Statistics/Data analysis

```
      name:  <unnamed>
       log:  d:\GitSoftware\mlad\Examples\mlad_fpm\mlad_example.smcl
  log type:  smcl
 opened on:  29 Nov 2023, 11:26:45

1 . // Load data
2 . use https://www.pclambert.net/data/rott3, clear
  (Rotterdam breast cancer data (augmented with cause of death))

3 .
4 . // Use stpm3 model to fit simple model in Stata
5 . // Variables
6 . //   -- hormon (factor variable)
7 . //   -- age (natural spline)
8 . stpm3  i.hormon @ns(age,df(3)), scale(lncumhazard) df(4)

  Iteration 0:  Log likelihood = -2886.9797
  Iteration 1:  Log likelihood = -2884.0565
  Iteration 2:  Log likelihood = -2884.0429
  Iteration 3:  Log likelihood = -2884.0429

                                                Number of obs =  2,982
                                                Wald chi2(4)  =  132.42
  Log likelihood = -2884.0429                   Prob > chi2   =  0.0000

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
xb           |
      hormon |
         yes |   .3731572   .0885452     4.21   0.000     .1996119    .5467026
  _ns_f1_age1|  -3.142327   .8938032    -3.52   0.000    -4.894149   -1.390505
  _ns_f1_age2|  -.9788548   .3796951    -2.58   0.010    -1.723043   -.2346661
  _ns_f1_age3|  -2.199723   .3953551    -5.56   0.000    -2.974604   -1.424841
-------------+----------------------------------------------------------------
time         |
         _ns1|  -25.52175   1.864219   -13.69   0.000    -29.17555   -21.86795
         _ns2|   8.048509   .9963344     8.08   0.000     6.095729    10.00129
         _ns3|  -1.016756   .0438515   -23.19   0.000    -1.102703   -.9308083
         _ns4|  -.6569238   .0476898   -13.77   0.000     -.750394   -.5634536
        _cons|   .8747299   .1904223     4.59   0.000     .5015089    1.247951
------------------------------------------------------------------------------
  Extended functions
   (1) @ns(age, df(3))

9 . estimates store stpm3

10.
11. // I will use the spline variables created by stpm3 when I use mlad
12.
13. // Here is a simple way to get initial values
14. // fit exponential model & use least squares
15. streg i.hormon _ns_f1_age1 _ns_f1_age2 _ns_f1_age3, dist(exp)

          Failure _d: osi==1
    Analysis time _t: os/12
  Exit on or before: time 10*12
        ID variable: pid

  Iteration 0:  Log likelihood = -3022.1979
  Iteration 1:  Log likelihood = -2977.5856
  Iteration 2:  Log likelihood =  -2967.277
  Iteration 3:  Log likelihood = -2967.2473
  Iteration 4:  Log likelihood = -2967.2473

  Exponential PH regression
```

No. of subjects = **2,982**
No. of failures = **1,171**
Time at risk = **20,002.4244**

Number of obs = **2,982**

LR chi2(**4**) = **109.90**
Log likelihood = **-2967.2473**
Prob > chi2 = **0.0000**

_t	Haz. ratio	Std. err.	z	P>|z|	[95% conf.	interval]
hormon						
yes	**1.424495**	**.1256015**	**4.01**	**0.000**	**1.198417**	**1.693221**
_ns_f1_age1	**.0509612**	**.0453617**	**-3.34**	**0.001**	**.0089036**	**.291684**
_ns_f1_age2	**.3769882**	**.1429753**	**-2.57**	**0.010**	**.1792684**	**.7927781**
_ns_f1_age3	**.1230702**	**.0483402**	**-5.33**	**0.000**	**.0569919**	**.2657618**
_cons	**.2219702**	**.0417761**	**-8.00**	**0.000**	**.1534947**	**.3209933**

Note: **_cons** estimates baseline hazard.

```
16. predict surv, surv

17. gen logcumH = log(-log(surv))

18. regress logcumH i.hormon _ns_f1_age1 _ns_f1_age2 _ns_f1_age3 _ns1 _ns2 _ns3 _ns4 if
> _d
```

Source	SS	df	MS		
Model	**639.715523**	**8**	**79.9644404**	Number of obs	= **1,171**
Residual	**2.9834e-11**	**1,162**	**2.5675e-14**	F(8, 1162)	> **99999.00**
				Prob > F	= **0.0000**
				R-squared	= **1.0000**
				Adj R-squared	= **1.0000**
Total	**639.715523**	**1,170**	**.546765404**	Root MSE	= **1.6e-07**

logcumH	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
hormon						
yes	**.3538171**	**1.41e-08**	**2.5e+07**	**0.000**	**.3538171**	**.3538172**
_ns_f1_age1	**-2.976691**	**1.40e-07**	**-2.1e+07**	**0.000**	**-2.976691**	**-2.97669**
_ns_f1_age2	**-.9755414**	**6.06e-08**	**-1.6e+07**	**0.000**	**-.9755415**	**-.9755413**
_ns_f1_age3	**-2.095001**	**6.14e-08**	**-3.4e+07**	**0.000**	**-2.095001**	**-2.095**
_ns1	**-13.18746**	**2.23e-07**	**-5.9e+07**	**0.000**	**-13.18746**	**-13.18746**
_ns2	**3.111902**	**1.39e-07**	**2.2e+07**	**0.000**	**3.111902**	**3.111902**
_ns3	**-.9738248**	**2.18e-08**	**-4.5e+07**	**0.000**	**-.9738249**	**-.9738248**
_ns4	**-.6637326**	**4.45e-08**	**-1.5e+07**	**0.000**	**-.6637327**	**-.6637325**
_cons	**.7966883**	**3.46e-08**	**2.3e+07**	**0.000**	**.7966882**	**.7966883**

```
19. // Store inital values
20. matrix b_init = e(b)

21.
22. // mlad is an alternative optimizer in Stata
23. // It calls Python and most calculations are performed within Python
24. // mlad requires a Python file to define the log-likelihood
25.
26. // Python likelihood file
27. // arguments for function are
28. //   -- beta - parameters
29. //   -- X    - data
30. //   -- wt   - weights (vector of 1's if no weights in mlad)
```

```
31. //   -- M    - dictionary containing othervars etc
32.
33. // The Python file is only a few lines as the log-likelhood is simple
34. type fpm_hazard_ll.py
  import jax.numpy as jnp
  import mladutil as mu

  def python_ll(beta,X,wt,M):
    xb   =  mu.linpred(beta,X,1)
    time =  mu.linpred(beta,X,2)
    eta = xb + time
    dtime = jnp.dot(M["dns"], beta[1])[:,None]
    return(jnp.sum(wt*(M["_d"]*(jnp.log(dtime) + eta) - jnp.exp(eta))))

35.
36. // Note that gradient (score) and Hessian functions are automatically obtained
37. // from the above when using mlad using automatic differentiation.
38.
39. // There is an option to include a Python setup file
40. // It is useful here as we need the derivatives of the log(time) spline variables
41. // These are needed within the log-likelihood function
42. // The setup file is called once before the iterations start.
43. type fpm_setup.py
  import jax.numpy as jnp
  from sfi import Macro
  def mlad_setup(M):
    dnsvars = Macro.getGlobal("dnsvars").split()
    dns = []
    for v in range(len(dnsvars)):
      dns.append(M[dnsvars[v]])

    dns.append(jnp.zeros((len(M[dnsvars[1]]),1)))
    dns = (jnp.array(dns).squeeze(axis=2)).T
    M["dns"] = dns
    return(M)


44.
45. // Fit model using mlad
46. // need to supply the two python files
47. // Setup two equations as stpm3
48. //   -- xb equation is for covariates effects
49. //   -- time equation is for effect of time
50. // Also the event indicator (_d) and derivatives (_dns1 _dns2 _dns3 _dns4) of
51. // the spline variables are needed
52. global dnsvars _dns1 _dns2 _dns3 _dns4

53. mlad (xb:   = i.hormon _ns_f1_age1 _ns_f1_age2 _ns_f1_age3, nocons ) ///
  >      (time: = _ns1 _ns2 _ns3 _ns4)                                  ///
  >      , othervars(_d _dns1 _dns2 _dns3 _dns4)                        ///
  >        pysetup(fpm_setup)                                           ///
  >        llfile(fpm_hazard_ll)                                        ///
  >        init(b_init) search(off)

  Iteration 0:  Log likelihood = -2967.2473
  Iteration 1:  Log likelihood = -2886.9559
  Iteration 2:  Log likelihood = -2884.0586
  Iteration 3:  Log likelihood = -2884.0429
  Iteration 4:  Log likelihood = -2884.0429
```

```
54. estimates store mlad

55. // display estimates
56. ml display
```

Number of obs = **2,982**
Wald chi2(**4**) = **132.42**
Log likelihood = **-2884.0429** Prob > chi2 = **0.0000**

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
hormon						
yes	**.3731572**	**.0885452**	**4.21**	**0.000**	**.1996119**	**.5467026**
_ns_f1_age1	**-3.142327**	**.8938032**	**-3.52**	**0.000**	**-4.894149**	**-1.390505**
_ns_f1_age2	**-.9788548**	**.3796951**	**-2.58**	**0.010**	**-1.723043**	**-.2346661**
_ns_f1_age3	**-2.199723**	**.3953551**	**-5.56**	**0.000**	**-2.974604**	**-1.424841**
time						
_ns1	**-25.52175**	**1.864219**	**-13.69**	**0.000**	**-29.17555**	**-21.86795**
_ns2	**8.048509**	**.9963344**	**8.08**	**0.000**	**6.095729**	**10.00129**
_ns3	**-1.016756**	**.0438515**	**-23.19**	**0.000**	**-1.102703**	**-.9308083**
_ns4	**-.6569238**	**.0476898**	**-13.77**	**0.000**	**-.750394**	**-.5634536**
_cons	**.8747299**	**.1904223**	**4.59**	**0.000**	**.5015089**	**1.247951**

```
57.
58. // compare estimates and standard errors
59. estimates table stpm3 mlad, se
```

Variable	stpm3	mlad
xb		
hormon		
yes	**.37315725**	**.37315725**
	.08854516	**.08854516**
_ns_f1_age1	**-3.1423274**	**-3.1423274**
	.89380321	**.89380321**
_ns_f1_age2	**-.97885475**	**-.97885475**
	.37969509	**.37969509**
_ns_f1_age3	**-2.1997227**	**-2.1997227**
	.39535509	**.39535509**
time		
_ns1	**-25.521752**	**-25.521752**
	1.864219	**1.864219**
_ns2	**8.0485086**	**8.0485087**
	.99633437	**.99633437**
_ns3	**-1.0167556**	**-1.0167556**
	.04385147	**.04385147**
_ns4	**-.65692379**	**-.65692379**
	.04768975	**.04768975**
_cons	**.87472985**	**.87472985**
	.19042234	**.19042234**

Legend: b/se

```
60.
61. // Note: leads to identical estimates
62. // mlad is faster in large datasets (depending on number of cores)
63. log close
      name:  <unnamed>
       log:  d:\GitSoftware\mlad\Examples\mlad_fpm\mlad_example.smcl
  log type:  smcl
 closed on:  29 Nov 2023, 11:26:47
```