Medical University of Vienna

Core Unit for Medical Statistics and Informatics
Section of Clinical Biometrics
Section Head: Prof. M. Schemper
A-1090 VIENNA, Spitalgasse 23

Phone: (+43)(1) 40400/6688
Fax: (+43)(1) 40400/6687
http://www.meduniwien.ac.at/msi/biometrie

Technical Report 1/2006

FC06: A SAS® macro for Cox regression with Firth's penalization

Georg HEINZE

e-mail: georg.heinze@meduniwien.ac.at

Abstract

The phenomenon of monotone likelihood is observed in the fitting process of a Cox model if the likelihood converges while at least one parameter estimate diverges to ± infinity. Monotone likelihood primarily occurs in small samples with several unbalanced and highly predictive covariates, and with a high percentage of censoring. A procedure by Firth (1993) originally developed to reduce the bias of maximum likelihood estimates is shown to provide an ideal solution to monotone likelihood (cf. Heinze & Schemper, 2001). It produces finite parameter estimates by means of penalized maximum likelihood estimation. Corresponding Wald tests and confidence intervals are available but it is shown that penalized likelihood ratio tests and profile penalized likelihood confidence intervals are often preferable.

The SAS macro FC (Heinze & Ploner, 2002) implements Firth's penalization to Cox regression and has been available since 2000. This macro was restricted to time-invariant covariates and effects. A new SAS macro program FC06 was written to enhance the functionality of its predecessor FC by providing options that allow to fit models including time-varying covariates and time-dependent effects. It is even possible to use the macro for conditional logistic regression analysis of $1 : m$ matched case-control studies. The present report contains the complete User's Guide to this macro including syntax, computational methods and examples.

Contents

1 Overview

The SAS macro `%FC06` was written to facilitate application of Firth's penalization procedure (Firth, 1993) in Cox regression analysis. Heinze & Schemper (2001) show that, given some weak regularity conditions, Firth's procedure applied to Cox regression (FC) removes the problem of monotone likelihood.

Supplied with a SAS data set as input, the macro output contains FC-type Cox regression parameters, standard errors, confidence limits, *p*-values, the value of the maximized penalized log likelihood, and the number of iterations needed to arrive at the maximum. The parameter estimates, the maximized penalized log likelihood and the estimated covariance matrix are stored in a SAS data set that has the same structure as the output data set that can be obtained by SAS/PROC PHREG (http://support.sas.com/onlinedoc/913/getDoc/en/statug.hlp/phreg_index.htm) using the OUTEST option in the PROC PHREG statement. Furthermore, parameter estimates, confidence limits and *p*-values are stored in a second data set. The structure of this data set resembles the output table written to the output window. The core routine `fc06.dll` was written in FORTRAN.

The profile of the penalized log likelihood function for any parameter can be plotted to judge the adequacy of Wald confidence intervals. Multiple input data sets can be efficiently processed using grouping variables similarly to PROC PHREG. Offset values can be used to exclude particular model parameters from the estimation process.

New features of FC06

`%FC06` offers some new features compared to its predecessor (`%FC`). These are at a glance:

- Time-dependent effects of covariates can be included in the model by specifying interactions of such covariates with arbitrarily complex functions of time.
- The optional counting process formulation similar to SAS/PROC PHREG allows to incorporate

  – time-varying covariates

  – left-censoring (delayed entry)

  – recurrent events

  and to

  – stratify the analysis for the levels of particular variables

  - estimate a conditional logistic regression model for $1 : m$-matched case-control studies using penalized maximum likelihood

- The ability of the macro to handle large data sets has been improved by offering an alternative to the imperative DLL call used in the predecessor of the macro

2 Installation

2.1 What's inside the ZIP file?

The `fc06.zip` file contains the following files needed for running the macro:

File name	Description
fc06.sas	SAS macro code
fc06.dll	dynamic link library containing core routines of the program
fc06exe.exe	executable console application
fc06.def	definition table needed by SAS to access fc06.dll
dforrt.dll	run-time dynamic link library needed by fc06.dll
breast.sas	the breast cancer data set

2.2 Step-by-step installation guide

The installation is described assuming a computer running on Windows XP and SAS Version 9. If you have troubles with the installation which may be due to a different configuration please contact the author. Since the core routines reside in a dynamic link library that has to be accessed by SAS, some installation steps are necessary. Please read these instructions carefully, and follow them point-by-point.

If for any reason the CALL=DLL option does not work with your configuration, you may use the CALL=EXE option instead. In this case you can omit steps 3 and 4.

1. (a) Create a subfolder in your SAS working directory for DLLs (e. g. `c:\MYFOLDER\DLLs`). Copy the files `fc06.def`, `fc06.dll` and `fc06exe.exe` into that subfolder.

   (b) In file `fc06.sas` change the default value of the path option in the macro header to `path=%str(c:\MYFOLDER\DLLs\)` and save `fc06.sas`.

2. Copy the file `'dforrt.dll'` into your `C:\WINDOWS\system32` folder (`C:\Winnt\system32` under Windows NT/2000) if it's not already there.

3. In the following, we assume that SAS 9 has been installed into the folder `C:\Program Files\SAS\SAS 9.1\`. In that folder, you will find a file called

`SASV9.CFG`. Open it, it points to one or several other `SASV9.CFG` files which contain information that is used each time SAS is opened. If you have only an english version of SAS 9 installed, then the file reads like

```
-CONFIG "C:\Program Files\SAS\SAS 9.1\nls\en\SASV9.CFG"
```

Now open `C:\Program Files\SAS\SAS 9.1\nls\en\SASV9.CFG`.

After the two lines

```
    /* Setup the SAS System load image search paths definition */
    -PATH           (
```

insert the line

```
                                 "c:\MYFOLDER\DLLs"
```

and save `C:\Program Files\SAS\SAS 9.1\nls\en\SASV9.CFG`. This point has to repeated for each file that appears in the CONFIG statement of `C:\Program Files\SAS\SAS 9.1\SASV9.CFG`.

4. Restart SAS 9.

3 Working with the macro

3.1 Syntax

The following options are available in `%FC06`:

```
%fc06(<data=SAS data set,>
<time=variable,>
<time1=variable,>
<time2=variable,>
<cens=variable,>
<censval=value,>
varlist=variables,
<ft=expression,>
<ftmap=value(s),>
<tdenames=string,>
<pl=value,>
<risk=value,>
<profile=variables,
<profsel=value,>
<profser=value,>
<profn=value,>
<outprof=SAS data set,>
<outmod=SAS data set,>
<global=SAS data set,>
<outest=SAS data set,>
<outtab=SAS data set,>
<outs0=SAS data set,>
<s0method=string,>
<print=value,>
<test=variables,>
<testtype=string,>
<call=string,>
<path=string,>
<maxit=value,>
<maxhs=value,>
<epsilon=value,>
<maxstep=value>
<by=variables,>
```

<notes=*value*,>
<offset=*SAS data set*>);

These options are described in the subsequent sections.

3.2 Basic options

- `data=`*SAS data set* names the input SAS data set. The default value is `_LAST_`.
- `time=`*variable* names a variable containing survival times. The default value is `time`.
- `cens=`*variable* names a variable containing the censoring indicator for each survival time. Default value is `cens`.
- `censval=`*value* names the censoring value. The default value is 0, meaning that if `censval`=0, then the corresponding survival time is treated as censored.
- `varlist=`*variables* names a list of independent variables, separated by blanks. There is no default value. This option is required.
- `pl=`*value* requests profile penalized likelihood confidence intervals and penalized likelihood ratio tests for parameters if set to 1 (fast algorithm of Venzon & Moolgavkar, 1988) or 2 (slow but robust simple binary search). With some data sets we noticed numerical problems when `pl` is set to 1. These problems disappear if you use `pl=2`. However, `pl=2` is only allowed with `call=DLL` (which is the default value). If `pl`=0, Wald confidence intervals and tests will be computed. Default value is 1.
- `profile=`*variables* requests a plot of the profile penalized log likelihood function for all effects specified in this options. Of course, these effects have to appear either in the `varlist` or in the `tdenames` options, otherwise the macro will run into an error. The x-axis ranges for this plot are automatically chosen by the macro, but can also be specified by the user in terms of standard errors to the left and right from the point estimate (options `profsel` and `profser`, respectively). Also the number of profile likelihood evaluations (`profn`, default value=100) can be chosen. If the `profile` options is used, then the data set specified in option `outprof` will consist of the variables
  - `_name_` containing the covariate's name
  - `_b_` containing the values on the x-axis (the values for β)
  - `_profli_` containing the values of the profile penalized log likelihood

- `_normal_` containing the values $\ell_{\max} - 0.5(\beta - \hat{\beta})^2/\hat{\sigma}^2$ (where $\ell_{\max}$ is the maximized penalized log likelihood) which represent the Wald (normal) approximation to the profile penalized log likelihood
- `_refer_` containing the reference line (the values of β where the profile penalized log likelihood function and normal approximation crosses the reference line are the profile penalized likelihood and Wald confidence limits, respectively).
- any BY-variables specified

3.3 Counting process style of input

The macro adopts the counting process formulation of Cox's model from SAS/PROC PHREG. In this formulation, the data for each subject can be represented by multiple observations, each identifying a semiclosed time interval (`time1, time2`], the values of the explanatory variables over that interval, and the event status at `time2`. The subject remains at risk during the interval (`time1, time2`], and an event may occur at `time2`. Values of the explanatory variables for the subject remain unchanged in the interval.

- `time1=`*variable* and `time2=`*variable* name variables containing the the endpoints of a semi-closed interval (`time1,time2`] during which the subject is at risk. Specification of `time2` overrules any specification of the option `time`. Option `time1` has the default value 0.

3.4 Time-dependent effects

Some new options allow the definition of interactions of covariates contained in the option `varlist` and arbitrary functions of time. These options are:

- `ft=`*expression* specifies functions of time $f(t)$ that are used to define such interactions. Any expression compatible to SAS syntax is allowed. Time is represented by the string `_TIME_`. E. g., if $f(t) = \log(t)$, specify `ft=log(_TIME_)`.
- `ftmap=`*values* maps the functions of time to the covariates specified in macro option `varlist`.
- `tdenames=`*string* assigns names to the interactions defined by `ft` and `ftmap`. These names appear in the printed output and the output data sets of the macro.

A simple example

Consider the model $h_i(t) = h_0(t)\exp(\beta_1 x_i + \beta_2 y_i + \beta_3 y_i \log(t) + \beta_4 z_i + \beta_5 z_i t)$. This example uses two different functions $f(t)$ for the time-dependent effects of y and z. The model is estimated by specifying

```
varlist=x y z,
ft=log(_time_) _time_,
ftmap=2 3,
tdenames=y_logt z_t,
```

The last option, `tdenames`, assigns the variable names `y_logt` and `z_t` to the interactions of $\log(t)$ with y and t with z, respectively.

3.5 Output options

- `risk=`*value* requests estimated relative risks and confidence intervals to be included in the output table if set to 1. Default value is 0.
- `print=`*value* suppresses printed output if set to 0. Default value is 1.
- `outmod=`*SAS data set* names a SAS data set containing, for each BY group, the number of iterations (`_it_`), the null penalized log likelihood (`_penli0_`), the maximized penalized log likelihood (`_penlik_`), the global penalized likelihood ratio χ^2 test statistic (`_modchi_`), the associated P-value (`_p_`), the global scores statistic and P-value (`_scorechi_` and `_scop_`), the global Wald statistic and P-value (`_waldchi_` and `_waldp_`), number and percentage of non-censored and censored observations (`_events`, `_cens_`, `_pev_` and `_pce_`, respectively), the total number of observations (`_nobs_`), and, if used, the variables specified in the `by` option.
- `global=`*SAS data set*— names a SAS data set containing, for each BY group, the likelihood ratio, the scores and the Wald test of the global null hypothesis that all model parameters are zero.
- `outest=`*SAS data set* names a SAS data set containing parameter estimates, penalized log likelihood and covariance matrix. There is no default value. The data set contains one variable for each explanatory variable in the `varlist` option. The `outest` data set contains one observation for each `by` group containing the FC-type estimates of the regression coefficients. Additionally, there are observations containing the rows of the estimated covariance matrix of the parameter estimators for each BY–group. The `outest` data set contains the following variables:

  - any BY variables specified
  - one variable for each explanatory variable in the `varlist` option.
  - `_penlik_`, the maximized penalized log likelihood at the FC estimate
  - `_TYPE_`, a character variable of length 8 with two possible values: PARMS for parameter estimates or COV for covariance estimates
  - `_NAME_`, a character variable of length 8 containing the name of the `time` variable for parameter estimates or the name of each explanatory variable for the covariance estimates

- `outtab=`*SAS data set* names a SAS data set containing parameter estimates, standard errors, confidence limits and p–values. The default value is `_OUTTAB`. The data set contains one observation per explanatory variable and BY–group. It contains the following variables:

  - any BY variables specified
  - `_var_`, the subsequent number for each explanatory variable in the `varlist` option
  - `_name_`, the name of each explanatory variable in the model (as specified in the `varlist` option)
  - `_beta_`, the FC parameter estimates
  - `_stderr_`, the estimated standard error of the corresponding parameter estimate
  - `_bstd_`, the standardized parameter estimate
  - `_lo_`, the lower confidence limit for the parameter estimate
  - `_up_`, the upper confidence limit for the parameter estimate
  - `_p_`, the p–value for $H_0 : \beta_r = 0$.

  The method of computation of confidence intervals and p-values (Wald or profile penalized likelihood) can be controlled using the `pl` option. The significance level of the intervals can be set by the `alpha` option.

3.6 Baseline survival function

Two options control the output of a SAS data set containing the baseline survival function, i. e. the estimated survival curve corresponding to an individual with a zero covariate vector.

- `outs0=`*SAS data set* names the SAS data set with the baseline survival function. This data set is produced using the baseline statement of a PROC PHREG. This data set contains any variables specified in macro option `by` and the variables `_surv_`, `_cumhaz_`, `_loglogs_`, and `_time_`, denoting the baseline survival function, the cumulative hazard function, the $\log(-\log(S))$ function and the time, respectively.

- `s0method=`*string* specifies the method used to compute the survivor function estimates. The two available methods are

  - `CH|EMP|NELSON` specifies that the Nelson (empirical) cumulative hazard function estimate of the survivor function is to be computed; that is, the survivor function is estimated by exponentiating the negative empirical cumulative hazard function.

  - `PL` specifies that the product-limit estimate of the survivor function is to be computed. The default is METHOD=PL.

Note that survival functions are not computed if the counting process style of input is used, and can be computed but are not correct if time-dependent effect were defined.

3.7 Model fitting options

- `test=`*variables* requests a test of the null hypothesis that all parameters corresponding to effects (appearing in `varlist` or `tdenames`) listed in the `test` option are zero. The type of this test (penalized likelihood ratio or scores test) can be chosen by the macro option `testtype`.

- `testtype=`*string* specifies the type of test that the `test` option should perform. Two values are allowed: while `testtype=LR|L` requests a (penalized) likelihood ratio test, `testtype=SCORES|S` a scores test.

- `call=`*string* specifies by which way the core computations are performed. With `call=DLL`, a dynamic link library (DLL) is called using PROC IML's CALL MODULE statement. Specifying `call=EXE`, an external application is executed by SAS's X command. While the DLL option is more comfortable to the user (provided that the installation instructions have been followed carefully), it has some size limitations and may produce an error if the input data set is too large. The EXE option imposes no relevant limitations, but needs to write and read temporary files to the hard disc and when the application is called by the X command, a 'black' console window pops up. Default setting is `call=DLL`.

- `path=`*string* specifies the folder (including a final backslash) where fc06.exe (part of the distribution) has been saved. This option is needed if `call=EXE` is used. It is recommended to use the `%STR( )` function, e. g. `path=%str(c:\MYFOLDER\DLLs\)`.

- `maxit=`*value* specifies the maximum number of iterations. Default value is 25.

- `maxhs=`*value* specifies the maximum number of step-halvings allowed in one iteration. Default value is 2.

- `epsilon=`*value* specifies the maximum allowed change in penalized log likelihood to declare convergence. Default value is 10^{-6}.

- `maxstep=`*value* specifies the maximum change of (standardized) parameter values allowed in one iteration. Default value is 1.

3.8 Options useful for simulation

- `by=`*variables* requests separate analyses on observations in groups defined by the BY variable(s).

- `offset=`*SAS data set* names an input data set containing offset values of parameter estimates. This data set should contain the same variables as are specified in `varlist` and, if the `by`–option is used, the variable(s) specified in this option. Therefore the `offset` data set should have as many observations as there are BY–groups in the input data set. If a particular parameter in a particular BY–group should be estimated, then its value should be missing in the `offset` data set, otherwise the parameter will be treated as fixed at the value found in the `offset` data set. If a variable contained in `varlist` ist not defined in the `offset` data set, its parameter value will be estimated in any BY–group.

- `notes=`*value* If set to 1, requests a notification in the log file about the number of the BY–group that is currently processed (default=0).

3.9 Titles

Titles 1–3 are not used by the macro. These titles can be set by the user in a statement before the macro call. Titles 4 and 5 are used by the macro. These titles are deleted on exit.

3.10 Printed output

Unless `print=0`, printed output usually consists of four pages. The first page includes

- the name of the input data set
- the name of the variable containing survival times
- the name of the variable containing the censoring indicator values
- the censoring value
- a list of time-invariant and time-dependent covariates
- a message on where estimates, confidence limits and covariance matrix have been stored to

The second page includes

- the number of iterations needed to arrive at the maximum of the penalized log likelihood
- the value of the maximized penalized log likelihood
- the value of the null penalized log likelihood
- a summary of the number and percentage of events and censored observations (note that when using the counting process formulation, these numbers correspond to input data lines, not to individuals)

The third page includes a summary of three tests (likelihood ratio, scores and Wald) testing the global null hypothesis that all parameters are 0. Finally, the fourth page includes a table containing variable names, FC parameter estimates and associated estimated standard errors, confidence intervals for the parameters and p-values. If the `risk` option was set to 1, then an additional page includes a table with the estimated risk ratios and associated confidence intervals.

If a special penalized likelihood ratio test for testing more than one parameter at a time was requested by using the `test` option, an additional page gives information on the χ^2-statistic for testing the hypothesis that all parameters listed in the `test` option are 0, and the associated degrees and freedom and p-value.

All pages except the first one are repeated for all BY-groups if the `by`-option was used.

3.11 A computational issue

The program assumes that two survival times are equal if the absolute value of their difference is smaller than or equal to 10^{-5}. This is of importance if the counting process formulation is used: the values of `time1` and `time2` of one data line must differ by at least $2 \cdot 10^{-5}$, otherwise the line is ignored by the program.

4 Examples

4.1 Introduction

Use of %FC06 is exemplified using a breast cancer data set (Loesch et al., 1998). Survival times of 100 patients were recorded (74 of them censored) and also values of four potential risk factors: tumour stage (variable T), nodal status (N), histological grading (G) and Cathepsin D immunoreactivity (CD). For analysis these factors were dichotomised to levels of 0 and 1 (unfavourable). The survival and censoring times are stored in variable time, with the censoring indicator CENS, 0 indicating a censored survival time. Assume that the data has been stored in SAS data set breast.

4.2 Simple analysis

To obtain an FC analysis with a table containing variable names, parameter estimates, standard errors, confidence limits and p-values, one submits the macro call

```
%fc06(data=breast, varlist=t n g cd, time=time, cens=cens);
```

The following output is produced (*Page 1*):

```
  FFFFF  CCC  000   666       Cox regression
  F     C    0   0 6          using penalized maximum likelihood estimation
  FFFF  C    0   0 6666
  F     C    0   0 6   6
  F      CCC  000   666



  Author:                     Georg Heinze
  Version:                    2006.01 (beta)

  Documentation:              Heinze, G. (2006).
                              Technical Report 1/2006:
                              fc06: A program for estimating Cox models
                              using Firth's penalization
                              Section of Clinical Biometrics,
                              Core Unit for Medical Statistics and Informatics,
                              Medical University of Vienna.
```

```
Data set:                   BREAST
Dependent variable:         TIME
Censoring indicator:        CENS
Censoring value:            0
Ties handling:              Breslow
Time-invariant effects:     t n g cd

Table with parameter estimates saved in _TAB.
Estimates and covariance matrix saved in _EST.
```

Page 2:

```
                      Model fitting information

                                        Null
                         Penalized    penalized    Number
                            log          log         of
           Iterations   likelihood   likelihood    events

                8        -89.2824     -107.258       26



                        Number of       %          %
           Censored    observations   events    censored

              74          100           26         74
```

Page 3:

```
                 Testing global null hypothesis: beta=0

                                        Degrees
                              Chi-        of        Pr >
         Test                Square     freedom   Chi-Square

         Likelihood Ratio    35.9514       4        <.0001
         Scores              40.7348       4        <.0001
```

```
           Wald                23.2001       4         0.0001
```

Page 4:

```
       FC estimates, profile penalized likelihood confidence limits
                   and penalized likelihood ratio tests

          Parameter  Standard  Standardized    Lower     Upper      Pr >
Variable   estimate    error     estimate    95% c.l.  95% c.l.  Chi-Square

   T       1.22444    0.49160     0.60619     0.30950   2.24836    0.0082
   N       0.91889    0.42257     0.42864     0.11374   1.76351    0.0253
   G       2.42441    1.47355     1.06343     0.38231   7.28066    0.0136
   CD      0.39712    0.44186     0.18198    -0.46702   1.25612    0.3645
```

4.3 Wald tests and confidence limits

To obtain Wald tests and confidence limits for all covariates based on the inverse Fisher information matrix, one submits the call

```
%fc06(data=breast, time=time, cens=cens, censval=0, varlist=t n g cd,
    pl=0);
```

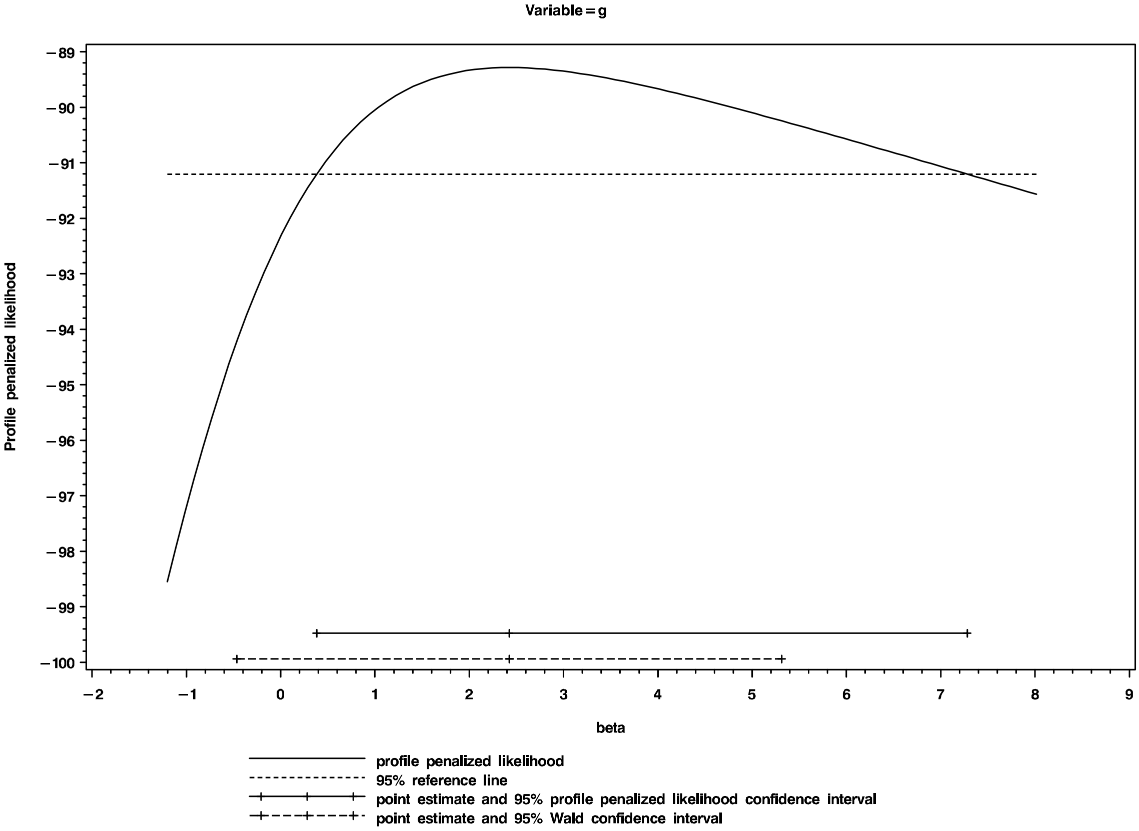
leading to the following output (compare to the output of p. 17):
Output omitted

```
          FC estimates and Wald confidence limits and tests

         Parameter Standard Standardized   Lower    Upper     Pr >
Variable  estimate   error    estimate   95% c.l. 95% c.l. Chi-Square

   T      1.22444   0.49160    0.60619    0.26091  2.18796   0.0127
   N      0.91888   0.42257    0.42864    0.09066  1.74711   0.0297
   G      2.42442   1.47354    1.06343   -0.46368  5.31251   0.0999
   CD     0.39711   0.44186    0.18198   -0.46891  1.26313   0.3688
```

Figure 1: Profile penalized likelihood function for parameter β_G of breast cancer example.



4.4 Plotting the profile penalized log likelihood function

To obtain the profile of the penalized log likelihood as a function of parameter β_G, the `profile` option can be used.

```
%fc06(data=breast, time=time, cens=cens, varlist=t n g cd,
    profile=g);
```

100 equally spaced values of the penalized log likelihood are plotted against the parameter values of β_G using PROC GPLOT (Fig. 1).

4.5 Time-dependent effect

In the following macro call, an interaction of `N` and $\log(t)$ is specified to assess a possible time-dependent effect of `N`:

```
%fc06(data=breast, time=time, cens=cens, varlist=t n g cd,
 ft=log(_TIME_), ftmap=2,
 tdenames=n_logt);
```

The first page of output now lists all time-invariant and time-dependent effects:

```
  Time-invariant effects:     t n g cd
  Time-dependent effects:
                              n_logt = n * log(_TIME_)
```

The following output table suggests a marginally significant interaction of `N` with $\log(t)$:

```
       FC estimates, profile penalized likelihood confidence limits
                   and penalized likelihood ratio tests

          Parameter  Standard  Standardized   Lower     Upper      Pr >
Variable   estimate    error     estimate    95% c.l.  95% c.l.  Chi-Square


 T          1.17592  0.49272     0.58217     0.25965   2.20267    0.0113
 N         -1.51799  1.56134    -0.70811    -4.91887   1.14977    0.2768
 G          2.40396  1.47295     1.05446     0.35896   7.26074    0.0149
 CD         0.44738  0.44073     0.20502    -0.41312   1.30421    0.3052
 N_LOGT     0.77224  0.47422     0.36023    -0.03736   1.80743    0.0621
```

4.6 Simultaneous test of parameters

The `test` option can be used to test the simultaneous effect of more than one effect on survival. In our model, to test the hypothesis that `N` has no effect on survival, the following macro call is submitted:

```
%fc06(data=breast, time=time, cens=cens, varlist=t n g cd,
 ft=log(_TIME_), ftmap=2,
 tdenames=n_logt, test=n n_logt);
```

On the fifth page of output, the test results are displayed:

```
                Penalized likelihood ratio tests for parameters


                                      Degrees
                 Tested                  of         Pr >
               parameters   Chi-Square  freedom   Chi-Square


               n n_logt        7.38138     2        0.024955
```

4.7 Interactions of covariates with more complex functions of time

The `ft` and `ftmap` options may be used to specify even more complex functions of time, consider e. g.

```
%fc06(data=breast, time=time, cens=cens, varlist=t n g cd,
 ft=log(_TIME_) log(_TIME_)**2, ftmap=2 2,
 tdenames=n_logt n_logt2);
```

which specifies the Cox model

$$h_i(t) = h_0(t)\exp(\beta_1 T_i + (\beta_{20} + \beta_{21}\log(t) + \beta_{22}\log(t)^2)N_i + \beta_3 G_i + \beta_4 CD_i)$$

The flexibility of these options could be used to specify cubic splines, which are easily obtained using the `%RCS` macro (Heinzl & Kaider, 1997) offered at our website.

4.8 Survivor function estimate

Suppose we want to produce survival function estimates for four groups of patient defined by the two dichotomous variables G and N from the model $h_i(t) = h_0(t)\exp(\beta_1 G_i + \beta_2 N_i)$. Submit the following code to obtain a plot similar to Fig. 2:

```
data sur;
set bsp.breast;
GN="G=0, N=0";
output;
GN="G=1, N=0";
G=G-1;
output;
GN="G=0, N=1";
G=G+1;
N=N-1;
output;
GN="G=1, N=1";
G=G-1;
output;
run;
proc sort;
by GN;
```

```
run;
%fc06(data=sur, time=time, cens=cens, varlist=G,
outs0=survivor, by=GN, print=0, pl=0);

axis1 label=(angle=90) order=(0 to 1 by 0.2);
proc gplot data=survivor;
plot _surv_*_time_=G_group/vaxis=axis1;
run;
```

4.9 Time-varying covariates

Consider a subject experiencing an event at 100 time units, and a time-varying covariate $x(t)$ that changes from 0 to 1 at 20 time units and from 1 to 0 at 70 units for that subject. The data of that subject has the following structure:

```
t1  t2  cens    x
0   20  0       0
20  70  0       1
70  100 1       0
```

To estimate the model $h_i(t) = h_0(t)\exp(\beta x_i(t))$ the following macro options are specified:

```
time1=t1,
time2=t2,
cens=cens,
censval=0,
varlist=x,
```

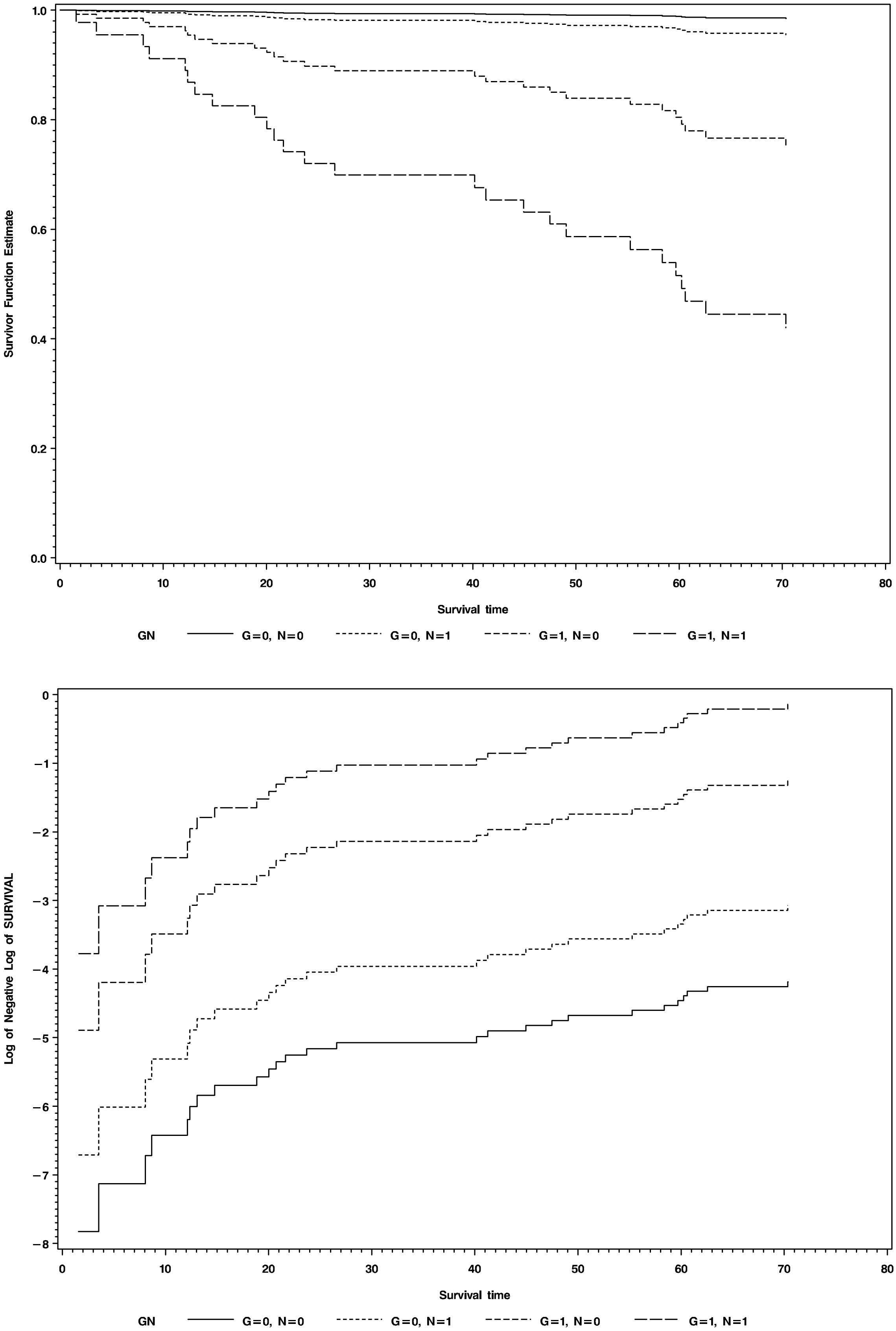
4.10 Recurrent events

Consider a subject experiencing events at 20 and 100 time units, and assume that after an event, the subject is not at risk for 30 time units. Let the follow-up period end immediately after the second event. The only covariate considered here is the time-invariant variable x, assuming the value 0 for that subject. The data of that subject has the structure:

```
t1  t2  cens    x
0   20  1       0
50  100 1       0
```

The model is estimated by specifying

Figure 2: Survivor function estimates (top) and log-minus-log survival (bottom) for the four groups defined by G and N.



```
time1=t1,
time2=t2,
cens=cens,
censval=0,
varlist=x,
```

Note that unlike PROC PHREG, no variance correction (e. g., the sandwich estimate) has been implemented in the macro. Therefore, it is suggested to use the jackknife or the bootstrap (resampling subjects) using the `by` option to estimate confidence intervals for parameters if subjects may experience recurrent events.

4.11 Stratification

Although the macro was not intended to allow for stratification, a stratified analysis can be achieved by making use of the counting process formulation. Assume that the largest survival time is 999. Let s denote the variable containing the stratum of each individual; $s = 1, \ldots, S$. Further, let x, t and c denote a covariate, the survival time and the censoring indicator, respectively. Then the following statements stratify the analysis by the levels of s:

```
data one;
set one;
t1=s*1000;
t2=t+s*1000;
run;
%fc06(data=one, time1=t1, time2=t2, cens=c, varlist=x);
```

4.12 Conditional logistic regression for designs with $1 : m$ matching

The macro can be (ab)used to apply Firth's penalization to conditional logistic regression models for analysis of matched case-control studies in which each case is matched to several controls. (The more general case of $n : m$ matching cannot be solved by the macro adequately because only with $1 : m$ matching the ties handling method is irrelevant.)

Suppose that the data structure is like the following, with `x1` and `x2` denoting two risk factors, `s` denoting the stratum number, and `c` denoting the case status (1=case, 0=control):

```
data one;
input x1 x2 s c;
```

```
cards;
1 1 1 1
1 0 1 0
0 0 1 0
0 1 2 1
0 1 2 0
0 0 2 0
...
;
run;
```

Then penalized maximum likelihood conditional logistic regression can be performed by the following data step and macro call:

```
data one1;
set one;
t1=1000*s;
t2=1000*s+1;
run;
%fc06(data=one1, time1=t1, time2=t2, cens=c, censval=0, varlist=x1 x2);
```

5 Availability

The macro `%FC06` described in the present manuscript is available at the www site

`www.muw.ac.at/msi/biometrie/programme/fc`

References

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* **80**, 27–38.

Heinze, G. & Ploner, M. (2002). SAS and SPLUS programs to perform Cox regression without convergence problems. *Computer methods and programs in Biomedicine* **67**, 217–223.

Heinze, G. & Schemper, M. (2001). A solution to the problem of monotone likelihood in Cox regression. *Biometrics* **57**, 114–119.

Heinzl, H. & Kaider, A. (1997). Gaining more flexibility in Cox proportional hazards regression models with cubic spline functions. *Computer Methods and Programs in Biomedicine* **54**, 201–208.

Loesch, A., Tempfer, C., Kohlberger, P., Joura, E. A., Denk, M., Zajic, B., Breitenecker, G., & Kainz, C. (1998). Prognostic value of cathepsin D expression and association with histomorphological subtypes in breast cancer. *British Journal of Cancer* **78**, 205–209.

Venzon, D. J. & Moolgavkar, S. H. (1988). A method for computing profile-likelihood based confidence intervals. *Applied Statistics* **37**, 87–94.