

# Stata analysis template files

19 May 2023 14:22

| File                    | Description  |
|-------------------------|--|
| profile.do              | In general no edits needed.  |
| master.do               | Control entire analysis and report generation.   |
| globals.do              | Specific setup for the analyses.   |
| newvariables.do         | Data edits for the analysis.   |
| header.org              | Basic setup, include title and author information, mandatory part of the report.   |
| intro.org               | Include data documentation and some description on the problem analysed, .   |
| flowchart.org           | Structured data exclusion process.   |
| table1.org              | Make basic baseline descriptions of the material.  |
| rates.org               | Outcome rates. Beware! this section writes data to rates.dta. It may be nescesarry to delete rates.dta if the analysis process has been altered fundamentally.                   |
| CumulativeIncidence.org | Survival or cumulative incidence curves.   |
| cox.org                 | Cox proportional hazards regression. Beware! this section writes data to cox.dta. It may be nescesarry to delete cox.dta if the analysis process has been altered fundamentally. |
| prepareForestplot.do    | Based on cox.dta, this file and makeForestplot.R is used to produce standard forestplots.  |
| plotMSD.org             | Based on data from table1 this file and callplotMSD.R is used to produce standard plots of standardized differences.   |
| cstat.org               | All you need for ROC curves  |

A suggested flow within master.do to generate one or several reports and place them in a subfolder in the output library along the analysis could be:

```
/* Part 1 */
dowe Header.org using "$LocalOutDir/${rename}Part1.org", replace
dowe Intro.org using "$LocalOutDir/${rename}Part1.org", append
dowe Flowchart.org using "$LocalOutDir/${rename}Part1.org", append
dowe Table1.org using "$LocalOutDir/${rename}Part1.org", append
/* Part 2 */
dowe Header.org using "$LocalOutDir/${rename}Part2.org", replace
dowe Rates.org using "$LocalOutDir/${rename}Part2.org", append
dowe CumulativeIncidence.org using "$LocalOutDir/${rename}Part2.org",
append
/* Generate pdf or html */
dowex, file("$LocalOutDir/${rename}Part1.org") type(html)
dowex, file("$LocalOutDir/${rename}Part2.org") type(pdf)
/* move to subfolders */
copyout , copypath("$LocalOutDir/Part1) inpath("$LocalOutDir") ///
        file(${rename}Part1.html) replace
copyout , copypath("$LocalOutDir/Part2) inpath("$LocalOutDir") ///
        file(${rename}Part1.pdf) replace
```

```

copyout , copypath("$LocalOutDir/Part2/graphics) inpath("$LocalOutDir")
///
        movetype(eps, png, pdf) replace

```

Alternatively to generate several subfiles and eventually join in a single report:

```

/* Part 1 */
dowe Header.org using "$LocalOutDir/${repname}Part1.org", replace
dowe Intro.org using "$LocalOutDir/${repname}Part2.org", replace
dowe Flowchart.org using "$LocalOutDir/${repname}Part3.org", replace
dowe Table1.org using "$LocalOutDir/${repname}Part4.org", replace
dowe Rates.org using "$LocalOutDir/${repname}Part5.org", replace
dowe CumulativeIncidence.org using "$LocalOutDir/${repname}Part6.org",
replace
/* join files */
cd $LocalOutDir
foreach i in 1/6{
    if `i'==1{
        cap erase ${repname}.org
        copy ${repname}Part1.org ${repname}.org
    }
    else shell type ${repname}Part`i'.org >> ${repname}.org
}
cd $LocalCodedir
/* Generate pdf or html */
dowex, file("$LocalOutDir/${repname}.org") type(pdf)

```

This approach appears in the template version.

# checklog

22. maj 2023 15:23

Scan logfile for stata error, currently all instances with „r(“

```
checklog , inpath(string) [filetype(string)]
```

| Argument | Explanation  | Defaults |
|----------|--|----------|
| inpath   | name of dir where files are located, \$LocalOutDir | .        |
| filetype | file extension                                     | log      |

Redigér  
checklog

# copyout

22. maj 2023 15:23

copy files with extension „movetype“ from „inpath“ to „copypath“

```
copyout , copypath(string) inpath(string) [movetype(string) file(string) replace]
```

| Argument | Explanation  | Defaults  |
|----------|--|-----------|
| copypath | name of new dir                                    |           |
| inpath   | name of dir where files are located, \$LocalOutDir |           |
| movetype | list of extensions like pdf png                    | all files |
| file     | filename   |           |
| replace  | if files needed to be replaced                     |           |

Redigér

Alternative

```
copyout, copypath("$localOutDir/$num")    inpath($LocalOutDir) replace
copyout, copypath("$localOutDir/$num/pdf") inpath($LocalOutDir) movetype(pdf)
replace
copyout, copypath("$localOutDir/$num/png") inpath($LocalOutDir) movetype(png)
replace
copyout, copypath("$localOutDir/$num")    inpath($LocalOutDir)
file(myreport.org) replace
```

# createList

22. maj 2023 15:23

Utility to create a global variable with a number of correctly formatted variable names and optional numbers for coding the table1 function

```
createList ''var'', num(string) list(string) addtxt(string)
```

| Argument | Explanation                                | Defaults |
|----------|--|----------|
| var      | name of new global variable                |          |
| num      | value to insert after each entry           |          |
| list     | list of variables or prefixes              |          |
| addtxt   | textstring to append to each entry in list |          |

Redigér

Alternative

```
gl ADJ warfarinbaseline statinsbaseline
createList ADJ , list(warfarin statins) addtxt("baseline")
gl Bvar warfarinbaseline 0 statinsbaseline 0
createList Bvar , num(0) list(warfarinbaseline statinsbaseline)
createList Bvar , num(0) list(warfarin statins) addtxt("baseline")
gl meds warfarin statins
createList ADJ,          list($meds) addtxt("baseline")
createList Bvar, num(0) list($meds) addtxt("baseline")
```

# beginhide/endhide

22. maj 2023 15:22

**beginhide** Insert begin\_comment in org file if report is censored and add comment that section is censored. Is ended by endhide. Can be used instead of \$beginhide, if \$CENSORREPORT (TRUE/FALSE) is used to flag censoring of the report.

beginhide [tag]

| Argument | Explanation                             | Defaults |
|----------|---|----------|
| tag      | type of current environment, eg example | -        |

## endhide

Insert end\_comment in org file if report is censored. Is started by beginhide. Can be used instead of \$endhide, if \$CENSORREPORT (TRUE/FALSE) is used to flag censoring of the report.

endhide [tag]

| Argument | Explanation                             | Defaults |
|----------|---|----------|
| tag      | type of current environment, eg example | -        |

# mkcspline

22. maj 2023 15:21

Generate natural cubic spline covariates centered around specific value.

```
mkcspline newvar = oldvar , nknots() center()
```

| Argument | Explanation     | Defaults |
|----------|-----------------|----------|
| newvar   | Stub for spline | -        |
| oldvar   | Covariate       | -        |
| nknots   | number of knots | -        |
| center   | reference level | -        |

Redigér

# table1

22. maj 2023 15:21

Safe generation of table 1. table1 function is by-enabled.

```
[bysort:] table1 covar type [covar type] [if] [fweight pweight aweight iweight],  
[by(string) test(string) balance(string) maxmsd all sep(string)  
fewdata(integer 4)  
saving(string) append varnames q(real) landscape size(string) missing]
```

| Argument  | Explanation  | Defaults   |
|-----------|--|------------|
| covar     | Covariate  | -          |
| type      | 0 (0/1), 1 (categorical), 2 (cont. mean/sd) 3 (cont. median/IQR) | -          |
| by        | By variable (columns)  | -          |
| test      | Add p-values for difference between by-variable                  | -          |
| balance   | Add standardized differences                                     | -          |
| maxmsd    | report max standardised across rows for categorical variables    | -          |
| all       | Add total column if by() is set                                  | -          |
| sep       | Seperator  | ' '        |
| fewdata   | Mask rows with risk of microdata                                 | -          |
| saving    | Dataset to save table in   | -          |
| append    | Append instead of replace saving dataset                         | -          |
| varnames  | Use names from datalist files insted of variablenames            | -          |
| q         | Specify quantiles for type 3                                     | .25        |
| landscape | Rotate table   | -          |
| size      | Specify LaTeX fontsize for table                                 | normalsize |
| missing   | Add missing category to type 1 (categorical) summary             | nomissing  |

I org filen kan disse to former anvendes (og er ikke specielt for table1).

Her får vi en fin formateret tabel, som desværre ikke umiddelbart lader sig kopiere fra PDF til WORD:

```
#+BEGIN_SRC stata  
table1 $BVAR , by($BGRP)  
#+END_SRC
```

Her får vi en uformateret men separeret tabel, som istedet lader sig kopiere fra PDF til WORD:

```
#+BEGIN_SRC stata :wrap example  
table1 $BVAR , by($BGRP)  
#+END_SRC
```

Alternativet er at danne rapporten i html istedet for pdf ved hjælp af type(html) option til dowe.

Tilladte size værdier:

| standard font size |      |      |      |
|--------------------|------|------|------|
| command            | 10pt | 11pt | 12pt |
| tiny               | 5pt  | 6pt  | 6pt  |
| scriptsize         | 7pt  | 8pt  | 8pt  |
| footnotesize       | 8pt  | 9pt  | 10pt |
| small              | 9pt  | 10pt | 11pt |
| normalsize         | 10pt | 11pt | 12pt |



|       |      |      |      |
|-------|------|------|------|
| large | 12pt | 12pt | 14pt |
| Large | 14pt | 14pt | 17pt |
| LARGE | 17pt | 17pt | 20pt |
| huge  | 20pt | 20pt | 25pt |
| Huge  | 25pt | 25pt | 25pt |

# stsplitPeriods

22. maj 2023 15:20

Rearrange data set to accommodate time-varying covariates. Our version of *stsplit*.

```
stsplitPeriods id startdate endpoint [endpoints] [if], using(string)
split(string)
               splitstart(varname) splitend(varname)
               [datestub(string) statusstub(string) saving(string asis)]
```

| Argument   | Explanation                                      | Defaults |
|------------|--|----------|
| id         | Identifier (fx pnr) in using                     | -        |
| startdate  | Date variable in current dataset                 | -        |
| endpoint   | Endpoints  | -        |
| using      | Dataset with periods                             | -        |
| split      | Indicator variable in using                      | -        |
| splitstart | Date variable in using                           | -        |
| splitend   | Date variable in using                           | -        |
| datestub   | String used as stub for date                     | EndDate  |
| statusstub | String used as stub for status                   | Status   |
| saving     | Save final data set instead of overwrite current | -        |

The using dataset may be the output dataset from %ReduceMediPeriods(); used in the makefile part of the SAS-datamanagement process. In this case *split* will be converted from a char to integer with labels.

As exemplified in the following *stsplitPeriods* may be applied succesively. It is recommended to apply stset before and after for an outcome to ensure all events and risktime is still in the splitted data.

```
* Make example basis dataset with 2 records and two outcomes end1 and end2
clear all
set obs 2
gen pid=_n
gen indexdate = 0
gen end1Status=1 if _n==1
gen end1EndDate=20 if _n==1
replace end1Status=0 if _n==2
replace end1EndDate=100 if _n==2
gen end2Status=0 if _n==1
gen end2EndDate=150 if _n==1
replace end2Status=1 if _n==2
replace end2EndDate=50 if _n==2
tostring pid, replace /* as person-id normally is a stringvariable */
save basis, replace
* Make an example period dataset as provided by the SAS
* macro %reducemediperiods
* in this case only one person has periods
clear all
set obs 2
gen pid="1"
gen treat=1
gen treatstart=10 if _n==1
gen treatend=40 if _n==1
replace treatstart=60 if _n==2
replace treatend=120 if _n==2
```

```

save treatperiods, replace
* Make example period dataset based in basis dataset
use basis, replace
gen maxend=max(end1EndDate,end2EndDate)
keep pid indexdate maxend
gen nperiods=ceil((maxend-index)/60)
expand nperiods
bysort pid: gen period=_n
gen periodstart=indexdate+(period-1)*60
gen periodend=59+(period-1)*60
replace periodend=min(periodend,maxend)
keep pid period periodstart periodend
save fixedperiods, replace
* Now we do some splitting
use basis, replace
* use stset to check that we do not miss observationtime and events
stset end1EndDate, failure(end1Status) origin(index) enter(index)
stsplittperiods pid indexdate end1 end2, using(treatperiods) ///
    split(treat) splitstart(treatstart) splitend(treatend)
* use stset to check that we do not miss observationtime and events,
*   check with previous output
stset end1EndDate, failure(end1Status) origin(index) enter(index) id(pid)
* note event though the periods only cover treatment then the treat
*   variable is coded zero for periods with no treatment
* note also that the stsplittperiods nicely copes with several endpoints
* now the magic comes, lets split the time once more with the fixed period dataset
stsplittperiods pid indexdate end1 end2, using(fixedperiods) ///
    split(period) splitstart(periodstart) splitend(periodend)
* use stset to check that we do not miss observationtime and events,
*   check with previous output
stset end1EndDate, failure(end1Status) origin(index) enter(index) id(pid)

```

# stsplitFixed

22. maj 2023 15:19

Rearrange data set to transform from continuous time to discrete time, and optionally include markers for simple one-way status shift.

```
stsplitFixed id startdate [endpoint [endpoints]] [if], splittime(numlist)
               [gentime(string) datestub(string) statusstub(string)
saving(string asis)]
```

| Argument   | Explanation                                      | Defaults  |
|------------|--|-----------|
| id         | Identifier (fx pnr) in using                     | -         |
| startdate  | Date variable in current dataset                 | -         |
| endpoint   | Endpoints  | -         |
| splittime  | Cutpoints (Days in follow-up)                    | -         |
| gentime    | Variable in output with periods                  | splittime |
| datestub   | String used as stub for date                     | EndDate   |
| statusstub | String used as stub for status                   | Status    |
| saving     | Save final data set instead of overwrite current | -         |

If called without endpoints it will generate a dataset with id, startdate, splittime/gentime for each cutpoint value. If endpoints are added, the status at end of period is added. If the endpoint was observed prior to period entry then endpoint status is missing.

Endpoints should not be taken literally as it may also be used to generate a dataset with time-varying covariates if the covariates are provided with date and status variables.

The function can also be used in combination with *stsplitPeriods* if also having time-varying covariates, such as treatment periods.

```
stsplitFixed pid indexdate, splittime(0 (30) 365) saving(mysplit.dta,
replace)
```

# storeGLM

22. maj 2023 15:16

Structured saving of regression analysis. Useful for later use fx in forest plot.  
storeGLM , using(string) id(string) saving(string) [append]

| Argument | Explanation                                | Defaults |
|----------|--|----------|
| using    | Dataset to read                            | -        |
| id       | Identifier for the current results         | -        |
| saving   | Dataset acting as database for all results | -        |
| append   | Append to existing                         | Noappend |

Redigér

# storeFG

22. maj 2023 15:16

Structured saving of subdistribution hazard rate ratio. Useful for later use fx in forest plot.  
storeFG , using(string) id(string) saving(string) [append]

| Argument | Explanation                                | Defaults |
|----------|--|----------|
| using    | Dataset to read                            | -        |
| id       | Identifier for the current results         | -        |
| saving   | Dataset acting as database for all results | -        |
| append   | Append to existing                         | Noappend |

Redigér

# storeHR

22. maj 2023 15:16

Structured saving of hazard rate ratio. Useful for later use fx in forest plot.  
storeHR , using(string) id(string) saving(string) [append]

| Argument | Explanation                                | Defaults |
|----------|--|----------|
| using    | Dataset to read                            | -        |
| id       | Identifier for the current results         | -        |
| saving   | Dataset acting as database for all results | -        |
| append   | Append to existing                         | Noappend |

Redigér

# storeRates

22. maj 2023 15:16

Structured saving of incidence rates. Useful for later use fx in forest plot.

```
storeRates , using(string) id(string) saving(string) [strata(varname) append]
```

| Argument | Explanation                                | Defaults |
|----------|--|----------|
| using    | Dataset to read                            | -        |
| id       | Identifier for the current results         | -        |
| strata   | Variable acting as exposure                | -        |
| saving   | Dataset acting as database for all results | -        |
| append   | Append to existing                         | Noappend |

Redigér



# plotCuminc

22. maj 2023 15:15

Generate nice plots (gph, png, pdf, eps) of the cumulative incidence function or Kaplan-Meier survival/failure function.

```
plotCuminc Cistub, ENDPoints(string)
[BY(string) Maxt(real 0) mint(0) lineopt(string) plotopt(string)
plotopt2(string)
title(string) sep(string) savingpath(string) name(string) orglegend(string)
survival ci headlev(string)
atrisk atrisktimes(string) atriskposx(real 0.2) atriskposy(real 0.1)
atriskopt(string) atriskcap(string) fewdata(5) quietly noflatline]
```

| Argument    | Explanation   | Defaults           |
|-------------|---|--------------------|
| Cistub      | Stub used in genCuminc  | -                  |
| Endpoints   | Stub(s) for endpoints to be plotted                                 | -                  |
| BY          | Variable to split tables  |                    |
| mint        | Start of followup in plot   | -                  |
| Maxt        | End of followup   | -                  |
| Lineopt     | Options to specify each line, separated by sep()                    |                    |
| plotopt     | Overall plot options  |                    |
| plotopt2    | Endpoint specific plot options, separated by sep()                  |                    |
| Title       | Endpoint specific titles, separated by sep()                        |                    |
| sep         | Character to use as separator, change if „,“ is to be used in title | ,                  |
| savingpath  | Place to store plotfiles  | .                  |
| name        | Prefix of plotfiles   |                    |
| orglegend   | text to use as legend in report                                     |                    |
| survival    | Plot survival instead of failure function                           |                    |
| ci          | Add confidence limits to plot                                       |                    |
| atrisk      | Add at-risk table to plot   |                    |
| atrisktimes | Times to report at-risk   | required if atrisk |
| atriskposx  | Horizontal adjustment of caption for at-risk                        |                    |
| atriskposy  | Vertical adjustment of at-risk table                                |                    |
| atriskopt   | Further options for display of at-risk table                        | size(medsmall)     |
| atriskcap   | Caption   | Numbers at risk    |
| headlev     | Heading level of generated plots in report                          | **                 |
| fewdata     | Suppress curve if events less than fewdata                          | 5                  |
| quietly     | Avoid sometimes annoying amount of graph windows popping up         | -                  |
| noflatline  | Suppress extension of curves to max1                                |                    |

Redigér

# reportData

21. august 2023 10:40

Generic function to list dataset content in the report (org-formattet)

```
ReportData varlist [if] , [using(string) by(string)  
                        format(string) sorting(string) ]
```

| Argument | Explanation  | Defaults |
|----------|--|----------|
| using    | Dataset name with variables in varlist   |          |
| by       | Variable to split tables   |          |
| format   | Numeric format, use syntax format(var1(format) var2(format))                       |          |
| sorting  | variables to sort after (allways a good idear to have by statement in the sorting) |          |

```
ReportData A B C if D==1 , using(data.dta) by(E)  
                        format(C(%4.1f) sorting(A B))
```

# reportHR

22. maj 2023 15:14

Generate table to report of hazard rate ratios, calculated by *genHR* and optionally e-values. The e-value quantifies the necessary size of effect of a single unobserved confounder to explain the observed exposure HR. (Vanderweels & Ding (2017) Annals of Internal Medicine)

```
reportHR [if] , using(string) [ by(string) eval evalue notrare format(string)
sorting(string)]
```

| Argument    | Explanation  | Defaults |
|-------------|--|----------|
| using       | Dataset name with hazard rates   | -        |
| BY          | Variable to split tables   | -        |
| eval evalue | Add e-values to report   | -        |
| notrare     | If the outcome is not rate (fx > 15%)  |          |
| format      | Numeric format   | %6.2f    |
| sorting     | variables to sort after (allways a good idear to have by statement in the sorting) |          |

Redigér

# reportGLM

22. maj 2023 15:14

Generate table to report of regression contrasts, calculated by *genGLM*.

```
reportGLM [if] , using(string) [ by(string) format(string) sorting(string)]
```

| Argument | Explanation  | Defaults |
|----------|--|----------|
| using    | Dataset name with hazard rates   | -        |
| BY       | Variable to split tables   | -        |
| format   | Numeric format   | %6.2f    |
| sorting  | variables to sort after (allways a good idear to have by statement in the sorting) |          |

Redigér

# reportRates

22. maj 2023 13:26

Generate table to report of incidence rates, calculated by *genRates*.

```
reportRates [if] , using(string) [BY(string) strata(string)  
format(string) sorting(strinn) fewdata(num)]
```

| Argument | Explanation  | Defaults  |
|----------|--|-----------|
| using    | Dataset name with rates  | -         |
| BY       | Variable to split tables   | -         |
| Strata   | Variable to define rows in table   | -         |
| format   | Numeric format   | %6.2f     |
| sorting  | variables to sort after (allways a good idear to have by statement in the sorting) |           |
| fewdata  | Threshold to suppress number of events   | \$fewdata |

Redigér

# reportCuminc

22. maj 2023 13:26

Generate table to report of cumulative incidence or Kaplan-Meier failure probabilities, calculated by *genCuminc*.

```
reportCuminc Clstub [if] , endpoints(string) time(numlist)
               [BY(string) survival format(string) sorting(string)
               saving(string) append fewdata(string) id(string)]
```

| Argument  | Explanation  | Defaults  |
|-----------|--|-----------|
| Clstub    | Stub of variables generated by <i>genCuminc</i>                                    | -         |
| ENDPoints | Stub(s) of endpoint(s) generated by <i>genEndpoints</i>                            | -         |
| BY        | By variable used in <i>genCuminc</i>   | -         |
| Time      | Followup time(s) to be reported  | -         |
| Survival  | Trigger survival estimates (if type=km)  | -         |
| format    | Numeric format   | %4.3f     |
| sorting   | variables to sort after (allways a good idear to have by statement in the sorting) | .         |
| saving    | save table in a dataset  |           |
| append    | append to dataset instead of replace   | noappend  |
| id        | Text string to identify results in saved dataset                                   |           |
| fewdata   | Mask results if number of events less than fewdata                                 | \$fewdata |

Redigér

```
reportCuminc CIm , endpoints(istroke death) time(1 5)
reportCuminc CI , endpoints(istroke death) by(treat) time(1 5)
```

# plotmsd

22. maj 2023 13:25

Generate plot of two sets of standardised differences, typically in relation to an IPTW analysis.

```
plotmsd varlist [if], using(file) rows(string) [rowlabels(string)  
labelopt(string)  
vref(real) legend(string) title(string)]
```

| Argument         | Explanation  | Defaults   |
|------------------|--|--|
| <i>varlist</i>   | 2 or 3 Variables with one or two variables to identify the quantity and one variable with the estimate | (required)   |
| <i>using</i>     | the second dataset, same structure as input dataset  | (required)   |
| <i>rows</i>      | list of entries from first variable in <i>varlist</i> to be plotted                                    | (required)   |
| <i>rowlabels</i> | Strings used in plot instead of rows entries, see example for adding spaces and special characters     |  |
| <i>labelopt</i>  | Plot instruction for strings   | angle(0) labsize(small)  |
| <i>vref</i>      | Vertical reference lines   | -0.1 0.1   |
| <i>legend</i>    | Legend instructions  | label(1 „Unweighted“) label(2 „Weighted“) position(5) ring(1) col(1) size(small) |
| <i>title</i>     | Main title   |  |

Redigér

```
* standard setup with data from table1 function
```

```
use table1, clear
```

```
plotmsd V1 V2 V18, using(table1w.dta) rows(sex age age65 chadsvasc)  
rowlabels(Female|gender Age Age|65+ CHA(sub:2)DS(sub:2)-VASC)
```

# forestplot

22. maj 2023 13:24

Generate forestplots, is actually an interface to an R script producing the plot (in pdf).

```

foresplot varlist [if], rowvar(varname) [groupvar(varname) columnvar(varname)
    valtab(string) plab(string) hadj(string) xlim(string) ppos(string)
rpos(real)
    vref(real) cutx(real) log(string)
    gcex(real) rcex(real) mcex(real)
    sublabelside(integer) hlines(string)
    addcol(varname) addcollabel(string)
    grouplabel(string) rowlabel(string) columnlabel(string)
columnsublabel(string)
    outdata(string) objpath(string) rcmd(string) plotname(string) sep(string)
    width(real) height(real) wpan(string) xseqn(string)
]

```

| Argument            | Explanation  | Defaults                              |
|---------------------|--|---------------------------------------|
| <i>varlist</i>      | 3 Variables with estimate, lower, and upper CI   | (required)                            |
| <i>rowvar</i>       | String variable for rows   | (required)                            |
| <i>groupvar</i>     | String variable for groups   |                                       |
| <i>columnvar</i>    | String variable for columns  | -                                     |
| <i>valtab</i>       | Display numbers left to plot (TRUE/FALSE)  | TRUE                                  |
| <i>plab</i>         | 4 strings to be used beneath each plot, separated by sep()                               | Favour Standard<br>Favour Alternative |
| <i>hadj</i>         | Horizontal adjustment of column headers, one number for each column                      | 0                                     |
| <i>xlim</i>         | Axis limits, lower and upper value pair for each column                                  | -                                     |
| <i>ppos</i>         | Horizontal position of text beneath each plot, left and right value pair for each column | -                                     |
| <i>rpos</i>         | Horizontal insert of row texts, suitable if groups are displayed                         | 0.1                                   |
| <i>vref</i>         | Position of vertical reference line  | 1                                     |
| <i>cutx</i>         | Criteria for cutting errorbars   | .1                                    |
| <i>log</i>          | Log scale ? TRUE/FALSE   | TRUE                                  |
| <i>gcex</i>         | Character size Group and coloumn text  | 1.5                                   |
| <i>rcex</i>         | Character size Row text  | 1.2                                   |
| <i>mcex</i>         | Character size other text  | 0.8                                   |
| <i>sublabelside</i> | Position of column sublabel above(3) or below(1)   | 3                                     |
| <i>hlines</i>       | Dashed lines between groups? TRUE/FALSE/row number list                                  | TRUE                                  |
| <i>addcol</i>       | Content of variable added right to entire plot fx interaction p-values                   |                                       |
| <i>addcollabel</i>  | Header of added column   |                                       |
| <i>grouplabel</i>   | Header of Group  |                                       |
| <i>rowlabel</i>     | Header of row  |                                       |
| <i>columnlabel</i>  | Header of column if columnvar is not specified   |                                       |



|                |   |             |
|----------------|---|-------------|
| columnsublabel | Subheader of column   |             |
| outdata        | Name of dataset exported to R   |             |
| objpath        | Path where R objects are stored   | ./          |
| outdata        | Name of dataset exported to R   |             |
| rcmd           | R command   | \$RPROGRAM  |
| plotname       | Name of produced foresplot  | foresplot   |
| sep            | Seperator used for special purpose in plab  |             |
| width          | Width of pdf (inches)   | 16.7        |
| height         | Height of pdf (inches)  | 6.5         |
| wpan           | Relative width of panels, separated by commas. 1st: Group/row text 2nd: value tables 3rd: graphics 4rd: optional added column | 45,20,30,10 |
| xseqn          | Positions of x-axis markers   | 0.5,1,2,5   |

Redigér

\* Example with several columns, and groups and rows

```
forestplot HR HRl HRu, rowvar(exposure) groupvar(subgroup) columnvar(endpoint)
  grouplabel("Population") rowlabel(Treatment)
```

\* Example with one column, and groups and rows

```
forestplot HR HRl HRu if subgroup=="main", rowvar(exposure) groupvar(endpoint)
  grouplabel("Endpoint") rowlabel(Treatment) plab(Favour old Favour new)
  xlim(0.2 5) ppos(0.5 2) columnsublabel(HR (95%CI))
  plotname($LocalOutDir/MyForestplot) hlines(FALSE) width(6) height(6)
```

# genEndpoints

19 May 2023 14:26

14:33

Generate variables *stubEndDate* and *stubStatus*, which are used in outcome analysis functions.

```
genEndpoint stub endpointdate(s) , deadDate(string) deadCode(string)
[combined] studyEndDate(string)
```

| Argument     | Explanation                               | Defaults |
|--------------|---|----------|
| stub         | Name of generated outcome                 | -        |
| varlist      | Endpoint date(s)/time(s)                  | -        |
| deadDate     | Death date/time                           | -        |
| deathCode    | Indicator for death                       | -        |
| studyEndDate | Study End date/time                       | -        |
| combined     | Trigger combined death - varlist endpoint |          |

Redigér

```
. genEndpoint stroke istrokefidaafidate sefidaafidate, deadDate(deathdate)
deadCode(death) studyEndDate(studyend)
. genEndpoint strokedead istrokefidaafidate sefidaafidate,
deadDate(deathdate) deadCode(death) studyEndDate(studyend) combined
. genEndpoint death, deadDate(deathdate) deadCode(death)
studyEndDate(studyend)
```

# GenRates

19 May 2023

Calculate incidence rates. Wrapper for strates, including stset. Output is generated using *reportRates*.

```
genRates [iweight aweight pweight fweight] [if], ENDPoints(string) at(numlist)
        Origin(string) Enter(string) Scale(string) Per(string)
        SAVing(string) [BY(string) id(string) append label(string)]
```

| Argument  | Explanation                                      | Defaults |
|-----------|--|----------|
| ENDPoints | Stub(s) of endpoint(s) generated by genEndpoints | -        |
| at        | Followup times to calculate rates at             | -        |
| Origin    | Start of timescale                               | -        |
| Enter     | Time at entry                                    | -        |
| Scale     | Scaling of timescale                             | -        |
| Per       | Rates expressed per                              | -        |
| SAVing    | Dataset to save rates                            | -        |
| BY        | Calculate for levels of variables                | -        |
| id        | If dataset is splitted, this identifies person   | -        |
| append    | Append to Rates dataset                          | noappend |
| label     | Label identifying current analysis               | -        |

```
genRates, endpoints(stroke death) at(1 5 99)
        origin(idate) enter(idate) scale(365.25) per(100)
        saving(rates.dta) label("Overall")
genRates, endpoints(stroke death) at(1 5 99)
        origin(idate) enter(idate) scale(365.25) per(100)
        saving(rates.dta) by(group) append
```

# genFG

22. maj 2023 13:24

Calculate Fine & Gray regression subdistribution hazard rate ratio estimates for exposure. Wrapper for `stcrreg`, including `stset`. Output is generated using *reportFG*.

```
genFG [iweight aweight pweight fweight], ENDPoints(string)
      compete(string) at(numlist)
      Origin(string) Enter(string) Scale(string) EXPosure(varlist)
      [SAving(string) label(string) id(string) crregopt(string)
      ADJust(string) ref(string) show append estore headlev(string)]
```

| Argument  | Explanation   | Defaults   |
|-----------|---|------------|
| ENDPoints | Stub(s) of endpoint(s) generated by <code>genEndpoints</code>                         | -          |
| compete   | Stub(s) of endpoint(s) generated by <code>genEndpoints</code> considered as competing | (required) |
| at        | Followup times to calculate hazard rate ratios at                                     | -          |
| Origin    | Start of timescale  | -          |
| Enter     | Time at entry   | -          |
| Scale     | Scaling of timescale  | -          |
| Exposure  | Exposure variable(s)  | -          |
| Saving    | Dataset to store current analysis   | -          |
| label     | Label identifying current analysis  | -          |
| id        | If dataset is splitted, this identifies person  | -          |
| crregopt  | Additional options to <code>stcrreg</code>  | -          |
| adjust    | Adjustment covariates   | -          |
| ref       | Reference category  | 1          |
| show      | Display full analysis output in report  | noshow     |
| append    | Append results to <code>Saving()</code>   | noappend   |
| estore    | Save estimation by estimates store  |            |
| headlev   | Header level in report of analysis (show only)  |            |

Redigér

```
genFG $weight, endpoints($ENDP) compete($ENDPd) at($FUP)
      origin($origin) enter($index) scale(365.25) exposure($BGRP)
      label("Crude") saving(tmpfg)
genFG $weight, endpoints($ENDP) compete($ENDPd) at($FUP)
      origin($origin) enter($index) scale(365.25) exposure($BGRP)
      exposure($BGRP) adjust($ADJ)
      label("Adjusted") saving(tmpfg) append
```

# genGLM

22. maj 2023 13:23

Calculate generalised linear regression estimates for exposure. Wrapper for glm or other estimation functions. Output is generated using *reportGLM*.

```
genGLM outcome [outcome [..]] [iweight aweight pweight fweight] [if],  
    EXPosure(varlist)  
    [Outcomestub(string) at(numlist)  
    SAVing(string) label(string)  
    glmopt(string) ADJust(string) ref(string) show append  
    estore elabel(string) posttest(string) headlev(string)  
    engine(string) byexposure]
```

| Argument    | Explanation  | Defaults     |
|-------------|--|--------------|
| Exposure    | Exposure variable(s)   | (required)   |
| outcomestub | Stub of outcome(s), prefix added to variable names   |              |
| at          | Possible enumeration of outcome variables, postfix added to variable names   | -            |
| Saving      | Dataset to store current analysis, only possible if exposure is one variable or byexposure is chosen   | -            |
| label       | Label identifying current analysis   | -            |
| glmopt      | Additional options to glm  | -            |
| adjust      | Adjustment covariates  | -            |
| ref         | Reference category   | 1            |
| show        | Display full analysis output in report   | noshow       |
| append      | Append results to Saving()   | noappend     |
| estore      | Save estimation by estimates store   |              |
| elabel      | Prefix label for estimates store, default is based in oucomestub, exposure and at enumeration  |              |
| posttest    | Postestimation commands, multiple commands separated by vertical sign,{e} is a reserved placeholder for Endpoint and followup text (see example) |              |
| headlev     | Header level in report of analysis (show only)   |              |
| engine      | Estimation function to use   | glm          |
| byexposure  | Request separate analyses for each exposure() entry  | nobyexposure |

Redigér

```
genGLM m n o , exp(sex) show  
genGLM $ENDP $weight, exposure($BGRP) outcomestub(risk) at($FUP)  
    label("Crude") saving(tmprisk)  
genGLM $ENDP $weight, exposure($BGRP) adjust($ADJ) outcomestub(risk) at($FUP)  
    label("Adjusted") saving(tmprisk) append posttest(margins $BGRP | predict  
pred{e}, mu)  
genGLM diseaseA diseaseB, exposure(covarA covarB covarC) byexposure label("logit")  
saving(tmprisk) engine(logist)
```

# genHR

22. maj 2023 09:14

Calculate Cox regression estimates for exposure. Wrapper for `stcox`, including `stset`. Output is generated using `reportHR`.

```
genHR [iweight aweight pweight fweight] [if], ENDPoints(string) at(numlist)
      Origin(string) Enter(string) Scale(string) EXPosure(varlist)
      [SAVing(string) label(string) id(string) coxopt(string) ADJust(string)
      ref(string) show append estore elabel posttest(string)
      headlev(string) assumption byexposure]
```

| Argument   | Explanation   | Defaults     |
|------------|---|--------------|
| ENDPoints  | Stub(s) of endpoint(s) generated by <code>genEndpoints</code>   | -            |
| at         | Followup times to calculate hazard rate ratios at   | -            |
| Origin     | Start of timescale  | -            |
| Enter      | Time at entry   | -            |
| Scale      | Scaling of timescale  | -            |
| Exposure   | Exposure variable   | -            |
| Saving     | Dataset to store current analysis   | -            |
| label      | Label identifying current analysis  | -            |
| id         | If dataset is splitted, this identifies person  | -            |
| Coxopt     | Additional options to <code>stcox</code>  | -            |
| adjust     | Adjustment covariates   | -            |
| ref        | Reference category  | 1            |
| show       | Display full analysis output in report  | noshow       |
| append     | Append results to <code>Saving()</code>   | noappend     |
| estore     | Save estimation by estimates store  |              |
| elabel     | Optional prefix for estore element, default is based in label   |              |
| posttest   | Postestimation commands, multiple commands separated by vertical sign, {e} is a reserved placeholder for Endpoint and followup text (see example) |              |
| headlev    | Header level in report of analysis (show only)  |              |
| assumption | Performs proportional hazard test and plots, do need „show“ to be enabled to see results in the PDF report  |              |
| byexposure | Request separate analyses for each <code>exposure()</code> entry  | nobyexposure |

Redigér

```
genHR , endp(istroke mi death) at(5 99) origin(afdate) enter(afdate) scale(365.25)
      exposure(aspirin)
genHR , endp(istroke mi death) at(5 99) origin(afdate) enter(afdate) scale(365.25)
      exposure(aspirin) adjust(age sex)
genHR , endp(istroke mi death) at(5 99) origin(afdate) enter(afdate) scale(365.25)
      exposure(i.aspirin age sex)
genHR , endp(istroke mi death) at(5 99) origin(afdate) enter(afdate) scale(365.25)
      exposure(i.aspirin age sex) posttest(margin | predict pred{e})
```

# genPV

22. maj 2023 09:14

Calculate pseudo value risk estimates. Wrapper for `stpcuminc`, including `stset`. Output is generated using *reportCuminc*. Generated data may be analysed by various regression models, see example in *Risks.org*.

Note only single-line data supported and no delayed entry.

```
genPV PVstub [iweight aweight pweight fweight], COMpete(string)
      ENDPoints(string) at(numlist)
      Origin(string) Enter(string) Scale(string) [strata(string)]
```

| Argument  | Explanation   | Defaults |
|-----------|---|----------|
| PVstub    | Stub for generated variables added to the current dataset     | -        |
| ENDPoints | Stub(s) of endpoint(s) generated by <code>genEndpoints</code> | -        |
| at        | Followup times to calculate risks at                          | -        |
| Origin    | Start of timescale  | -        |
| Enter     | Time at entry   | -        |
| Scale     | Scaling of timescale  | -        |
| strata    | Invokes stratified calculation                                | -        |

Redigér

```
genPV risk , compete($ENDPd) endpoints($ENDP) at($FUP) origin($origin)
enter($index) scale(365.25)
```

# genCuminc

22. maj 2023 09:12

Calculate cumulative incidence or Kaplan-Meier failure estimates. Wrapper for sts, Cox, stcompet, stcuminc including stset. Output is generated using *reportCuminc* and *plotCuminc*. If competing risks present use either stcompet or stcuminc, note that stcuminc is MUCH faster and give same result.

Observe genCuminc call with compete(stcuminc) does not allow left-truncated data (timesplitted) and does not provide confidenceintervals. Neither stcuminc nor stcompet provide confidence intervals if weights are used.

```
genCuminc Clistub [iweight aweight pweight fweight], ENDPoints(string)
          Origin(string) Enter(string) Scale(string)
          BY(string) type(string) [compete(string) id(string)]
```

| Argument  | Explanation  | Defaults |
|-----------|--|----------|
| Clistub   | Stub for generated variables added to the current dataset          | -        |
| ENDPoints | Stub(s) of endpoint(s) generated by genEndpoints                   | -        |
| Origin    | Start of timescale   | -        |
| Enter     | Time at entry  | -        |
| Scale     | Scaling of timescale   | -        |
| BY        | Calculate for levels of variables                                  | -        |
| type      | km, Cox, stcompet, stcuminc (if not km then specify compete())     | -        |
| compete   | Stub(s) of endpoint(s) generated by genEndpoints assumed competing | -        |
| id        | If dataset is splitted, this identifies person                     |          |

Redigér

```
genCuminc CI, endpoints(stroke)
          origin(idate) enter(idate) scale(365.25)
          by(group) type(stcompet) compete(death)
genCuminc CI , endpoints(death)
          origin(idate) enter(idate) scale(365.25)
          by(group) type(km)
```



# genFlowline

Friday, 19 May 2023 14.41

Create flowchart entry. All entries define exclusions.

```
genFlowline varname , text(string) CRITerion(expression) [new SAMEline]
```

| Argument | Explanation  | Defaults |
|----------|--|----------|
| varname  | Name of flowchart variable   | -        |
| text     | Description of exclusion   | -        |
| crit     | Criterion  | -        |
| new      | Start af new flowchart   | nonew    |
| same     | Add another criterion to previous entry, use this to include criteria with very few events | nosame   |

```
genFlowline flow , text("Male") crit(sex==2) new
genFlowline flow , text("To old ") crit(age>150)
genFlowline flow , text("To young") crit(age<0) same
genFlowline flow , text("Text criteria") crit(textvar== "dude")
```