)=ss1(1)
    s2(1)=ss2(1)
    do i=2, i t max
        s1(i)=s1(i-1)*ss1(i)
        s2(i)=s2(i-1)*ss2(i)
    enddo
    t emp1=1.0/(nt 1*nt 2)
    t emp2=(nt 1+nt 2)*t emp1
    psi 11=0.0
    psi 12=0.0
    psi 21=0.0
    psi 22=0.0
    i f (i t er. eq. n_ i t er) t hen
        w r i t e(2,*)'EXAMPLE DATA TABLE:'
        w r i t e(2,110)n_ i t er, bet a, nt 1, nt 2
        w r i t e(2,*)' i t i d y d c y1 d1 c1 y2 d2 c2
& s1 s2 rr phi 1 phi 2 k1 k2'
    endi f
    do i=1, i t max
        f 1(i)=1.0-s1(i)
        f 2(i)=1.0-s2(i)
        phi 1(i)=0
        phi 2(i)=0
        i f (f 1(i). ne. 0) phi 1(i)=s1(i)/f 1(i)
        i f (f 2(i). ne. 0) phi 2(i)=s2(i)/f 2(i)
        t mp=y1(i)*y2(i)
        k1(i)=t emp2*t mp/y(i)
        k2(i)=t emp1*t mp
        i f (phi 1(i)*phi 2(i). eq. 0) t hen
            r r=0.0
        el se
            r r=1.0
        endi f
        psi 11=psi 11+k1(i)*phi 1(i)*r r
        psi 12=psi 12+k1(i)*phi 2(i)*r r
        psi 21=psi 21+k2(i)*phi 1(i)*r r
        psi 22=psi 22+k2(i)*phi 2(i)*r r
        i f (i t er. eq. n_ i t er) t hen
            w r i t e(2,100)i, t (i), i d(i), y(i), d(i), c(i),
& y1(i), d1(i), c1(i), y2(i), d2(i), c2(i), s1(i), s2(i), r r, phi 1(i),
& phi 2(i), k1(i), k2(i)
        endi f
    enddo
    gama=psi 11*psi 22-psi 12*psi 21
    v11=0.0
    v12=0.0
    v21=0.0
    v22=0.0

```

```

do kt =1, i t max
  do ks=1, i t max
    i f (k t . l t . k s) t h e n
      n u =k t
    e l s e
      n u =k s
    e n d i f
    d b y y 1 =0 . 0
    d b y y 2 =0 . 0
    t e m p 1 =0 . 0
    t e m p 2 =0 . 0
    d o k u =1, n u
      i f (y 1 (k u) . n e . 0) d b y y 1 =d b y y 1 +d 1 (k u) / (y 1 (k u) * y 1 (k u))
      i f (y 2 (k u) . n e . 0) d b y y 2 =d b y y 2 +d 2 (k u) / (y 2 (k u) * y 2 (k u))
    e n d d o
    i f ((f 1 (k s) . n e . 0) . a n d . (f 1 (k t) . n e . 0))
&     t e m p 1 =d b y y 1 * n t 1 / (f 1 (k s) * f 1 (k t))
    i f ((f 2 (k s) . n e . 0) . a n d . (f 2 (k t) . n e . 0))
&     t e m p 2 =d b y y 2 * n t 2 / (f 2 (k s) * f 2 (k t))
    t m p =p h i 1 (k s) * p h i 2 (k t) * (t e m p 1 +t e m p 2)
    v 1 1 =v 1 1 +k 1 (k t) * k 1 (k s) * t m p
    v 1 2 =v 1 2 +k 1 (k t) * k 2 (k s) * t m p
    v 2 1 =v 2 1 +k 2 (k t) * k 1 (k s) * t m p
    v 2 2 =v 2 2 +k 2 (k t) * k 2 (k s) * t m p
  e n d d o
  e n d d o
  v a r 2 =p s i 2 2 * p s i 2 1 * v 1 1 - p s i 2 2 * p s i 1 1 * v 1 2
&     - p s i 1 2 * p s i 2 1 * v 2 1 + p s i 1 2 * p s i 1 1 * v 2 2
  i f (v a r 2 . g t . 0) t h e n
    z s =s q r t (f l o a t (n t 1 +n t 2) ) * g a m a / s q r t (v a r 2)
  e l s e
    z s =0
  e n d i f
  i f (z s . g e . 1 . 9 6 . o r . z s . l e . - 1 . 9 6) p _ s i g 5 =p _ s i g 5 +1 . 0
  i f (z s . g e . 2 . 3 2 . o r . z s . l e . - 2 . 3 2) p _ s i g 2 =p _ s i g 2 +1 . 0
  e n d d o
100  f o r m a t (i 3, f 1 1 . 3, i 2, f 5 . 0, 2(f 3 . 0), f 4 . 0, 2(f 3 . 0), f 4 . 0,
& 2(f 3 . 0), 2(f 6 . 3), f 3 . 0, 2(f 7 . 2), 2(f 6 . 2))

  c l v l =c t / (f l o a t (n _ i t e r) * (n t 1 +n t 2))
  p 2 =p _ s i g 2 / f l o a t (n _ i t e r)
  p 5 =p _ s i g 5 / f l o a t (n _ i t e r)
  w r i t e (*, 1 1 0) n _ i t e r, b e t a, n t 1, n t 2, c l v l, p 2, p 5
  w r i t e(1, 1 1 0) n _ i t e r, b e t a, n t 1, n t 2, c l v l, p 2, p 5
110  f o r m a t (' s i m u l a t i o n =', i 5, ' b e t a =', f 3 . 0, ' (n t 1, n t 2) =', 2(i 4),
& ' c e n s o r =', f 6 . 4, ' p o w e r 0 . 0 2 =', f 6 . 4, ' p o w e r 0 . 0 5 =', f 6 . 4)
  e n d d o
  e n d d o
  e n d

```

c shell method to sort number

```

sub r o u t i n e s h e l l (n, a r r, i d)
p a r a m e t e r (a l n 2 i =1 . / 0 . 6 9 3 1 4 7 1 8, t i n y =1 . e - 5)
d i m e n s i o n i d (n)
r e a l * 8 a r r (n)
l o g n b 2 =i n t (a l o g (f l o a t (n) ) * a l n 2 i +t i n y)
m =n
d o n n =1, l o g n b 2
  m =m / 2
  k =n - m
  d o j =1, k
    i =j
3    l =i +m
    i f (a r r (l) . l t . a r r (i) ) t h e n
      t =a r r (i)
      a r r (i) =a r r (l)
      a r r (l) =t
      i d t =i d (i)
      i d (i) =i d (l)
      i d (l) =i d t
      i =i - m

```

```

        if(i.ge.1)go to 3
      endif
    enddo
  enddo
return
end

```

c random number generator subroutine

```

subroutine rmarin(ij,kl)
real*8 u(97), c, cd, cm s, t
integer i97, j97, ij, kl
integer i, j, k, l, ii, jj, m
logical test
common /raset1/ u
common /raset2/ c, cd, cm
common /raset3/ i97, j97
common /raset4/ test

```

```

if( ij .lt. 0 .or. ij .gt. 31328 .or.
* kl .lt. 0 .or. kl .gt. 30081 ) then
print '(a)', ' the first random number seed must have a value
* between 0 and 31328'
print '(a)', ' the second seed must have a value between 0 and
* 30081'
stop
endif

```

```

i = mod(ij/177, 177) + 2
j = mod(ij, 177) + 2
k = mod(kl/169, 178) + 1
l = mod(kl, 169)
do 2 ii = 1, 97
  s = 0.0
  t = 0.5
  do 3 jj = 1, 24
    m = mod(mod(i*j, 179)*k, 179)
    i = j
    j = k
    k = m
    l = mod(53*l+1, 169)
    if (mod(l*m 64) .ge. 32) then
      s = s + t
    endif
  enddo
  t = 0.5 * t

```

```

3 continue
  u(ii) = s
2 continue
c = 362436.0 / 16777216.0
cd = 7654321.0 / 16777216.0
cm = 16777213.0 / 16777216.0
i97 = 97
j97 = 33

test = .true.
return
end

```

c random number function

```

function ranmar()
real*8 u(97), c, cd, cm
real*8 ranmar, uni
integer i97, j97
logical test
common /raset1/ u
common /raset2/ c, cd, cm
common /raset3/ i97, j97
common /raset4/ test

```

```

if( .not. test ) then
print '(a)', ' call the init routine (rmarin) before calling

```

```

* ranmar'
stop
endif

uni = u(i97) - u(j97)
if( uni .lt. 0.0 ) uni = uni + 1.0
u(i97) = uni
i97 = i97 - 1
if(i97 .eq. 0) i97 = 97
j97 = j97 - 1
if(j97 .eq. 0) j97 = 97
c = c - cd
if( c .lt. 0.0 ) c = c + cm
uni = uni - c
if( uni .lt. 0.0 ) uni = uni + 1.0
ranmar = uni

return
end

```

Program 2.5: R programs for estimating the parameters from real data using MLE method, the log-logistic distribution is assumed for the real data

```

# Estimate parameter (xi, beta) using MLE
# Assume survival data follow log-logistic distribution
# with nonproportional hazards

# read in real data
dd<-read.table("d:xuqing/phdthesi/s/qingb14.txt", header=TRUE, sep=",")
# dd$group; #group 1 placebo, group 2 tamoxifen group
itmax<-2817;
time.temp<-sort(dd$time, index.return=TRUE)
t<-time.temp$x; tindex<-time.temp$ix;
g<-rep(0,itmax); c<-g; e<-c;
for (i in 1:itmax)
{
  ii<-tindex[i];
  c[i]<-dd$ndi.event[s[ii]];
  g[i]<-dd$group[ii];
  e[i]<-dd$event[s[ii]];
}
# number of events
r<-sum(c);
# MLE functions Eq. (4.35) and (4.36)
fff<-function(p){
  tbeta<-t^(p[1]*(g-1)); #p[1]=beta, p[2]=xi
  aa<-(c+1)*tbeta;
  bb<-p[2]^p[1]+tbeta;
  cc<-aa/bb;
  dd<-cc*log(t);
  ee<-c*log(t);
  tst<-(r-sum(cc))^2+(r/p[1]+sum(ee)-sum(dd))^2;
  tst;
}
# solve the nonlinear equations to get beta and xi
res<-nlm(fff,c(1,2)) #initial value of beta=1, xi=2
beta.hat<-res$estimate[1]; #MLE beta.hat
xi.hat<-res$estimate[2]; #MLE xi.hat
rmin<-res$minimum #final fff value
beta.hat; xi.hat;

```

Program 2.6: R programs for testing the proportional odds assumption using Dauxois-Kirmani method for real data, the log-logistic distribution is assumed for the real data

```

# read in real data
dd<-read.table("d:/xuqing/phdthesi/s/qingb14.txt", header=TRUE, sep=",")

```

```

# dd$group; #group 1 placebo, group 2 tamoxifen group
itmax<-2817;
time.temp<-sort(dd$time,index.return=TRUE)
t<-time.temp$x; tindex<-time.temp$x;
g<-rep(0,itmax); c<-g; e<-c;
for (i in 1:itmax)
{
  ii<-tindex[i];
  c[i]<-dd$indi.events[ii];
  g[i]<-dd$group[ii];
  e[i]<-dd$events[ii];
}
newdata<-data.frame(g,e,t,c,tindex);
#newdata;
y1<-0; y2<-0; c1<-0; c2<-0;
for (i in 1:itmax)
{
  if(g[i]==1)
  {
    y1<-y1+1;
    c1<-c1+c[i];
  }
  if(g[i]==2)
  {
    y2<-y2+1;
    c2<-c2+c[i];
  }
}
# number of objects in group 1
nt1<-y1;
# number of objects in group 2
nt2<-y2;
# observed events in group 1
c1;
# observed events in group 2
c2;
sum(c);

S1<-rep(1,itmax); S2<-S1; yy1<-rep(y1,itmax); yy2<-rep(y2,itmax);
d1<-rep(0,itmax); d2<-d1;
if(g[1]==1){
  y1<-y1-1;
  if(c[1]==1){d1[1]<-1}
}
if(g[1]==2){
  y2<-y2-1;
  if(c[1]==1){d2[1]<-1}
}
for (i in 2:itmax)
{
  if(g[i]==1)
  {
    yy1[i]<-y1;
    yy2[i]<-yy2[i-1];
    S2[i]<-S2[i-1];
    if(c[i]==1){
      S1[i]<-S1[i-1]*(1-1/y1);
      d1[i]<-1;
    }
    if(c[i]==0){S1[i]<-S1[i-1];}
    y1<-y1-1;
  }
  if(g[i]==2)
  {
    yy2[i]<-y2;
    yy1[i]<-yy1[i-1];
    S1[i]<-S1[i-1];
    if(c[i]==1){
      S2[i]<-S2[i-1]*(1-1/y2);
      d2[i]<-1;
    }
    if(c[i]==0){S2[i]<-S2[i-1];}
  }
}

```

```

    }
    y2<-y2- 1;
}

dat a1<- dat a. f r a m e ( t , yy1, yy2, d1, d2, S1, S2, c );
#dat a1;

yy<-yy1+yy2;
phi 1<-r e p ( 0, i t m a x ); phi 2<-r e p ( 0, i t m a x );

### phi , the survi val odds
for ( i i n 1: i t m a x )
{
  i f ( S1 [ i ] ! = 1 ) phi 1 [ i ] <- S1 [ i ] / ( 1 - S1 [ i ] );
  i f ( S2 [ i ] ! = 1 ) phi 2 [ i ] <- S2 [ i ] / ( 1 - S2 [ i ] );
};

### F
F1 <- 1 - S1; F2 <- 1 - S2;

### K1 K2
K1<- ( ( n t 1 + n t 2 ) / ( n t 1 * n t 2 ) ) * yy1 * yy2 / yy;
K2<- yy1 * yy2 / n t 1 / n t 2;

### set i n t e g r a t i o n i n t e r v a l ( o v e r l a p r e g i o n )
rr<-r e p ( 0, i t m a x );
for ( i i n 1: i t m a x )
{
  i f ( ( phi 1 [ i ] * phi 2 [ i ] ) ! = 0 ) rr [ i ] <- 1;
};

psi 11<- s u m ( K1 * phi 1 * rr ); psi 12<- s u m ( K1 * phi 2 * rr );
psi 21<- s u m ( K2 * phi 1 * rr ); psi 22<- s u m ( K2 * phi 2 * rr );
dat a s e t <- dat a. f r a m e ( t , yy, yy1, yy2, S1, S2, rr );

### gamma
GAMMA<- psi 11 * psi 22 - psi 12 * psi 21;

### v i j c a l c u l a t i o n
### i n i t i a l v a l u e s o f v i j
v11 <- 0; v12 <- 0; v21 <- 0; v22 <- 0;
for ( kt i n 1: i t m a x )
{
  for ( ks i n 1: i t m a x )
  {
    nu<-m i n ( ks, kt );
    dbyy1 <- 0; dbyy2 <- 0;
    t e m p 1 <- 0; t e m p 2 <- 0;
    for ( ku i n 1: nu )
    {
      i f ( yy1 [ ku ] ! = 0 ) dbyy1 <- dbyy1 + d1 [ ku ] / ( yy1 [ ku ] * yy1 [ ku ] );
      i f ( yy2 [ ku ] ! = 0 ) dbyy2 <- dbyy2 + d2 [ ku ] / ( yy2 [ ku ] * yy2 [ ku ] );
    };
    i f ( ( F1 [ ks ] ! = 0 ) & ( F1 [ kt ] ! = 0 ) ) t e m p 1 <- n t 1 * dbyy1 / ( F1 [ ks ] * F1 [ kt ] );
    i f ( ( F2 [ ks ] ! = 0 ) & ( F2 [ kt ] ! = 0 ) ) t e m p 2 <- n t 2 * dbyy2 / ( F2 [ ks ] * F2 [ kt ] );
    t m p <- phi 1 [ ks ] * phi 2 [ kt ] * ( t e m p 1 + t e m p 2 );
    v11 <- v11 + K1 [ kt ] * K1 [ ks ] * t m p;
    v12 <- v12 + K1 [ kt ] * K2 [ ks ] * t m p;
    v21 <- v21 + K2 [ kt ] * K1 [ ks ] * t m p;
    v22 <- v22 + K2 [ kt ] * K2 [ ks ] * t m p;
  }; # end of ks loop
}; # end of kt loop

### v a r i a n c e
var 2<- psi 22 * psi 21 * v11 - psi 22 * psi 11 * v12 - psi 12 * psi 21 * v21 + psi 12 * psi 11 * v22;

### s t a t i s t i c s
zs<- 0;
i f ( var 2 ! = 0 ) zs<- s q r t ( n t 1 + n t 2 ) * GAMMA / s q r t ( var 2 );

### p - v a l u e
i f ( zs > 0 ) ( p. z s <- 2 * ( 1 - p n o r m ( zs ) ) );

```



```
if(zs < 0) (p.zs <- 2*pnorm(zs));
p.zs;
```

Program 2.7: R programs for testing the proportional odds assumption using the extended method for real data, the log-logistic distribution is assumed for the real data

```
# read in real data
dd<-read.table("d:/xuqing/phdthesis/qingb14.txt", header=TRUE, sep=",")

# dd$group; #group 1 placebo, group 2 tamoxifen group
itmax<-2817;
time.temp<-sort(dd$time, index.return=TRUE)
t<-time.temp$x; tindex<-time.temp$ix;
g<-rep(0,itmax); c<-g; e<-c;
for (i in 1:itmax)
{
  ii<-tindex[i];
  c[i]<-dd$ndi.event$[ii];
  g[i]<-dd$group[ii];
  e[i]<-dd$event$[ii];
}
newdata<-data.frame(g, e, t, c, tindex);
#newdata:
y1<-0; y2<-0; c1<-0; c2<-0;
for (i in 1:itmax)
{
  if(g[i]==1)
  {
    y1<-y1+1;
    c1<-c1+c[i];
  }
  if(g[i]==2)
  {
    y2<-y2+1;
    c2<-c2+c[i];
  }
}
# number of objects in group 1
y1;
# number of objects in group 2
y2;
# observed events in group 1
c1;
# observed events in group 2
c2;
sum(c);
xi<-1.2927;
beta<-6.1214;
rho<-y1/(y1+y2);

### estimate of survival functions
phi1<-rep(0,itmax); phi2<-rep(0,itmax);
S1<-rep(1,itmax); S2<-rep(1,itmax);
txi<-t/xi;

for (i in 2:itmax)
{
  if(g[i]==1){
    if(c[i]==1){S1[i]<-1.0/(1+txi[i])};
    if(c[i]==0){S1[i]<-S1[i-1]};
    S2[i]<-S2[i-1];
  }
  if(g[i]==2){
    if(c[i]==1){S2[i]<-1.0/(1+(txi[i]^beta))};
    if(c[i]==0){S2[i]<-S2[i-1]};
    S1[i]<-S1[i-1];
  }
}
```

```

}

### odds, phi
for (i in 1:itmax)
{
  if (S1[i]!=1) phi 1[i] <- S1[i]/(1-S1[i]);
  if (S2[i]!=1) phi 2[i] <- S2[i]/(1-S2[i]);
};
dat aset 11<- data.frame(t, tindex, S1, S2, phi 1, phi 2)

### F
F1 <- 1-S1; F2 <- 1-S2;

### K1 K2
k1tmp<-(1+(t/xi));
k2tmp<-(1+(t/xi)^beta);
K2<-1.0/(k1tmp*k2tmp);
K1<-1.0/(rho*k2tmp+(1-rho)*k1tmp);

### set integration interval (overlap region)
rr<-rep(0,itmax);
for (i in 1:itmax)
{
  if((phi 1[i]*phi 2[i])!=0)rr[i]<-1;
};

psi 11<-sum(K1*phi 1*rr); psi 12<-sum(K1*phi 2*rr);
psi 21<-sum(K2*phi 1*rr); psi 22<-sum(K2*phi 2*rr);

dat aset 1<-data.frame(t, S1, S2, F1, F2, phi 1, phi 2, K1, K2, rr);

###gamma
GAMMA<-psi 11*psi 22-psi 12*psi 21;

### vij calculation
### initial values of vij
v11 <- 0; v12 <- 0; v21 <- 0; v22 <- 0;
for (kt in 1:itmax)
{
  for (ks in 1:itmax)
  {
    nu<-min(ks, kt);
    ss1tmp<-S1[nu]; ss2tmp<-S2[nu];
    #if (ss1tmp<0.000001) ss1tmp<-0.000001;
    #if (ss2tmp<0.000001) ss2tmp<-0.000001;
    temp1 <- 0; temp2 <- 0;
    if ((F1[ks]!=0)&(F1[kt]!=0)){
      temp1 <- (1.0/ss1tmp-1)/(F1[ks]*F1[kt]);
    }
    if ((F2[ks]!=0)&(F2[kt]!=0)){
      temp2 <- (1.0/ss2tmp-1)/(F2[ks]*F2[kt]);
    }
    tmp<-phi 1[ks]*phi 2[kt]*(temp1+temp2);
    v11 <- v11 + K1[kt]*K1[ks]*tmp;
    v12 <- v12 + K1[kt]*K2[ks]*tmp;
    v21 <- v21 + K2[kt]*K1[ks]*tmp;
    v22 <- v22 + K2[kt]*K2[ks]*tmp;
  }; # end of ks loop
}; # end of kt loop

### variance
var 2<-psi 22*psi 21*v11-psi 22*psi 11*v12-psi 12*psi 21*v21+psi 12*psi 11*v22;

### statistics
zs<-0;
if (var 2>0) zs<-sqrt(itmax)*GAMMA/sqrt(var 2);

### p-value
if (zs >= 0) (p.zs <- 2*(1-pnorm(zs)));
if (zs < 0) (p.zs <- 2*pnorm(zs));

p.zs;

```

BIBLIOGRAPHY

- Aalen, O. O. (1978). Nonparametric inference for a family of counting processes. *Annals of Statistics*, **6**, 534-545.
- Aalen, O. O. (1998). Frailty models. In *Statistical Analysis of Medical Data: New Developments*. (eds. B. S. Everitt & G. Dunn). Arnold: London.
- Altshuler, B. (1970). Theory for the measurement of competing risks in animal experiments. *Mathematical Biosciences*, **6**, 67-77.
- Andersen, P. K. (1982). Testing goodness of fit of Cox's regression and life model. *Biometrics*, **38**, 67-77. Correction: **40** (1984): 1217.
- Bennett, S. (1983a). Log-logistic regression models for survival data. *Applied Statistics*, **32**, 165-171.
- Bennett, S. (1983b). Analysis of survival data by the proportional odds model. *Statistics in Medicine*, **2**, 273-277.
- Burr, W. (1942) Cumulative frequency functions. *Annals of Mathematical Statistics*, **13**, 215-232.
- Cheng, S. C., Wei, L. J. & Ying Z. (1995) Analysis of transformation models with censored data. *Biometrika*, **82**, 835-845.
- Collett, D. (2003). *Modelling Survival Data in Medical Research*. Second Edition. Chapman & Hall/CRC: Boca Raton.
- Colosimo, E. A., Chalita, L. V. A., & Demetrio, C. G. B. (2000) Tests of proportional hazards and proportional odds models for grouped survival data. *Biometrics*, **56**, 1233-1240.
- Cox, D. R. (1972). Regression models and life tables. *Journal of the Royal Statistical Society, B*, **34**, 187-220.
- Cox, D. R. (1975). Partial likelihood. *Biometrika*, **62**, 269-276.
- Dauxios, J. Y. & Kirmani S. N. U. A. (2003). Testing the proportional odds model under random censoring. *Biometrika*, **90**, 913-922.
- DiRienzo, A. G. (2003). Nonparametric comparison of two survival-time distributions in the presence of dependent censoring. *Biometrics*, **59**, 497-504.

- Economou, P. & Caroni, C. (2005). Graphical tests for the assumption of Gamma and Inverse Gaussian frailty distributions. *Lifetime Data Analysis*, **11**, 565-582.
- Eng, K. H. & Kosorok, M. R. (2005). A sample size formula for the supremum log-rank statistic. *Biometrics*, **61**, 86-91.
- Fisher, B.; Costantino, J.; Redmond, C. et al. (1989). A randomized clinical trial evaluating tamoxifen in the treatment of patients with node-negative breast cancer who have estrogen-receptor-positive tumors. *New England J. Medicine*, **320**, 479-484.
- Fisher, B.; Dignam, J.; Bryant, J. et al. (1996). Five versus more than five years of tamoxifen therapy for breast cancer patients with negative lymph nodes and estrogen receptor-positive tumors. *J. National Cancer Inst.*, **88**, 1529-1542.
- Fleming, T. R., Harrington, D. P., & O'Sullivan, M. (1987) Supremum versions of the log-rank and generalized Wilcoxon statistics. *J. Am. Stat. Assoc.*, **82**, 312-320.
- Fleming, T. R. & Harrington, D. P. (1991). *Counting Processes and Survival Analysis*. Wiley: New York.
- Gehan, E. A. (1965) A generalized Wilcoxon test for comparing arbitrarily singly censored samples. *Biometrika*, **52**, 203-223.
- Gill, R. & Schumacher, M. (1987). A simple test of the proportional hazards assumption. *Biometrika*, **74**, 289-300.
- Harrington, D. P. & Fleming T. R. (1982). A class of rank test procedures for censored survival data. *Biometrika*, **69**, 553-566.
- Hougaard, P. (1984). Life table methods for heterogeneous populations: Distributions describing the heterogeneity, *Biometrika*, **71**, 75-83
- Hougaard, P. (1986). Survival models for heterogeneous populations derived from stable distributions. *Biometrika*, **73**, 387-396.
- Hougaard, P. (1991). Modeling heterogeneity in survival data. *J. Appl. Prob.* **28**, 695-701.
- Hougaard, P. (1995). Frailty models for survival data. *Lifetime Data Analysis*, **1**, 255-273.
- Hougaard, P. (2000). *Analysis of Multivariate Survival Data*. Springer, New York.
- Jeong, J. H., Jung, S. H., Wieand, S. (2003). A parametric model for long-term follow-up data from phase III breast cancer clinical trial. *Stat. Med.* **22**, 339-352.
- Kalbfleisch, J. D. & Prentice, R. L. (1980). *The statistical analysis of failure time data*. New York: John Wiley and Sons.
- Kaplan, E. L. & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, **53**, 457-481.

- Klein, J. P. & Moeschberger, M. L. (2003) *Survival Analysis*. Springer.
- Kosorok, M. R. & Lin, C. Y. (1999) *Journal of the American Statistical Association*, **94**, 320-332.
- Lam, K. F., Lee, Y. W. & Leung T. L. (2002) Modeling multivariate survival data by a semiparametric random effects proportional odds model. *Biometrics*, **58**, 316-323.
- Manatunga, A. & Oakes, D. (1999). Parametric analysis for matched pair survival data. *Lifetime Data Analysis*, **5**, 371-387.
- Mantel, N. (1966). Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemotherapy Reports*, **50**, 163-170.
- Mantel, N. & Haenszel, W. (1959). Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute*, **22**, 719-748.
- McCullagh, P. (1980) Regression models for ordinal data with discussion, *J. Royal Stat. Soc. Series B*, **42**, 109-142.
- Murphy, S. A.; Rossini, A. J. & van der Vaart A. W. (1997). Maximum likelihood estimation in the proportional odds model. *Journal of the American Statistical Association*, **92**, 968-976.
- Nelson, W. (1972). Theory and applications of hazard plotting for censored failure data. *Technometrics*, **14**, 945-965.
- Oakes, D. & Jeong, J. H. (1998). Frailty model and rank tests. *Lifetime Data Analysis*, **4**, 209-228.
- Oakes, D. (1977). The asymptotic information in survival data. *Biometrika*, **64**, 441-448.
- Pettitt, A. N. (1984). Proportional odds models for survival data and estimates using ranks. *Applied Statistics*, **2**, 169-175.
- Peto, R. (1972). Contribution to the discussion of a paper by D. R. Cox. *Journal of the Royal Statistical Society, B*, **34**, 205-207.
- Peto, R. & Peto, J. (1972). Asymptotically efficient rank invariant procedures. *Journal of the Royal Statistical Society, A*, **35**, 185-207.
- Prentice, L. R. & Marek, P. A. (1979) A qualitative discrepancy between censored data rank tests. *Biometrics*, **35**, 861-867.
- Schoenfeld, D. A. (1983) Sample-size formula for the proportional-hazards regression model. *Biometrics*, **39**, 499-503.

Sengupta, D., Bhattacharjee, A., & Rajeev, B. (1998) Testing for the proportionality of hazards in two samples against the increasing cumulative hazard ratio alternative. *Scand. J. Statist.* **25**, 637-647.

Shorank, G. R. & Wellner, J. A. (1986). *Empirical Processes with applications to statistics*. New York: Wiley.

Therneau, T. M. & Grambsch P. M. (2000). *Modelling Survival Data: Extending the Cox Model*. Springer: New York.

Vaupel, J. A.; Manton, K. G.; and Stallard, E. (1979) The impact of heterogeneity in individual frailty on the dynamics of mortality. *Demography*, **16**, 439-454.