```
/*******************************************************************************
* Malaria early morbidity and mortality                                        *
* Preliminary analysis  - generating tables and graphs - file last edited - 28 May 2015 *
* - first run do files '1a. Setup', and optionally '1b stvary diagnostics'     *
*******************************************************************************/

capture log close
version 13.1
set linesize 100
set more off
cd "C:\Users\Carl\Google Drive\MPH\Projects\Malaria project\Data\results"
loc today = c(current_date)
log using "malariaproject_log_`today'.txt", append text


*TIES
loc ties efron

*prepare folder for results
local T = c(current_time)
local T = subinstr("`T'",":","_",.)
mkdir "`ties' `today' `T'"
cd "`ties' `today' `T'"
mkdir figures
cd figures
mkdir PH
cd ..

keep obsno hrn Age Sex Ethnic AGR4 AdmNext14 DiedNext14 YearCat sexPreg hrnmal AdmFU14m
DiedFU14m whiteCat tgn SpeciesX AdmFU15 DiedFU15

*timer Start

timer clear 1
timer on 1

*** use "C:\Users\Carl\Google Drive\MPH\Internships\Malaria project\Data\File
archive\MalariaEpisodes_vs 1.9.3.dta", clear  /*FIRST RUN SET UP */

*Outcome / Failure variables
        loc outcome Adm Died        /* list of outcomes of interest to be analysed
        separately */
        loc Adm_f early admission           /* full title of 'admission' outcome for graph
        display */
        loc Died_f early death          /* full title of 'death' outcome for graph display */


*Exposures
loc commonExp   SpeciesX            ///
                AGR4                ///
                sexPreg             ///
                Ethnic              ///
                whiteCat            ///
                tgn

*Exposures
loc iExp        i.SpeciesX      ///
```

```
                b4.AGR4             ///
                i.sexPreg           ///
                i.Ethnic            ///
                i.whiteCat          ///
                i.tgn



*Model specifications
loc model1 ", cluster(hrn)"
loc model2 "i.SpeciesX b4.AGR4 i.sexPreg i.Ethnic i.whiteCat i.tgn, cluster(hrn) `ties'"

*save graph style in local macro
loc graph_style graphregion(fcolor(white) lcolor(white)) scheme(s2color)
di "`graph_style'"

    foreach outc of loc outcome {
*Initiate outcome timer
timer clear 2
timer on 2

*Copy template result sheet for each outcome
copy ..\Template_resTable_v2.xlsx `outc'_resTable.xlsx

*Set up for export of data to Excel results worksheet
putexcel set "`outc'_resTable.xlsx", modify keepcellformat
putexcel    A1  = ("Model 2: stcox `model2' ")      /* Title for excel results sheet */ ///
            C2 = ("`:di proper("``outc'_f'")'")

*Create macro references for excel export columns
loc nN_Cell         = "C"
loc uHRCell         = "D"
loc uHRpvalCell     = "E"
loc M2_HRCell       = "G"
loc M2_HRpvalCell   = "H"
loc mfp_HRCell      = "J"
loc mfp_HRpvalCell  = "K"



*Start cell for input values
loc vcell = 5
loc varcell = `vcell'
*Set up for survival analysis  (AdmFU14m and DiedFU14m are currently specified - recoding of
follow up time through 0.5 to 14.5)
    stset `outc'FU15, fail(`outc'Next14) id(obsno)
    loc axismax 15



* *Graph example risk set for outcome (need to fix outpoints and legend to be closed dot, not
arrow)



    loc Adm_hlstart = 72745
    loc Adm_hlend = 72752
```

```
loc Died_hlstart = 72745
loc Died_hlend = 72752
loc `outc'rs = ``outc'_hlstart'+1
loc `outc're = ``outc'_hlend'-1
loc hlimit = "if hrnmal>``outc'_hlstart' & hrnmal <``outc'_hlend'"
loc mcols = "black"
loc ts = "_t0"
loc te = "_t"
loc yvar = "hrnmal"
loc textv = ``outc'rs' -.2

twoway  sc  `yvar' `ts' `hlimit', mc(`mcols') ms(o)
||  ///
        pcspike `yvar' `ts' `yvar' `te' `hlimit' & _d==0, mc(`mcols') lcolor(`mcols')
        ||  ///
        pcspike `yvar' `ts' `yvar' `te' `hlimit' & _d==1, lcolor(`mcols')
        ||  ///
        sc `yvar' `te' `hlimit' & _d==1, mc(red)    ms(X)
        ||  ///
        sc  `yvar' `te' `hlimit' & _d==0, mc(`mcols') ms(o)
                                        ///
        text(72746.1 0 "(not malaria patient)", size(small) placement(e) color(gray))
              ///
        text(72748.1 0 "(not malaria patient)", size(small) placement(e) color(gray))
              ///
        text(72751.1 0 "(not malaria patient)", size(small) placement(e) color(gray))
              ///
            name("`outc'_egRiskset``outc'rs'to``outc're'",replace)
                                    ///
            title("HRN clusters (``outc'_f' riskset example)",size(medsmall)
                          ///
                placement(west) margin(-10 0 0 -3) justification(left))
                                   ///
            ylabel(72746.1(1)72751.1,            format(%9.0f)
                                        ///
                nogrid angle(horizontal) labsize(small))
                                                ///
            ytitle("Hospital Record Number clusters")
                                               ///
            xlabel(0(1)`axismax', labsize(small))    yscale(rev)
                                              ///
            xtitle("Time (days) from entry (_t0) until ``outc'_f' or censoring (_t)"
                    ///
                , margin(medsmall)) /*"*/
                                                                  ///
            xline(15, lpattern(shortdash) lc(edkblue) noextend)
                                        ///
            text(72746 15 "End of two weeks' follow up", size(small) placement(w))
                     ///
            legend(on order(1 "entry / exit (censored)" 2 "time at risk" 4 "``outc'_f'")
                ///
            colfirst notextfirst nostack cols(6) size(small) nobox
                                    ///
                region(fcolor(white) margin(zero) lcolor(white)) position(12) ring(1))
                     ///
            `graph_style' xscale(nofextend)
graph export figures/`outc'_egRiskset``outc'rs'to``outc're'.png, as(png) replace
```

```
***Loop code over explanatory variables for descriptive statistics

foreach v of varlist `commonExp' {

    *Export variable name
    putexcel    A`varcell'  = ("`: var label `v''")

    *Macros for key aspects (min max, n etc)
    su `v', meanonly
    loc vmax = r(max)
    loc vmin = r(min)
    loc vcat = (`vmax'-`vmin') +1
    if `vcat' <6 {

    /*alternate spacing for extra categories*/
            loc alt=""
        }
        else {
            loc alt="alt"
        }
    loc labname = `"`: val label `v''"' /*"*/
    qui: levelsof `v', loc(vl)

    *Macros for Kaplan-Meier curve and other graphs
    loc labname     = `"`: val label `v''"' /*"*/

    tempvar `v'_S   /* generating temporary survivor function variable by explanatory
    variable to establish scaling */
    sts gen ``v'_S' = s, by(`v')
    tempvar `v'_F
    gen ``v'_F' = 1 - ``v'_S'
    su ``v'_F', meanonly
    loc fmax = r(max)
    loc fmin = r(min)
    loc failmax     =round(trunc((r(max)*10))/10,.25)
    loc gap = round(`failmax'/5,.05)
    loc roundmax =`failmax'-`gap'
    loc mindif = r(max)-`roundmax'
    loc med = ""
    loc call = ""
    if `mindif' > .14 {
            loc med= `roundmax'+ .1
        }



    loc ordnum = 1
    foreach j of loc vl {       /*  establishing labels for value categories  */
        local call `call' `ordnum' "`: label `labname' `j''"
        loc ++ordnum
    }

    *KM survival curve (automatic y axis scaling)
```

```
sts graph, by(`v') failure
                                                                    ///
 name("`outc'_`v'_KM", replace)
                                                                ///
title("Probability of failure: ``outc'_f', by `: var label `v''", size(medsmall)
            ///
    placement(west) margin(-8 0 0 -3 ) justification(left))
                                          ///
xlab(0(1)`axismax', labsize(small)) xmtick(0(1)15)
                                                    ///
xtitle("Days since presentation with malaria", margin(medsmall))
                         ///
ylab(minmax `fmin' `fmax' 0(`gap')`roundmax' `med',
                                                          ///
    add format(%5.3f) nogrid labsize(small) angle(horizontal))
                                      ///
legend(on order(`call') colfirst notextfirst nostack cols(6) size(small)
                     ///
    nobox region(fcolor(white) margin(zero) lcolor(white)) position(12) ring(1))
                ///
`graph_style'  xscale(nofextend) yscale(nofextend)
                                             ///
note(" ")
graph export figures/`outc'_`v'_KM.png, as(png) replace

loc adjvar = subinstr("`commonExp'", "`v'", " ", 1)
loc adjcall
foreach av of varlist `adjvar' {
loc adjcall `adjcall' "`:var label `av'', "
}

loc adjcall = subinstr(`"`adjcall'"', char(34), "", . )            /*"*/
loc adjcall = substr("`adjcall'",1,length("`adjcall'")-2)

di "Adjusting for `adjcall'"


    sts graph, by(`v') failure   adjustfor("`adjvar'")                 /*"*/
    ///
     name("`outc'_`v'_KM_adj", replace)
                                                          ///
    title("Probability of failure: ``outc'_f', by `: var label `v'', adjusted*",
    size(medsmall) ///
        placement(west) margin(-8 0 0 -3 ) justification(left))
                                          ///
    xlab(0(1)`axismax', labsize(small)) xmtick(0(1)15)
                                                    ///
    xtitle("Days since presentation with malaria event", margin(medsmall))
                          ///
    ylab(minmax `fmin' `fmax' 0(`gap')`roundmax' `med',
                                                          ///
        add format(%5.3f) nogrid labsize(small) angle(horizontal))
                                      ///
    legend(on order(`call') colfirst notextfirst nostack cols(6) size(small)
                     ///
        nobox region(fcolor(white) margin(zero) lcolor(white)) position(12) ring(1))
                    ///
```

```
        `graph_style'  xscale(nofextend) yscale(nofextend)
                                                    ///
        note("*adjusted for:`adjcall'")
    graph export figures/`outc'_`v'_KM_adj.png, as(png) replace

    graph combine `outc'_`v'_KM `outc'_`v'_KM_adj, name("`outc'_`v'_KM_combo", replace)
    xsize(20) ysize(10.4) `graph_style'

    *Export cumulative incidence to Excel
    tab `v' `outc'Next14, row matcell(`v'_`outc'_tab)
    mata : st_matrix("`v'_`outc'_N", rowsum(st_matrix("`v'_`outc'_tab")))    /*sums columns
    for total N*/
    loc r = 1

    foreach i of loc vl {
        putexcel B`varcell' = (`"`: label `labname' `i'"'") /*"*/
        putexcel `nN_Cell'`varcell' = ("`:di %6.0fc `v'_`outc'_tab[`r',2]'/`:di %6.0fc (`v'_
        `outc'_N[`r',1])' (`:di%-5.2f ((`v'_`outc'_tab[`r',2]/(`v'_`outc'_N[`r',1]))*100)')")

        loc ++r     /*incremements the row number in stored matrix results */
        loc ++varcell     /*incremements the row number for output to Excel */
        }
    }

***Loop code over explanatory variables - Hazard ratios
    loc varcell = `vcell'
    foreach v in `iExp' {
        ***Univariable unadjusted model  (results for each outcome output to excel worksheets
        per variable in folder 'Results')
        stcox `v', `ties'
        matrix vHR = r(table)'
        local names: rownames vHR
        loc r = 1
        foreach n of loc names {
        loc vr `=substr("`n'",1,1)'   /*"*/
        loc br `=substr("`n'",2,1)'   /*"*/

        di "`vr'"
        di "`br'"

        if "`br'" == "b" {
                    putexcel            `uHRCell'`varcell' = ("1.00 (reference)")
                                                                        ///
                                        `uHRpvalCell'`varcell' = ("-")
                    }
        if "`br'" == "o" {
                    putexcel            `uHRCell'`varcell' = ("(omitted)")
                                                                    ///
                                        `uHRpvalCell'`varcell' = ("-")
                    }
        if "`br'" == "." {
                    putexcel            `uHRCell'`varcell' = ("`:di%3.2f vHR[`r',1]' (
                    `:di%3.2f vHR[`r',5]', `:di%3.2f vHR[`r',6]')")       ///
                                        `uHRpvalCell'`varcell' = ("`: di subinword("`: di
                                        %4.3f vHR[`r',4]'","0.000","< 0.001",1)'")
                    }
```

```
        loc ++r
        loc ++varcell
        di "r = `r'; varcell = `varcell'"
        }

}

     /* end of univariable loop */

***Plot  failure rate for Year with Era marker
loc `outc'_var YearCat                              /*define list of explanatory variables*/
foreach v of varlist ``outc'_var' {
    su `v', meanonly
    loc vmax = r(max)
    loc eramarker = ""
    set varabbrev off
    loc Adm_era_mark = 22.75
    loc Died_era_mark = 0.55

    strate `v', per(10000) graph cluster(hrn)
                                                        ///
        name("`outc'_YearEra_strate",replace)
                                                            ///
        title("Rate of ``outc'_f' per 10,000 patient-days, by Year & ACT Era",
                        ///
            size(medsmall) placement(west) margin(-10 0 0 -3) justification(left))
                        ///
        m(o) mc(black) ciopts(lc(black) ls(p2other))
                                                       ///
        xlabel(#`vmax',valuelabel labsize(small))
                                                        ///
        xtitle(`"`: var label `v''"', margin(medsmall)) /*"*/
                                        ///
        ylabel(, nogrid angle(horizontal) labsize(small)) ytitle("")
                                    ///
        xline(3.4, lpattern(shortdash) lc(blue) noextend)
                                                  ///
        text(``outc'_era_mark' 5.7 "ACT usage commences in April 2006", size(small)
                    ///
            justification(left))
                                                                        ///
        addplot(pcarrowi ``outc'_era_mark' 4 ``outc'_era_mark' 3.6 (3), mc(black)
                    ///
            msize(medsmall) mfc(black) lc(black))
                                                            ///
        legend(off)
                                                                              ///
        xscale(nofextend) yscale(nofextend) `graph_style'

    graph export figures/`outc'_Year-Era_strate10k.png, as(png) replace
    }

    timer off 2
    timer list 2
    di "Time to process data for `outc': " r(t1)/60 "minutes"
}  /* end of outcome loop */
```

```
timer off 1
timer list 1
di "Time to process complete do-file: " r(t1)/60 "minutes"
```