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 3/2008 (updated 2009.09.24)

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

Georg HEINZE

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

Abstract

A SAS macro program WCM is introduced which facilitates the application of weighted parameter estimation for Cox's model (cf. M. Schemper, S. Wakounig and G. Heinze, 2009, 'The estimation of average hazard ratios by weighted Cox regression', *Statistics in Medicine* **28**, 2473–2489). The present report contains the complete User's Guide to this macro program including syntax, computational methods and examples.

Contents

1 Overview

The SAS–macro `%WCM` was written to facilitate weighted parameter estimation for Cox regression analysis, following the suggestions of Schemper (1992), Sasieni () and Schemper et al. (2009). Primarily, weighted parameter estimation can be used to obtain an estimate of the hazard ratio which is properly averaged over the population at risk at various time points and as such still valid if the proportional hazards assumption of Cox's model is in doubt. Alternatively, weighted estimation may provide a hazard ratio estimate which is more robust against outliers in both survival time and covariate value than the unweighted estimate.

Supplied with a SAS data set as input, the 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 three variance formulas: the sandwich estimate proposed by Lin () and Sasieni (), the robust sandwich estimate proposed by Lin & Wei (1989), or the jackknife variance estimate.

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.

The user may chose among Prentice (survivor function) weights, Breslow (number-at-risk) weights, or even supply alternative weights which are contained in the input data set. A special option allows for a correction of the weights in case of censored survival times. Weighting can be used for all variables in the model, or only for a subset thereof.

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
wcm.sas	SAS macro code
wcm08dll.dll	dynamic link library containing core routines of the program
wcm08exe.exe	executable console application
wcm08.def	definition table needed by SAS to access wcm.dll
biofeedb.sas	the biofeedback data set
wcm.pdf	this technical report

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 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`.

2. 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`.

3. 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*`,`
`<template=`*string*`,>`
`<genvar=`*variables*`,>`
`<gentype=`*string*`,>`
`<normalize=`*value*`,>`
`<id=`*variable*`,>`
`<userweights=`*variable*`,>`
`<censcorr=`*value*`,>`
`<transform=`*expression*`,>`
`<scale=`*value*`,>`

<ft=*expression*,>
<ftmap=*value(s)*,>
<tdenames=*string*,>
<robust=*value*,>
<jack=*value*,>
<risk=*value*,>
<outmod=*SAS data set*,>
<global=*SAS data set*,>
<outest=*SAS data set*,>
<outtab=*SAS data set*,>
<outtest=*SAS data set*,>
<outs0=*SAS data set*,>
<outw=*SAS data set*,>
<print=*value*,>
<test=*variables*,>
<call=*string*,>
<path=*string*,>
<maxit=*value*,>
<maxhs=*value*,>
<epsilon=*value*,>
<maxstep=*value*>
<by=*variables*,>
<notes=*value*,>
<debugmode=*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.

3.3 Weighting options: templates

- template=*string* calls a template which itself is a collection of presettings of other macro options. Several templates are supplied with the macro:

– `template=AHR` can be used for average hazard ratio estimation in case of non-proportional hazards, as proposed by Schemper et al. (2009). It assigns values to the following macro options: `genvar=&varlist`, `gentype=KM` is set to `KM`, `censcorr=1`, `robust=1`.

– `template=robust` performs a weighting by number of patients at risk. It assigns `genvar=&varlist`, `gentype=N`, `censcorr=0`, `jack=1`.

– `Atemplate=RE` uses only inverse probability of censoring weights. These weights are proportional to those proposed by Xu & O'Quigley (2000), and can be used to obtain an average regression effect (averaged over the censoring-corrected density of event times). The template assigns `gentype=1`, `genvar=&varlist`, `censcorr=1`, `robust=1`.

– `template=PH` performs a standard Cox analysis; it assigns `genvar=`, `censcorr=0`, `robust=1`, i. e., the robust sandwich estimate is employed for covariance estimation.

– `template=STRATA` performs a stratified standard Cox analysis. For this template, the user must define the strata variable(s) by the macro option `genvar`. Since the core routine of WCM does currently not directly account for stratification, this template will pass the data to PROC PHREG. Currently, time-dependent effects are not allowed with this template (they will be ignored).

3.4 Going deeper into weighting options

- `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

- `censcorr=`*value* can be either 0 for no correction for censoring, or 1 for a correction which multiplies the weights by the inverse of the Kaplan-Meier estimator of follow-up time $G(t_h)$; i. e., the Kaplan-Meier estimator with reverse meaning of the censoring status indicator.

- `normalize=`*value*: if set to 1, normalizes the weights such that their sum equals the number of distinct failure times

- `scale=`*value*: a factor by which the weights are multiplied with. Experimental; may help with numerical problems (if `normalize` is turned off).

- `userweights=`*variable*: a variable defining user-supplied weights. There should one unique weight value for each distinct failure time, otherwise the results are unpredictable. This option is experimental.

- `transform=`*expression*: an expression to be used to transform the weights. In this expression, untransformed weights are to be specified as `_w_`, e. g. `transform=sqrt(_w_)`.

3.5 Variance computation options

The macro offers three types of variance estimation, which are controlled by the macro options `robust` and `jack`:

- Chosing `robust=0` and `jack=0` calls the Lin-Sasieni sandwich estimate, which is independent of scaling of the weights. This variance estimate is theoretically valid only in case of proportional hazards and with no model misspecification. In case of no weighting, it is algebraically equal to the usual variance estimate based on inversion of the Fisher information matrix.
- `robust=1` and `jack=0` calls the Lin-Wei robust sandwich estimate, which is independent of scaling of the weights and robust against model misspecification. In simulations, we found out that it may be outperformed by the Lin-Sasieni variance in case of no or only mild model misspecification.
- `jack=1` or `robust=2` calls the Jackknife variance estimate which is valid also in case of misspecification and if weights are applied to some but not all variables in the model. It it the most time-consuming variance formula.

3.6 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`], with `time1` and `time2` signifying interval entry time and (possibly censored) survival time, 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.

3.7 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.

3.8 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 confidence level of the intervals can be set by the `alpha` option; the confidence level computes to 1-α.

- `outw=` specifies a SAS data set containing the raw weights (either $N(t_h)$ or $S(t_h)$), censoring correction values $(G(t_h)^{-1})$ and normalized total weights at each failure times.

3.9 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.

3.10 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` ist not defined in the `offset` data set (using the `_off`*j*`_` 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).
- `debugmode=`*value* If set to 1 and with `call=EXE`, writes the data set to a flat file, but then stops the macro. This options allows to user to check the data set that is actually being processed by the Fortran code.

3.11 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.

3.12 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.

3.13 Computational issues

The program rounds all survival and interval entry times to the nearest number which is a multiple of 10^{-6}. Therefore, it assumes that two survival times are equal if the absolute value of their difference is smaller than or equal to 10^{-6}. 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 $2 \cdot 10^{-6}$, otherwise the line is deleted by the program. Warnings are issued if

- any interval entry time is rounded
- any survival time is rounded
- any survival time ≤ 0; or any survival time $\leq$ interval entry time
- any interval entry time < 0

Option `call=DLL` causes SAS/PROC IML to dynamically invoke a dynamic link library. Sometimes numerical problems may arise with the DLL call, and some of these problems may be solved by using `call=EXE`. Using this option, SAS/PROC IML writes the current data set to a flat file and uses SAS's X command to call an external executable file, WCM08EXE.EXE, which performs all computations and writes back an output data file. The macro reads this output file back into SAS data sets. Apart from a black command window that pops up several times per processed data set and then closes again by itself, the user will not notice any difference between these two ways for estimation. While the DLL call is faster and more convenient, the EXE call may sometimes be safer.

The Fortran code used for both WCM08DLL.DLL and WCM08EXE.EXE was developed, compiled and built using Microsoft Visual Studio 2005 and Intel Visual Fortran Compiler 9.1. Some routines (mainly for matrix inversion) of the publicly available Fortran collection LAPACK were used and translated from Fortran 77 into Fortran 90.

4 Examples

4.1 A macro call using default settings

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 valid average hazard ratio related to biofeedback treatment versus conservative treatment, a weighted Cox model may be used. Four pages of output (cf. Fig. 3–4) are produced by submitting

```
%wcm(data=biofeedb, time=thdauer, cens=erfolg, varlist=biofeedb lthbeg);
```

By default, the macro calls the template `AHR` which uses KM/censoring correction weights for all variables that have been specified. A corresponding note appears in the SASLOG:

```
NOTE: Template AHR, the following options will be used:
      genvar=biofeedb lthbeg
      gentype=KM
      censcorr=1
      robust=1
      jack=0
      normalize=1
      transform=_w_
      risk=1

NOTE: This template allows for estimation of average hazard ratio
even in case of non-proportional hazards.
```

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

```
data biofeedb;
input PAT_NR_ ERFOLG THDAUER BIOFEEDB THBEG LTHBEG;
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.

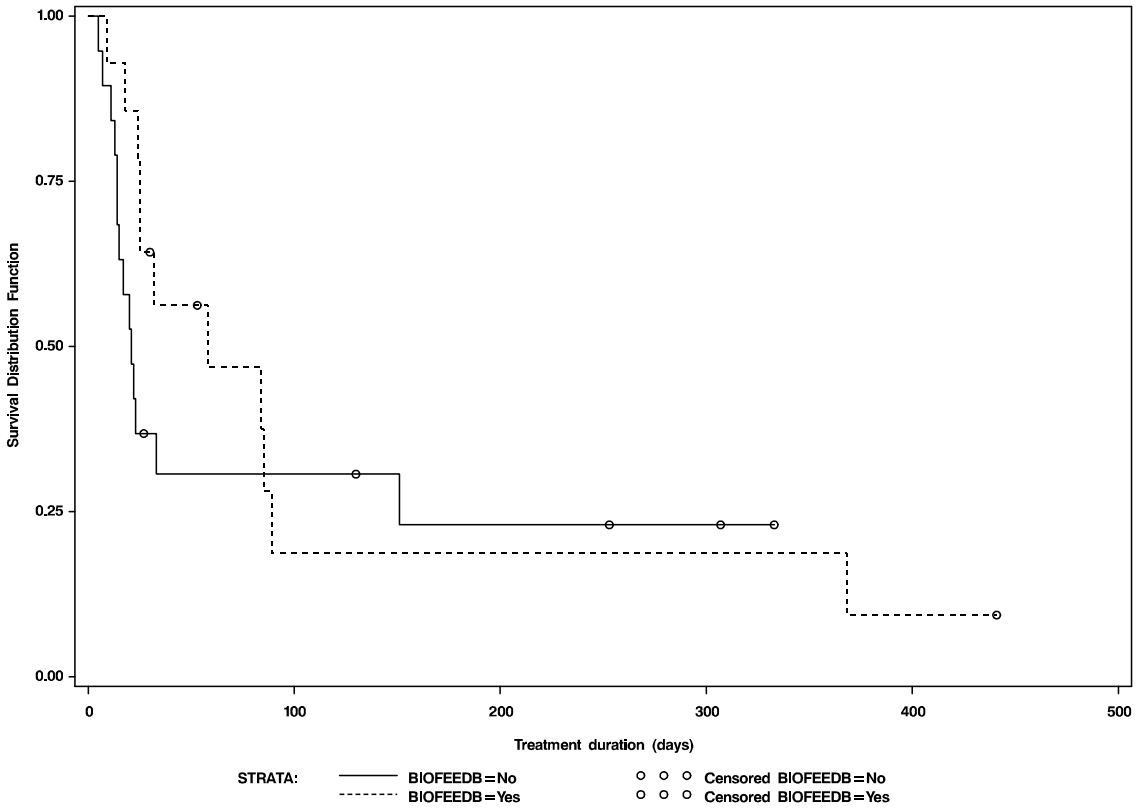

Figure 3: Page 1 of WCM output for the biofeedback study.

```
W   W    CCC   RRRR      Weighted Cox Regression
W   W   C      R   R
W W W   C      RRRR
W W W   C      R   R
 W W     CCC   R   R



Author:                  Georg Heinze
Version:                 2008.07

Documentation:           Heinze, G. (2008).
                         Technical Report 3/2008:
                         WCM - A SAS macro for weighted parameter estimation
                         in Cox's model
                         Section of Clinical Biometrics,
                         Core Unit for Medical Statistics and Informatics,
                         Medical University of Vienna.

Data set:                BIOFEEDB
Dependent variable:      THDAUER
Censoring indicator:     ERFOLG
Censoring value:         0
Ties handling:           Breslow
Time-invariant effects:  biofeedb lthbeg

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

Covariance matrix is based on the Lin-Wei robust sandwich method.
```

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	-69.9087	-73.4800	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	7.14266	2	0.0281	*** BIASED! ***
Scores	5.50703	2	0.0637	*** BIASED! ***
Wald (robust Lin-Wei)	4.13475	2	0.1265	

Weighted estimates, robust (Lin-Wei) Wald confidence limits and robust (Lin-Wei) Wald tests

Variable	Weighting	Parameter estimate	Standard error	Standardized estimate	Lower 95% c.l.	Upper 95% c.l.	Pr > Chi-Square
BIOFEEDB	S(t)/G(t)	0.55148	0.35743	0.27256	-0.14907	1.25203	0.1229
LTHBEG	S(t)/G(t)	-0.43463	0.35849	-0.69037	-1.13725	0.26800	0.2254

Weighted hazard ratio estimates, robust (Lin-Wei) Wald confidence limits and robust (Lin-Wei) Wald tests

Variable	Weighting	Hazard ratio	Lower 95% c.l.	Upper 95% c.l.	Pr > Chi-Square
BIOFEEDB	S(t)/G(t)	1.73583	0.86151	3.49745	0.1229
LTHBEG	S(t)/G(t)	0.64751	0.32070	1.30734	0.2254

The remarks on the third page should remind the user that the likelihood ratio and score tests may be biased when weighted parameter estimation has been requested.

4.2 The robust template

By specifying

```
%wcm(data=biofeedb, time=thdauer, cens=erfolg, varlist=biofeedb lthbeg, template=ROBUST);
```

an analysis is performed which applies weights proportional to the number of subjects at risk at each event time. Such an analysis may be useful for obtaining results robust against outliers, as the contributions to the likelihood are weighted according to the number of subjects they are based on. However, compared to the weights proposed by Schemper et al. (2009), the intuitive interpretability of the resulting hazard ratios is lost in case of non-proportional hazards.

4.3 Comparison to proportional hazards and stratified models

The new templates `PH` and `STRATA` allow a simple comparison of the weighted model with the proportional hazards model and the stratified proportional hazards model, without having to re-specify the model in a procedure step invoking PROC PHREG. For the standard proportional hazards model, specify

```
%wcm(data=biofeedb, time=thdauer, cens=erfolg, varlist=biofeedb lthbeg, template=PH);
```

For a proportional hazards model stratified by `biofeedb`, specify

```
%wcm(data=biofeedb, time=thdauer, cens=erfolg, varlist=biofeedb lthbeg, template=STRATA,
     genvar=biofeedb);
```

4.4 Average regression effect

Xu & O'Quigley (2000) proposed to estimate an average regression effect in case of non-proportional hazards. Their estimate is an average over the density of event times. This density is attenuated at later event times because of earlier censorship, but inverse probability of censoring weights can be applied to retain the uncensored density (up to the maximum follow-up time). To obtain average regression effects, specify

```
%wcm(data=biofeedb, time=thdauer, cens=erfolg, varlist=biofeedb lthbeg, template=ARE);
```

This template has been added for comparative purposes. We recommend the AHR template because of the numerical compatibility of the average hazard ratio obtained by weighted Cox regression and the odds of concordance, and the more intuitive population-based interpretation of the average hazard ratio compared to the average regression effect.

4.5 Comparison of AHR, PH, ARE, ROBUST and STRATA templates

The following table compares the results obtained by average hazard ratio estimation, average regression effect estimation, the standard model ignoring non-proportional effects and the stratified model. Note that the stratified model cannot supply an estimate for the hazard ratio of BIOFEEDB.

Template	Variable	Weighting	Hazard ratio	Lower 95% c.l.	Upper 95% c.l.	Pr > Chi-Square
AHR	BIOFEEDB	S(t)/G(t)	1.73583	0.86151	3.49745	0.1229
	LTHBEG	S(t)/G(t)	0.64751	0.32070	1.30734	0.2254
PH	BIOFEEDB		1.34154	0.69263	2.59839	0.3837
	LTHBEG		0.56516	0.27023	1.18199	0.1296
ARE	BIOFEEDB	1/G(t)	1.22589	0.63515	2.36608	0.5438
	LTHBEG	1/G(t)	0.54099	0.25909	1.12962	0.1019
ROBUST	BIOFEEDB	N(t)	1.99562	0.79404	5.01548	0.1417
	LTHBEG	N(t)	0.68269	0.42942	1.08535	0.1066
STRATA	LTHBEG		0.585	0.364	0.939	0.0264

4.6 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=thdauer, cens=erfolg, varlist=biofeedb lthbeg,
 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):

Weighted hazard ratio estimates, robust (Lin-Wei) Wald confidence limits and robust (Lin-Wei) Wald tests

Variable	Weighting	Hazard ratio	Lower 95% c.l.	Upper 95% c.l.	Pr > Chi-Square
BIOFEEDB	S(t)/G(t)	127.471	1.19454	13602.62	0.0419
LTHBEG	S(t)/G(t)	0.626	0.30206	1.30	0.2086
BF_LT	S(t)/G(t)	0.245	0.06493	0.93	0.0381

4.7 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:

test for parameters

Tested parameters	Chi-Square	Degrees of freedom	Pr > Chi-Square
biofeedb BF_logt	4.13947	2	0.1262

4.8 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,
```

4.9 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, id=patid,
robust=1
```

There are two options for providing a variance estimate that takes into account multiple events per subject: the robust sandwich estimate by Lin & Wei (1989) called by `robust=1` the jackknife estimate (`jack=1`). It is crucial that a variable defining patients is specified in the `id` option.

4.10 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 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 conceptional 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. Goodness-of-fit analysis for the Cox regression model based on a class of parameter estimators. *Journal of the American Statistical Association* **86**, 725–728.

Lin, D. Y. & Wei, L. J. (1989). The robust inference for the proportional hazards model. *Journal of the American Statistical Association* **84**, 1074–1078.

Sasieni, P. Maximum weighted partial likelihood estimators for the Cox model.

Schemper, M. (1992). Cox analysis of survival data with non-proportional hazard functions. *The Statistician* **41**, 455–465.

Schemper, M., Wakounig, S., & Heinze, G. (2009). The estimation of average hazard ratios by weighted Cox regression. *Statistics in Medicine* **28**, 2473–2489.

Xu, R. & O'Quigley, J. (2000). Estimating average regression effect under non-proportional hazards. *Biostatistics* **1**, 423–439.