Detecting bias arising from delayed recording of time: Appendix

Bianca L. De Stavola

Centre for Statistical Methodology, London School of Hygiene and Tropical Medicine, London, U.K.

E-mail: bianca.destavola@lshtm.ac.uk

D. R. Cox

Nuffield College, Oxford, U.K.

E-mail: david.cox@nuffield.oxford.ac.uk

Below we outline the Stata code including a simple Stata program called `origin_sens` used to produce Figure 4 in the main paper.

To produce the results for $\rho_{Z_0} = 1$ with 100 simulations for each combination of parameters, with the fitted model being a stratified semi-parametric proportional hazards model with one exposure and one stratifying variable, the relevant commands are:

```
********set seed for random number generation
set seed 3009

********prepare dataset with the relevant variables where the results
********of the simulations will be posted
clear all
set obs 100
qui gen lambda_Z=.
qui gen gamma_Z=.
qui gen beta_Z=.
qui gen cox_V=.
qui gen se_V=.
foreach n of numlist 1/10{
qui save results_‘n’, replace
}

**********read the original data
```

```
use cancer_test, clear

*******sensitivity analysis with Z as exponential with baseline mean 1

*set the counter for the file number where results are posted
local c=0

*run the simulation for values of beta_Z ranging from -010 to 0.30
*in steps of 0.05, and then add a final simulation with beta_Z=0.1229
*(the value estimated assuming exponential distributions for Z and V)
foreach z of numlist -0.10(0.05)0.30 0.1229{
  local c=‘c’+1
  di
  di in red "file number is  ‘c’  and BetaZ is  ‘z’"
origin_sens id time dead depbin yearcat, nsims(100) lambdaz(1) gammaz(1) ///
               betaz(‘z’) suffix(‘c’)
 }

*read the first set of simulated results and then append the other ones
use  results_1,clear
foreach c of numlist 1/10{
  append using results_‘c’
 }

*clean up the dataset
drop if lambda_Z>=.

*generate the lower and upper bounds for the estimated 95% CI
gen cox_lo=cox_V-1.96*se_V
gen cox_up=cox_V+1.96*se_V

*summarise the results as means specific to the combinations of
*(beta_Z, gamma_Z and lambda_Z); keep only the summary values
collapse  (mean) cox_V cox_lo cox_up (min) mincox_lo=cox_lo (max) maxcox_up=cox_up  ///
  ,by(beta_Z gamma_Z lambda_Z)
#delimit ;

*plot the results
twoway (scatter cox_V beta_Z)
```

```
        (rcap mincox_lo maxcox_up beta_Z)
        , ytitle(Estimated beta_V) yline(0.2598, lpat(dash))
          ylabel(0.20(0.01)0.29,angle(h))
          xtitle(beta_Z) legend(off)  xlabel(-0.10(0.10)0.30)
          xline(0.1229, lpat(dash));
#delimit cr
```

The `origin_sens` program invoked above has the following syntax:
`origin_sens <varlist>, nsims(.) lambdaz(.) gammaz(.) betaz(.) suffix(.)`
Note that it calls a user-written command called `survsim` which will need to be downloaded for the program to run (Crowther and Lambert, 2012).

`varlist` must include, in this order:

- the participant's identifier
- the follow-up time
- the outcome indicator
- the name of the exposure variable
- the name of the stratifying variable

The options are:

- `nsims(.)` states the number of simulations per combination of parameters
- `lambdaz(.)` states the selected value for ρ_{Z_0} (the option name was chosen to be consistent with the options used by `survsim`)
- `gammaz(.)` states the selected value for the shape parameter for Z: this is set to be 1 in order to generate an exponential distribution (the option name was chosen to be consistent with the options used by `survsim`)
- `betaz(.)` states the selected value for β_Z
- `suffix(.)` states the extension number to be given to the file holding the simulation results

The `origin_sens` program is listed below.

```
program define origin_sens,  rclass
version 14.1
syntax varlist(min=5 max=5),[ nsims(integer 100) lambdaz(real 1) gammaz(real 1) ///
                betaz(real 0) suffix(integer 1)]
```

```
tokenize ‘varlist’
local id ‘1’
local time ‘2’
local dead ‘3’
local X ‘4’
local yearcat ‘5’

 foreach sim of numlist 1/‘nsims’{
  di as text ‘sim’ ".." _cont
    cap drop Z V D_z
    qui survsim Z D_z, dist(weibull) lambdas(‘lambdaz’) gammas(‘gammaz’) ///
                   cov(‘X’ ‘betaz’) maxt(10000)
    qui gen V=time+Z
    qui stset V ,fail(‘dead’) id(‘id’)
qui stcox i.‘X’, strata(‘yearcat’)
 local cox_V=_b[1.‘X’]
 local se_V=_se[1.‘X’]
preserve
use results_‘suffix’, clear
qui replace cox_V=‘cox_V’ in ‘sim’
qui replace se_V=‘se_V’ in ‘sim’
qui replace lambda_Z=‘lambdaz’ in ‘sim’
qui replace gamma_Z=‘gammaz’ in ‘sim’
qui replace beta_Z=‘betaz’ in ‘sim’
qui save results_‘suffix’,replace
restore
}
end
```

Reference

Crowther, M. J. and Lambert, P. C. Simulating complex survival data (2012). Stata Journal 12 (4), 674–687.