1 Introduction

1.1 What is the YPmodel package?

The YPmodel package is an add-on package for the R [Team et al.(2008)] statistical computing system. It provides functions for the analysis of the comparison of failure times between a treated and control group under independent censorship, based on several research works by Song et al [Yang and Prentice(2005),Yang and Prentice(2009),Yang and Prentice(2011),Yang and Zhao(2012)]. All comments, criticisms and queries on the package or associated documentation are gratefully received. Citation information is available at http://cran.r-project.org/web/packages/YPmodel/citation.html.

1.2 Obtaining the package/guide

The package can be downloaded from CRAN (The Comprehensive R Archive Network) at http: //cran.r-project.org/. This guide (in pdf) will be in the directory *YPmodel/doc/* underneath wherever the package is installed. You can get it by invoking

```
browseVignettes(package = "YPmodel")
```

Listing 1: YPmodel help documentation.

To have a quick overview of what the package does, you might want to have a look at its own web page http://cran.r-project.org/web/packages/YPmodel/.

1.3 Contents

To help users to use properly the YPmodel packages, this report introduces all main functions in details. Some practical examples are inserted within the text to show how it works in practice. Section 2 introduces the data format on which the package is based. Section 3 describes how to estimate the model parameters. Section 4 presents functions to estimate confidence intervals and bands for the hazard ratio function. In section 5 and section 6, we show how to perform several hypothesis tests. Section 7 provides an all-in-one method corresponding to functions through this package. Section 8 concludes this manual. Note that algorithmic parameters are summarized in the Annex part.

1.4 Legalese

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details. A copy of the GNU General Public License can be obtained from http://www.gnu.org/copyleft/gpl.html.

2 Preparing Data

Consider the comparison of failure times between a treated and control group under independent censorship. Let $T_1, \ldots, T_n$ be the pooled lifetimes of the 2 groups, with $T_1, \ldots, T_{n_1}$, $n_1 < n$, constituting the control group having the survivor function S_C. Let $C_1, \ldots, C_n$ be the censoring variables, and $Z_i = I(i > n_1)$, $i = 1, \ldots, n$, where $I(\cdot)$ is the indicator function. The available data consist of the independent triplets (X_i, δ_i, Z_i), $i = 1, \ldots, n$, where $X_i = \min(T_i, C_i)$ and $\delta_i = I(T_i \leq C_i)$.

2.1 Loading gastric example

The Gastrointestinal Tumor Study Group (1982) [Gastrointestinal Tumor Study Group: Schein(1982)] compared chemotherapy with combined chemotherapy and radiation therapy, in the treatment of locally unresectable gastric cancer. Each treatment arm had 45 patients, with two observations of the chemotherapy group and six of the combination group censored. Kaplan-Meier plots of the two estimated survival curves cross at around 1000 days.

The study results are included in this package, named as 'gastric'. Before we call this sample data, we first must load the YPmodel (make sure the YPmodel has been installed in R). This can be done as follows:

```
library(YPmodel)
```

Listing 2: Loading YPmodel package.

Then we can load the sample data 'gastric' as follows:

```
data(gastric)
```

Listing 3: Loading sample data.

```
> gastric
             V1 V2 V3
1  0.002739726  1  0
2  0.046575342  1  1
3  0.115068493  1  1
4  0.120547945  1  1
5  0.131506849  1  1
6  0.164383562  1  1
7  0.172602740  1  0
8  0.197260274  1  1
9  0.202739726  1  1
10 0.260273973  1  1
...
```

Listing 4: Results of scripts.

where variables of each column present lifetime vector, censor indicator vector, and group indicator vector, respectively.

2.2 Inputting Data

In most cases, we want to use this package to run with our own data sets. In the following example, we demonstrate how to input data in R that can be directly executed by our package. Assume we have a simple test on one medicine effect, and only six persons were included. They were divided into treated and control group, where variable Z is the group indicators. This study lasted for 3 years, and researchers recorded their observed lifetime (unite: year, variable X) and wether or not they quitted or lived over the study's period (variable δ). Table 1 gives the data sample collected:

Before using algorithms in YPmodel, we first need to input this table into an R-based matrix. Let us assume that we want to name the matrix as 'data'. This can be done as follows:

```
data <- matrix(c(0.0027,0.8412,1.3741,1.8322,2.471,
             3,1,0,0,1,1,1,0,0,1,0,1,1),nrow=6,ncol=3)
```

Listing 5: Inputting data.

Table 1: Collected Data

No	X	δ	Z
1	0.0027	1	0
2	0.8412	0	0
3	1.3741	0	1
4	1.8322	1	0
5	2.471	1	1
6	3	0	1

```
> data
       [,1] [,2] [,3]
[1,] 0.0027    1    0
[2,] 0.8412    0    0
[3,] 1.3741    0    1
[4,] 1.8322    1    0
[5,] 2.4710    1    1
[6,] 3.0000    1    1
```

Listing 6: Results of scripts.

At the end of processing this call, the data matrix will be created.

Remark 1 *If we recorded the data into a text file (e.g. 'SampleData.txt') with three columns representing X, δ and Z respectively, then we can directly load the table into R-based matrix as follows:*

```
data <- read.table('SampleData.txt')
```

Listing 7: Loading data from text files.

In the YPmodel package, the name of those text files could be used as the input data name. See section 7 for more details.

3 Parameter Estimation

The model of Yang and Prentice (YPmodel) will be determined by the parameter $\beta = (\beta_1, \beta_2)^T$ and the baseline function $R(t)$, and [Yang and Prentice(2005)] presented a estimator for them.

To perform the estimator, we will use the function ”YPmodel.estimate”.

```
library(YPmodel)
data(gastric)
Estimate <- YPmodel.estimate(data=gastric)
```

Listing 8: Performing estimator.

where the dataframe *Estimate* contains the results of estimator, and Table 4 gives its structures.

Table 2: Dataframe *Estimate* Structure

Variables	Notes
beta	Value of $\hat{\beta}$.
r	Value of $\hat{R}(t, \hat{\beta})$.

If we want to obtain the value of estimates, then we could use the codes

```
> Estimate$beta
          [,1]       [,2]
[1,] 1.600217 -0.9059888
> Estimate$r
 [1]  0.01123526  0.01388395  0.01663907  0.01950618
    0.02249119  0.02560042
 [7]  0.03795434  0.04160842  0.04542013  0.04939834
    0.05355250  0.06705417
[13]  0.07188668  0.07694133  0.09129257  0.09713635
   0.10325975  0.10968023
...
```

Listing 9: Getting value of $\hat{\beta}$.

3.1 Analyzing results for the hazard ratio function

There are a few summary functions available in the package, including estimators. To get summary information about the $S3$ object, we can use

```
summary.YPmodel.estimate(Estimate, interval=0)
```

Listing 10: Summarizing estimates’ results.

or simply,

```
summary(Estimate, interval=0)
```

Listing 11: Summarizing estimates' results.

```
-----------------------------------------------------------
Parameters of short-term and long-term hazard ration
   model

 Adaptive weight (Beta):

          Beta_1 Beta_2
estimates    1.6 -0.906

 Hazard ratio:

          Theta_1 Theta_2
estimates  4.9541  0.4041
```

Listing 12: Summarizing estimates' results.

We can see that $\hat{\beta} = (1.6, -0.906)^T$ and $\hat{\theta} = (4.9541, 0.4041)^T$. Another summary function is available for plotting the survival functions. This can be done as follows:

```
plot.YPmodel.survivor(Estimate)
```

or simply,

```
plot(Estimate)
```

Listing 13: Plotting estimates' results.

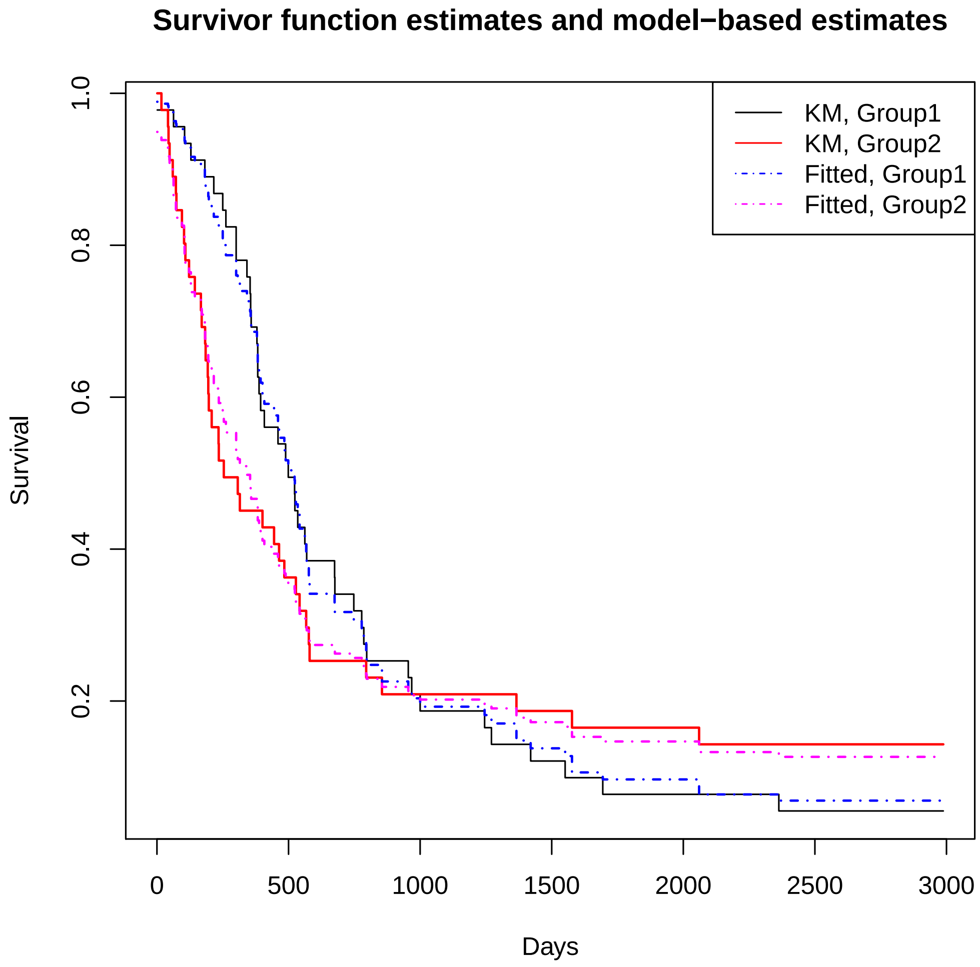


Figure 1: Plotting results.

3.2 Options

In the estimating process, the parameter

- $startPoint$ controls the start point of the iteration for $\hat{\beta}$, and the default value is $(0, 0)$.
- nm controls a bound for $\hat{\beta}$, and the default value is $\log(100)$.

- $maxIter1$ controls out-cycle iteration numbers, and the default value is 50.
- $maxIter2$ controls inner-cycle iteration numbers, and the default value is 20.

which can be customized as follows:

```
YPmodel.estimate(data=gastric, startPoint=c(0.5, 0.5),
    nm=3, maxIter1=100, maxIter2=50)
```

Listing 14: Setting parameter nm for estimator.

On the other hand, we set parameter *interval* to 1 to have confidential bands for $\hat{\beta}$. Then this can be done as follows:

```
YPmodel.estimate(data=gastric, interval=1)
```

Listing 15: Performing estimator with confidential bands.

Or equally,

```
Estimate <- YPmodel.estimate(data=gastric)
```

Listing 16: Performing estimator with confidential bands.

because the default value of *interval* is 1. The new dataframe *Estimate* contains both the estimates and confidential intervals for $\hat{\beta}$, and Table 3 gives structures of those additional values .

Table 3: Additional Dataframe *Estimate* Structure

Variables	Notes
$variance.beta1$	Variance of the first variable of $\hat{\beta}$.
$variance.beta2$	Variance of the second variable of $\hat{\beta}$.

To get summary information about the $S3$ object obtained, we can use

```
summary.YPmodel.estimate(Estimate, interval=1)
```

or simply,

```
summary(Estimate, interval=1)
```

Listing 17: Summarizing estimates' results with confidential bands.

```
---------------------------------------------------------
Parameters of short-term and long-term hazard ration
   model

 Adaptive weight (Beta):

          Beta_1 Beta_2
estimates    1.6 -0.906

 Hazard ratio:

          Theta_1 Theta_2
estimates  4.9541  0.4041

 Confidence Interval (Beta):

       Lower bound   Upper bound
Beta_1      0.5461  2.6543
Beta_2     -1.3947 -0.4173

 Confidence Interval (Theta)

        Lower bound   Upper bound
Theta_1     1.72651 14.2155
Theta_2     0.24791  0.6588
```

Listing 18: Results of scripts.

The corresponding 95% confidential intervals for $\hat{\beta}$ and $\hat{\theta}$ are $([0.5461, 2.6543], [-1.3947, -0.4173])^T$ and $([1.72651, 14.2155], [0.247910.6588])^T$, respectively.

4 Confidence intervals and bands for the hazard ratio function

[Yang and Prentice(2011)] presented the procedures for constructing pointwise confidence intervals and simultaneous confidence bands for the hazard ratio function.

To estimate the confidence intervals and bands for the hazard ratio function, we should know the parameters of the model, where section 3 presents a estimator. The estimating the confidence intervals and bands for the hazard ratio function after estimating parameters in section 3. However, we can directly get the confidence intervals and bands for the hazard ratio function as follows

```
library(YPmodel)
data(gastric)
IntervalBands <- YPmodel.IntervalBands(data=gastric)
```

Listing 19: Performing confidence intervals and bands estimation.

where the dataframe $IntervalBands$ contains the results of estimator, and Table 6 gives its structures. Besides, section 4.2 shows how to customize the parameter estimator.

Table 4: Dataframe $IntervalBands$ Structure

Variables	Notes
hr	Estimation of the hazard ratio function.
$ld2$	Lower bound of the time frame.
$ud2$	Upper bound of the time frame.
$low3$	Lower confidential intervals of the hazard ratio function.
$upp3$	Upper confidential intervals of the hazard ratio function.
$low22$	Lower 95% confidential bands of the hazard ratio function.
$upp22$	Upper 95% confidential bands of the hazard ratio function.
$low90$	Lower 90% confidential bands of the hazard ratio function.
$upp90$	Upper 90% confidential bands of the hazard ratio function.

If we want to obtain the value of $IntervalBands$, it can be done with similar codes in section 7.

4.1 Analyzing results for Estimating confidence intervals and bands

To get summary information about the $S3$ object, we can use

```
summary.YPmodel.IntervalBands(IntervalBands)
```

or simply,

```
summary(IntervalBands)
```

```
-----------------------------------------------------
Point estimates, Pointwise confidence intervals, and
   confidence bands of short-term and long-term hazard
   ration model

          Days         HR_fun  lower.cl   upper.cl
             lower.95%band upper.95%band
 [1,]  103.0000    3.1509    5.6889    1.7452
    7.8822        1.2596
 [2,]  105.0000    2.9014    5.0425    1.6694
    6.8412        1.2305
 [3,]  108.0000    2.8228    4.8091    1.6569
    6.4531        1.2348
 [4,]  122.0000    2.7457    4.5932    1.6412
    6.1018        1.2355
 [5,]  129.0000    2.5513    4.1564    1.5660
    5.4412        1.1962
 [6,]  144.0000    2.4811    3.9868    1.5440
    5.1798        1.1884
 [7,]  167.0000    2.4123    3.8296    1.5195
    4.9425        1.1773
 [8,]  170.0000    2.3448    3.6839    1.4925
    4.7270        1.1632
 [9,]  182.0000    2.1983    3.3989    1.4218
    4.3230        1.1179
[10,]  183.0000    2.1365    3.2799    1.3917
   4.1553        1.0985
[11,]  185.0000    2.0759    3.1687    1.3600
   4.0017        1.0769
     lower.90%band upper.90%band
 [1,]        7.3530        1.350
 [2,]        6.4105        1.313
 [3,]        6.0611        1.315
 [4,]        5.7434        1.313
 [5,]        5.1376        1.267
 [6,]        4.8987        1.257
 [7,]        4.6809        1.243
 [8,]          4.4824          1.227
 [9,]          4.1070          1.177
[10,]          3.9509          1.155
[11,]          3.8075          1.132
...
```

Listing 20: Results of scripts.

Another summary function is available for plotting the survival functions. This can be done as follows:

```
plot.YPmodel.IntervalBands(IntervalBands)
```

or simply,

```
plot(IntervalBands)
```

Listing 21: Plotting estimates results of confidence intervals and bands.

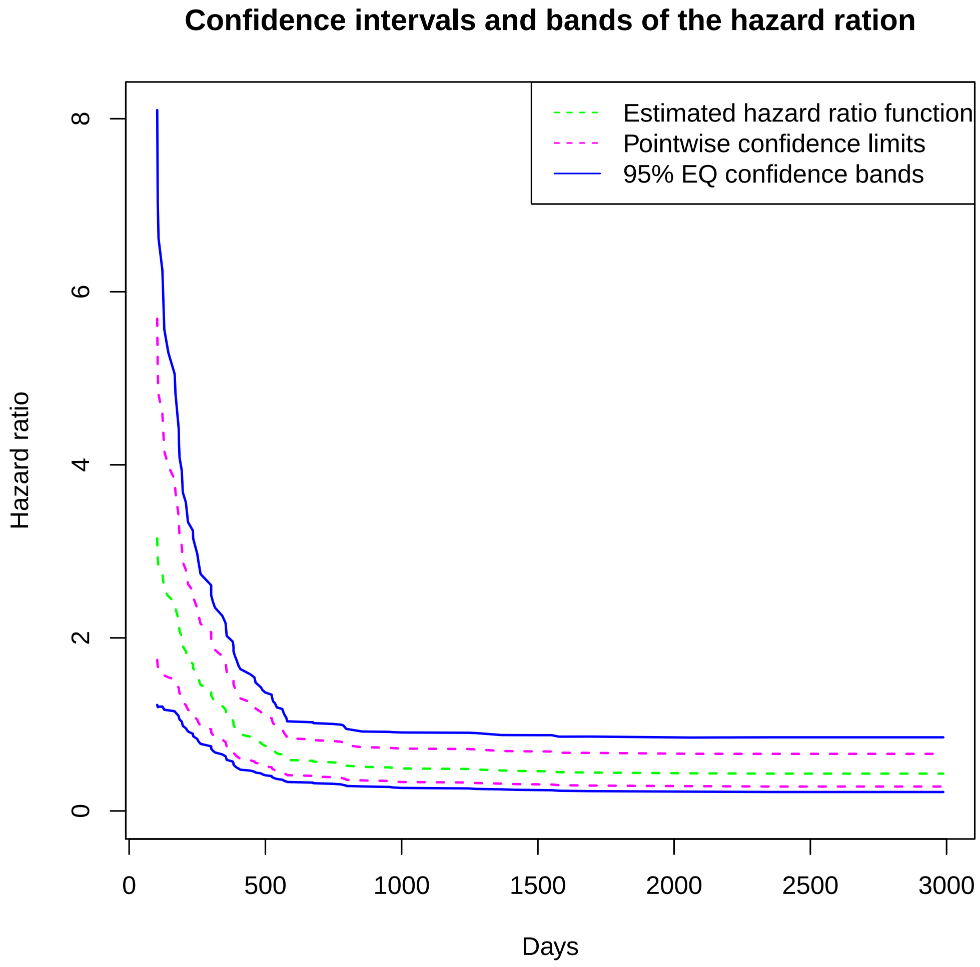


Figure 2: Plotting results.

4.2 Options

Since the results of confidence intervals and bands depend on the estimation in section 3, we can customize the estimates into the above functions as follows:

```
library(YPmodel)
data(gastric)
```

```
Estimate <- YPmodel.estimate(data=gastric, startPoint=c
    (0.5, 0.5), nm=3, maxIter1=100, maxIter2=50, interval
    =0)
IntervalBands <- YPmodel.IntervalBands(data=gastric,
    Estimate=Estimate)
```

Listing 22: Performing confidence intervals and bands estimation with customized estimates.

5 Adaptively weighted log-rank test

For testing for treatment effects with time-to-event data, the logrank test is the most popular choice and has some optimality properties under proportional hazards alternatives. [Yang and Prentice(2009)] showed that the logrank test and related tests can be improved by using weighted logrank statistics with adaptive weights.

We can directly perform the adaptively weighted log-rank test in [Yang and Prentice(2009)] as follows:

```
library(YPmodel)
data(gastric)
Adlgrk <- YPmodel.adlgrk(data=gastric)
```

Listing 23: Performing the adaptively weighted log-rank test.

where the dataframe *Adlgrk* contains the results of the test, and Table 6 gives its structures.

Table 5: Dataframe *Adlgrk* Structure

Variables	Notes
pval	P-value from adaptively weighted logrank test.

To get summary information about the $S3$ object obtained, we can use

```
summary.YPmodel.adlgrk(Adlgrk)
```

or simply,

```
summary(Adlgrk)
```

Listing 24: Summarizing results of the adaptively weighted logrank test.

```
----------------------------------------------------------
Improved Logrank-Type Tests (p-value):
0.0304
```

Listing 25: Results of scripts.

Because the estimation in section 3 will be used in the hypothesis test, we can customize the estimates into the above functions as follows:

```
library(YPmodel)
data(gastric)
Estimate <- YPmodel.estimate(data=gastric, startPoint=c
    (0.5, 0.5), nm=3, maxIter1=100, maxIter2=50, interval
    =0)
Adlgrk <- YPmodel.adlgrk(data=gastric, Estimate=Estimate)
```

Listing 26: Performing confidence intervals and bands estimation with customized estimates.

6 Two Lack-of-fit Tests

[Yang and Zhao(2012)] proposed two omnibus tests for checking this model, based, respectively, on the martingale residuals and the contrast between the non-parametric and model-based estimators of the survival function. These tests are shown to be consistent against any departure from the model.

We use the data from section 2.1, and the two lack-of-fit tests can be done as follows:

```
library(YPmodel)
data(gastric)
LackFitTest <- YPmodel.lackfittest(data=gastric)
```

Listing 27: Performing the two lack-of-fit tests.

where the dataframe $LackFitTest$ contains the two lack-of-fit tests, and Table 6 gives its structures.

If we want to obtain the value of $LackFitTest$, it can be done with similar codes in section 3.

Table 6: Dataframe *LackFitTest* Structure

Variables	Notes
newBest	Value of $\hat{\beta}$ used in the two tests.
*pvalu*1	P-value from martingale residual-based test.
*pvalu*2	P-value from contrast-based test.

6.1 Analyzing results for Estimating confidence intervals and bands

To get summary information about the $S3$ object, we can use

```
summary.YPmodel.lackfittest(LackFitTest)
```

or simply,

```
summary(LackFitTest)
```

Listing 28: Summarizing performance of the two lack-of-fit tests.

```
---------------------------------------------------------
Lack-of-fit tests for checking short-term and long-term
   hazard ration model

 Adaptive weight (Beta, sample odds function estimator
    using only the control group data):

          Beta_1 Beta_2
estimates  1.712 -0.949

 Residual, the martingale residual-based test (p-value):
0.119

 Contrast, the contrast-based test (p-value):
0.615
```

Listing 29: Results of scripts.

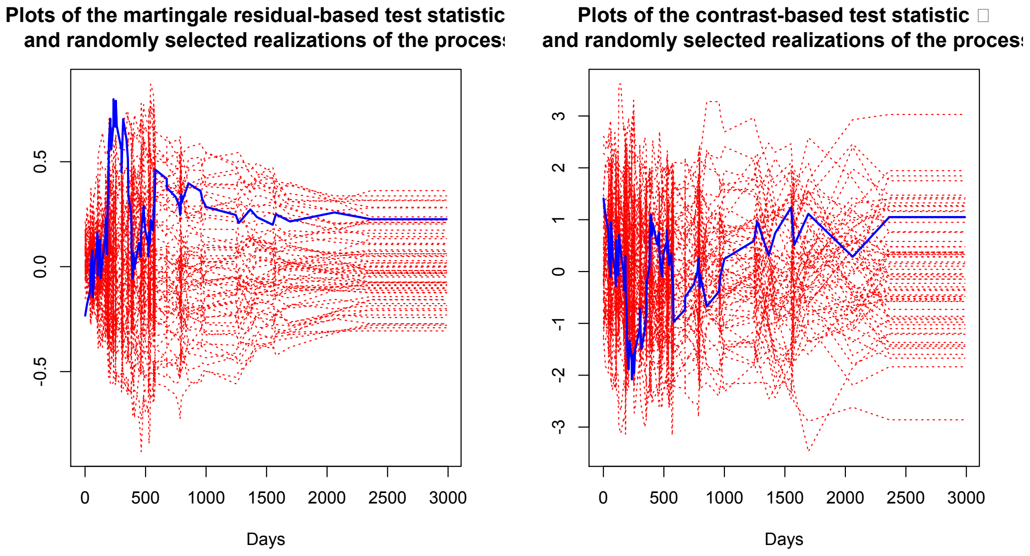
Another summary function is available for plotting the survival functions. This can be done as follows:

```
plot.YPmodel.martint(LackFitTest)
plot.YPmodel.survf(LackFitTest)
```

or simply,

```
plot(LackFitTest)
```

Listing 30: Plotting estimates process of the two lack-of-fit tests.


Plots of the martingale residual-based test statistic
and randomly selected realizations of the proces

Plots of the contrast-based test statistic
and randomly selected realizations of the proces

Figure 3: Plotting results.

6.2 Options

The parameter *repNum* sets number of random variables to be used the two lack-of-fit tests, and its default value is 1000. Assume we would like to use only 100 random variables instead. This can be done as follows:

```
LackFitTest <- YPmodel.lackfittest(data=gastric, repNum
    =100)
```

Listing 31: Loading YPmodel package and performing estimating process.

7 All-in-one Function

In the section 2 - 6, we see how different functions can be called to analysis survival data with the model of Yang and Prentice (YPmodel) (using data in section 2.1). The whole process (with default parameters) can be summarized with

```
library(YPmodel)
data(gastric)
Estimate <- YPmodel.estimate(data=gastric)
IntervalBands <- YPmodel.IntervalBands(data=gastric,
   Estimate=Estimate)
LackFitTest <- YPmodel.lackfittest(data=gastric)
Adlgrk <- YPmodel.adlgrk(data=gastric)
```

Listing 32: Summary of functions.

To simplify those procedures, we also provide one all-in-one function for YPmodel.

```
library(YPmodel)
data(gastric)
result <- YPmodel(gastric)
```

Listing 33: All-in-one Function.

where the parameters can be customized as follows:

```
result <- YPmodel(data=gastric, startPoint=c(0.5,0.5), nm
   =3, maxIter1=100, maxIter2=50, repNum=2000)
```

Listing 34: All-in-one Function with customized parameters.

And the S3 function "plot" and "summary" can be used to demonstrate the all four above results

```
summary(result)
plot(result)
```

Listing 35: Demonstrating the overall results.

Remark 2 *We can directly load the text file when using function "YPmodel" as follows:*

```
result <- YPmodel('SampleData.txt')
```

Listing 36: Performing YPmodel with text files.

8 Conclusions

We have implemented the methodology described above in an R package, called YPmodel. This would be used by analysts and others whose task is to analysis of the comparison of failure times between a treated and control group under inde- pendent censorship.

To estimate survival data using YPmodel, the user will usually follow three basic steps in the following order:

1. Preparing data set in section 2.

2. Estimate the model parameters in section 3.

3. Estimate confidence intervals and bands for the hazard ratio function in section 4.

4. Perform the Adaptively weighted log-rank test in section 5.

5. Perform the lack-of-fit tests in section 6.

and section 7 provides an all-in-one method to use step 2 - 5. Plus, each step can be independently performed. Use "help(function)" for more specifics about each function described below and details about its arguments.

9 Appendix

There are several parameters used by the package. The correspondence of these parameters to their argument names and default values used in YP-model version 1.0 is shown in Table 7.

Table 7: Dataframe *Adlgrk* Structure

Parameters	Default values	Notes
$startPoint$	$c(0,0)$	Define the start point of the iteration for $\hat{\beta}$.
nm	log(100)	Define a bound for $\hat{\beta}$.
$maxIter1$	50	Define out-cycle iteration numbers.
$maxIter2$	20	Define inner-cycle iteration numbers.
$repNum$	1000	Number of iterations in the two lack-of-fit tests.

References

[Gastrointestinal Tumor Study Group: Schein(1982)] Bruckner H. W. Douglass H. O. Mayer R. et al. Gastrointestinal Tumor Study Group: Schein, P. D. A comparison of combination chemotherapy and combined modality therapy for locally advanced gastric carcinoma. *Cancer*, (9), 1982.

[Team et al.(2008)] RDevelopment Core Team et al. R: A language and environment for statistical computing. *R Foundation Statistical Computing*, 2008.

[Yang and Prentice(2005)] Song Yang and Ross Prentice. Semiparametric analysis of short-term and long-term hazard ratios with two-sample survival data. *Biometrika*, 92(1):1–17, 2005.

[Yang and Prentice(2009)] Song Yang and Ross Prentice. Improved logrank-type tests for survival data using adaptive weights. *Biometrics*, 66(1): 30–38, 2009.

[Yang and Prentice(2011)] Song Yang and Ross L Prentice. Estimation of the 2-sample hazard ratio function using a semiparametric model. *Biostatistics*, 12(2):354–368, 2011.

[Yang and Zhao(2012)] Song Yang and Yichuan Zhao. Checking the short-term and long-term hazard ratio model for survival data. *Scandinavian Journal of Statistics*, 2012.