

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 2/2006

WCM: A SAS® macro for weighted parameter estimation for Cox's model

Georg HEINZE

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

Abstract

A new SAS macro program WCM was written to facilitate the application of weighted parameter estimation for Cox's model (cf. M. Schemper, 1992, Cox analysis of survival data with non-proportional hazard functions. *The Statistician* **41**, 455–465). The present report contains the complete User's Guide to this macro program including syntax, computational methods and examples.

Contents

1	Overview	4
2	Installation	4
2.1	What's inside the ZIP file?	4
2.2	Step-by-step installation guide	4
3	Working with the macro	5
3.1	Syntax	5
3.2	Basic options	6
3.3	Counting process style of input	7
3.4	Time-dependent effects	7
3.5	Output options	8
3.6	Model fitting options	9
3.7	Options useful for simulation	9
3.8	Titles	10
3.9	Printed output	10
3.10	Computational issues	11
4	Examples	11
4.1	Introduction	11
4.2	Time-dependent effect	15
4.3	Simultaneous test of parameters	16
4.4	Time-varying covariates	16
4.5	Recurrent events	17
4.6	Stratification	17
5	Availability and Disclaimer	18

1 Overview

The SAS-macro %WCM was written to facilitate weighted parameter estimation for Cox regression analysis, following the paper by Schemper (1992). Supplied with a SAS data set as input, its output contains the results of a Cox regression analysis in SAS/PROC PHREG (http://support.sas.com/onlinedoc/913/getDoc/en/statug.hlp/phreg_index.htm) style, where the hazard ratio corresponding to variables exhibiting non-proportional hazards may be estimated using weighted score functions (Schemper, 1992). The macro can be used to estimate confidence intervals based on the inversion of the scores test or the likelihood ratio test. All output of the macro is stored in SAS data sets, similarly to the output options supplied by SAS/PROC PHREG and ODS (output delivery system). The SAS-macro alternatively accesses the dynamic link library `wcm.dll` or the console application `wcmexe.exe`. The functions of those programs can also be accessed from programs other than SAS. Multiple input data sets can be efficiently processed using so-called BY-group variables similarly to PROC PHREG. Finally, offset values can be used to fix parameters at particular values.

2 Installation

2.1 What's inside the ZIP file?

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

File name	Description
<code>wcm.sas</code>	SAS macro code
<code>wcm.dll</code>	dynamic link library containing core routines of the program
<code>wcmexe.exe</code>	executable console application
<code>wcm.def</code>	definition table needed by SAS to access <code>wcm.dll</code>
<code>dforrt.dll</code>	run-time dynamic link library needed by <code>wcm.dll</code>
<code>biofeedb.sas</code>	the biofeedback 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.

- (a) Create a new folder (e. g. `c:\MYFOLDER\DLLs`) where you save the files `wcm.def`, `wcm.dll` and `wcmexe.exe` included in the ZIP file.
 - (b) In file `wcm.sas` change the default value of the path option in the macro header to `path=%str(c:\MYFOLDER\DLLs\)` and save `wcm.sas`.
- Copy the file '`dforrt.dll`' into your `C:\WINDOWS\system32` folder (`C:\Winnt\system32` if using Windows NT/2000) if it's not already there.
- 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 be 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 %WCM (the brackets < and > denote options that need not to be specified):

```
%wcm(<data=SAS data set,>  
<time=variable,>  
<time1=variable,>  
<time2=variable,>  
<cens=variable,>  
<censval=value,>  
varlist=variables,  
<genvar=variables,>  
<gentype=string,>  
<ft=expression,>  
<ftmap=value(s),>  
<tdenames=string,>  
<cltype=value,>  
<risk=value,>  
<outmod=SAS data set,>  
<global=SAS data set,>  
<outest=SAS data set,>  
<outtab=SAS data set,>  
<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 the variable specified in the `cens` option assumes the value 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.
- `cltype=value` controls the desired type of confidence interval estimation. In principle, three types are available:
 1. `cltype=0` requests Wald confidence intervals
 2. `cltype=1` requests confidence intervals based on the inversion of the scores test (the default)
 3. `cltype=2` requests profile likelihood confidence intervals
- `genvar=variables` names a list of independent variables for which weighted (generalized) parameter estimation should be performed. All variables appearing in `genvar` must be contained in `varlist`.
- `gentype=string` specifies the type of weights that should be used. Let N , N_h and $\hat{S}(t_h)$ denote the number of individuals in the sample, the number of individuals at risk just before failure time t_h , and the Kaplan-Meier estimate of the survival function just before t_h . The following specifications for `gentype` are allowed (the vertical separator | means that both specifications are equivalent):

Specification	Weight for risk set R_h	Equivalent linear rank test scores
KM PRENTICE	$N\hat{S}(t_h)$	Prentice
N WILCOXON	N_h	Generalized Wilcoxon
SQRTKM	$\sqrt{N\hat{S}(t_h)}$	-
SQRTN TARONE	$\sqrt{N_h}$	Tarone-Ware
1	1	Log-rank

The scores test from the weighted Cox model involving a single binary covariate is equivalent to a linear rank test employing the scores shown in the third column of the above table.

The average magnitude of the weights affects the Wald and profile likelihood methods for computing confidence intervals and tests, but not the scores test. Using the raw weights shown in the above table would lead to an artificially increased sample size and to Wald and profile likelihood confidence intervals that would be too narrow. Therefore, the weights are normalized to an average value of 1 by dividing the raw weights by their sum. However, I noticed that despite normalization, in some samples the P -values from Wald and likelihood ratio test on one side and the scores test on the other side may differ substantially, therefore use of the scores method for estimating confidence intervals and testing hypotheses is recommended.

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. The notation $(a, b]$ means that the interval ranges from a to b , excluding a and including b .

- `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

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

- `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` as exemplified below.
- `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.

Currently, time-dependent effects cannot be defined for variables for which weighted estimation has been requested (macro option `genvar`).

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 of 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 log likelihood (**_penli0_**), the maximized log likelihood (**_penlik_**), the global 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 any 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 maximum likelihood (ML) 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 log likelihood at the ML 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 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, scores or profile likelihood) can be controlled using the `cltype` option. The significance level of the intervals can be set by the `alpha` option.

3.6 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 likelihood ratio test, `testtype=SCORES|S` (the default) 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 exchange temporary files with 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 `wcm.exe` (included in the ZIP file) 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 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.

3.7 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. To explain this options, consider the following example: if the parameter corresponding to the 4th variable in `varlist` should be prevented from estimation and be fixed at 0, then the offset data set should contain single line with a variable called `_off4_` having the value 0. If the `by`-option is used, then it must be contained in the `offset` data set and the `offset` data set should have as many observations as there are BY-groups in the input data set. If a variable contained in `varlist` is not defined in the `offset` data set (using the `_offj_` syntax explained above), its parameter value will be estimated in any BY-group. Please note that in the current version of the macro the

simultaneous use of the `offset` and `test` options will prevent the parameters specified in `test` to be estimated. Nevertheless, the results of the test performed are valid.

- `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).

3.8 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.

3.9 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 log likelihood
- the value of the null 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) for the global null hypothesis that all parameters are 0. If weighted parameter estimation was requested for some parameters, then the global likelihood ratio and Wald tests are marked by the warning ***** BIASED! ***** which should remind the user that these tests may be biased. The full impact of weighted parameter estimation on these two tests and the type of normalization best suited to make them approximately valid has not yet been studied. Finally, the fourth page includes a table containing variable names, 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 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.

3.10 Computational issues

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.

The program may reserve 500MByte of memory. This amount of memory should be available on most machines and it should be enough to solve most problems. A data set with one covariate and 15,000 observations needs about 4MByte. However, SAS does not allow a DLL called by the PROC IML/CALL MODULE statement to reserve that much memory. Therefore, the `call=DLL` option may produce a ‘Visual Fortran run-time error’ if the data set to be analysed is too large (about 2,000 observations/one covariate). Instead, the `call=EXE` can be used, it should work fine with large data sets.

Occasionally, I noticed numerical difficulties when estimating partial scores confidence intervals in models including time-dependent effects. This can be remedied by requesting Wald confidence intervals for those effects. A comparison of Wald P -values (given in the table labelled ‘ML estimates’) and partial scores P -values (obtained one-by-one by repeated macro calls employing the `test` option) can help to roughly quantify a potential over- or underestimation of Wald confidence intervals.

4 Examples

4.1 Introduction

Use of %WCM is exemplified using a biofeedback treatment data set (Fig. 1). In this study the effect of biofeedback treatment on time until treatment success (swallowing rehabilitation) was evaluated in 33 patients suffering from aspiration after head and neck surgery (Denk & Kaider, 1997). The outcome of interest is the time from start of treatment until the patient could return to a full oral intake diet. Patients were randomized into two groups: one group of patients received the conservative treatment including thermal stimulation with ice and exercises for the lips, tongue, laryngeal closure and elevation. The second group received videoendoscopic biofeedback treatment, i.e. the patients and therapists could visually control swallowing maneuvers on a monitor. Treatment was started as soon as the healing process after surgery was finished. The time elapsing from surgery to start of treatment was considered as an important covariable determining the treatment success. Assume that the data have been stored in SAS data set `biofeedb`. Kaplan-Meier-analysis revealed that the benefit of biofeedback treatment that is visible soon after treatment onset seems to vanish with ongoing treatment duration (Fig. 2).

To obtain a hazard ratio related to biofeedback treatment versus conservative treatment that is averaged over time, a weighted Cox model may be used. Four pages of output (cf. Fig. 3–4) are produced by submitting

```
%wcm(data=biofeedb, varlist=biofeedb loghealing, time=treatdur,  
      cens=success, genvar=biofeedb, gentype=PRENTICE);
```

Figure 1: SAS data step processing the biofeedback data set.

```
data biofeedb;
input PAT_NR SUCCESS TREATDUR BIOFEEDB HEALING LOGHEALING;
cards;
  1  1      25  1  17 4.08746
  2  1       5  2  20 4.32193
  3  0      53  1  81 6.33985
  4  0     307  2 135 7.07682
  5  0      30  1 730 9.51175
  6  1      89  1  15 3.90689
  7  1      21  2  10 3.32193
  8  0     441  1 139 7.11894
  9  1      85  1  15 3.90689
 10  1      58  1  27 4.75489
 11  1      18  1   9 3.16993
 12  0      27  2  14 3.80735
 13  1      24  1  13 3.70044
 14  1      13  2  15 3.90689
 15  1      14  2  14 3.80735
 16  1      20  2  49 5.61471
 17  1      33  2  17 4.08746
 18  1      25  1  16 4.00000
 19  1     368  1 147 7.19967
 20  1      15  2  14 3.80735
 21  1      17  2  11 3.45943
 22  0     253  2  31 4.95420
 23  1      14  2 626 9.29002
 24  0     333  2  22 4.45943
 25  1      23  2  26 4.70044
 26  1     151  2  19 4.24793
 27  1      32  1  11 3.45943
 28  1      84  1  14 3.80735
 29  0     130  2  20 4.32193
 30  1      22  2  20 4.32193
 31  1      11  2  10 3.32193
 32  1       9  1  23 4.52356
 33  1       7  2  16 4.00000
;
run;
```

Figure 2: Kaplan-Meier analysis of biofeedback study.

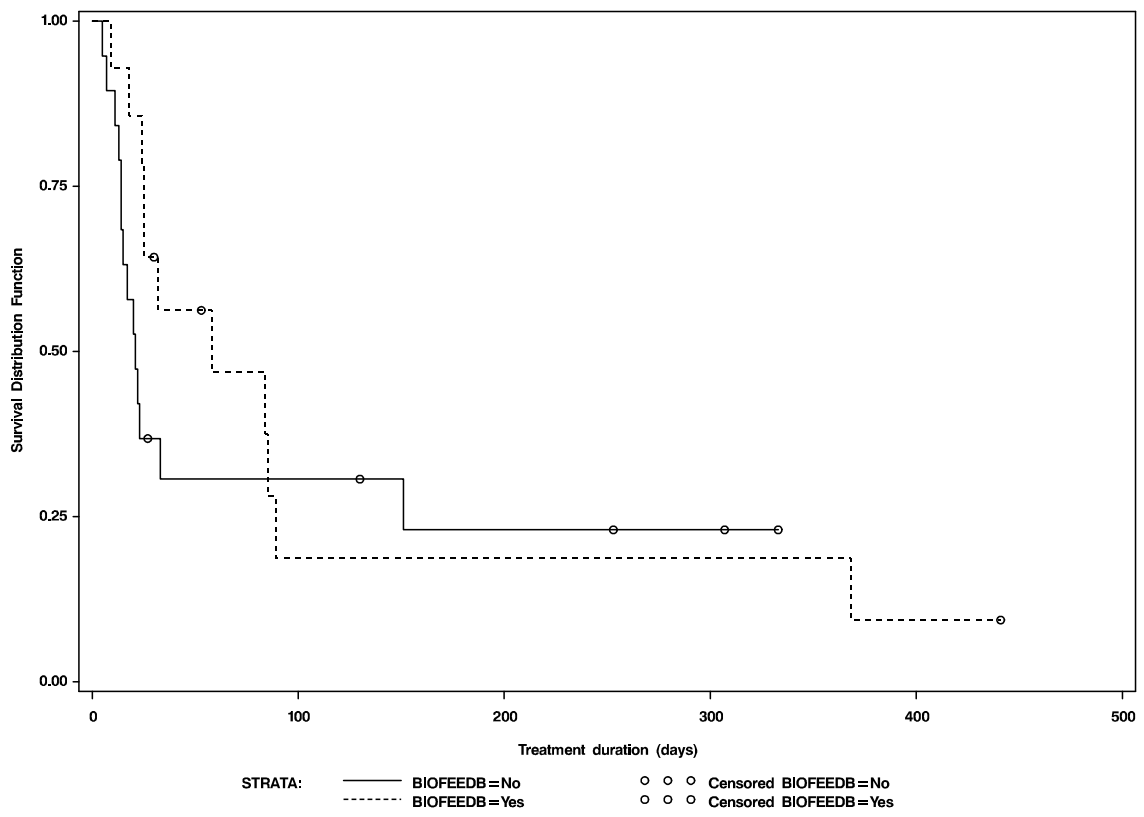


Figure 3: Page 1 of WCM output for the biofeedback study.

```
W  W  CCC  M  M  Weighted Cox Model
W  W  C    MM MM
W W W  C    M M M
W W W  C    M  M
  W W    CCC  M  M
```

Author: Georg Heinze

Version: 2006.01

Documentation: Heinze, G. (2006).

Technical Report 2/2006:

WCM: A program for weighted parameter estimation
for Cox's model

Section of Clinical Biometrics,

Core Unit for Medical Statistics and Informatics,
Medical University of Vienna.

Data set: BIOFEEDB

Dependent variable: TREATDUR

Censoring indicator: SUCCESS

Censoring value: 0

Ties handling: Breslow

Time-invariant effects: biofeedb loghealing

Table with parameter estimates saved in _TAB.

Estimates and covariance matrix saved in _EST.

Covariance matrix is based on inverse Fisher information.

Figure 4: Pages 2-4 of WCM output for the biofeedback study:

Model fitting information							
Iterations	Log likelihood	Null log likelihood	Number of events	Censored	Number of observations	% events	% censored
6	-67.2698	-71.1422	25	8	33	75.8	24.2

Testing global null hypothesis: beta=0

Test	Chi-Square	Degrees of freedom	Pr > Chi-Square	Remark
Likelihood Ratio	7.74485	2	0.0208	*** BIASED! ***
Scores	7.30948	2	0.0259	
Wald	6.17844	2	0.0455	*** BIASED! ***

ML estimates, partial scores confidence limits
and partial scores tests

Variable	Weighting	Parameter estimate	Standard error	Standardized estimate	Lower 95% c.l.	Upper 95% c.l.	Pr > Chi-Square
BIOFEEDB	KM	0.65925	0.47302	0.32582	-0.23962	1.55812	0.1506
LTHBEG	NONE	-0.58962	0.27155	-0.93657	-1.08763	-0.09162	0.0203

The remarks on the third page should remind the user that the likelihood ratio and Wald tests may be biased when weighted parameter estimation has been requested (cf. p. 7).

4.2 Time-dependent effect

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

```
%wcm(data=biofeedb, time=treatdur, cens=success, varlist=biofeedb loghealing,
ft=log(_TIME_), ftmap=1,
tdenames=BF_LT);
```

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

```
Time-invariant effects: biofeedb loghealing
Time-dependent effects:
BF_logt = biofeedb * log(_TIME_)
```

The output table suggests a significant interaction of biofeedback treatment with $\log(t)$:

ML estimates, partial scores confidence limits and partial scores tests							
Variable	Weighting	Parameter estimate	Standard error	Standardized estimate	Lower 95% c.l.	Upper 95% c.l.	Pr > Chi-Square
BIOFEEDB	NONE	4.83115	2.13335	2.38769	2.68750	6.97480	<.0001
LTHBEG	KM	-0.44465	0.25536	-0.70629	-0.90831	0.01902	0.0602
BF_LT	NONE	-1.41264	0.63423	-0.69816	-2.43562	-0.38965	0.0068

4.3 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 `biofeedb` has no effect on survival, the following macro call is submitted:

```
%wcm(data=biofeedb, time=treatdur, cens=success, varlist=biofeedb loghealing,
ft=log(_TIME_), ftmap=1,
tdenames=BF_logt, test=biofeedb BF_logt);
```

leading to the output page:

Partial scores test for parameters			
Tested parameters	Chi- Square	Degrees of freedom	Pr > Chi-Square
biofeedb BF_logt	6.30811	2	0.0427

4.4 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,
```

4.5 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. Assume that the follow-up period ends 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 the options

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

Note that unlike PROC PHREG, no variance correction (cf. Lin & Wei, 1989) has been implemented in the macro. Lipsitz & Parzen (1996) suggested to use the one-step jackknife variance estimate as an easy-to-use alternative. Assume that there are N individuals in the data set. The i th of N jackknife resamples is a copy of the original data set, leaving out the i th individual. The resamples can be efficiently analyzed using the `by` option. Option `outest` assigns a name to the output data set containing the variance estimates of all N resamples, which have to be combined to yield the jackknife variance estimate.

4.6 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; $s = 1, \dots, S$. Further, let x , t and c denote a covariate, the survival time and the censoring indicator, respectively. Then the following statements prepare the data set for stratified analysis (according to the levels of s):

```
data one;
set one;
t1=s*1000;
t2=t+s*1000;
run;
```

All individuals of stratum 0 enter the risk set at time 0 and have failed or have been censored before time 1000. At time 1000, the individuals of stratum 1 enter the risk set, etc. After preparing the data set, %WCM can be called:

```
%wcm(data=one, time1=t1, time2=t2, cens=c, varlist=x);
```

The ‘stratification trick’ cannot be used in combination with the `gentype=PRENTICE` (equivalent to `gentype=KM`) or `gentype=SQRTKM` options.

5 Availability and Disclaimer

Although the macro has been tested on various data sets, it must still be regarded as a ‘beta’ version that may suffer from numerical or other conceptual problems. The author appreciates any comments sent by e-mail (georg.heinze@meduniwien.ac.at) that may lead to an improved version of the macro. The author denies liability for results from the program.

%WCM is available at the WWW site <http://www.meduniwien.ac.at/msi/biometrie/programme/wcm>.

References

- DENK, D.-M. & KAIDER, A. (1997). Videoendoscopic biofeedback: A simple method to improve the efficacy of swallowing rehabilitation of patients after head and neck surgery. *Journal of Oto-Rhino-Laryngology and its related specialities (ORL)* **59**, 100–105.
- LIN, D. Y. & WEI, L. J. (1989). The robust inference for the proportional hazards model. *Journal of the American Statistical Association* **84**, 1074–1078.
- LIPSITZ, S. & PARZEN, M. (1996). A jackknife estimator of variance for Cox regression for correlated survival data. *Biometrics* **52**, 291–298.
- SCHEMPER, M. (1992). Cox analysis of survival data with non-proportional hazard functions. *The Statistician* **41**, 455–465.