

```

1
2 //Feel free to report errors & suggestions: ehsan@stat.ubc.ca
3 //Have future plans to modify it (and SAS/IML & R versions as well)
4 //See Young, et al. (2009) for a SAS macro version of it
5 //clear
6 //clear matrix
7 //clear mata
8
9
10 mata
11
12 function msm(newx, subjects, tpoints){
13 rseed(newx)
14
15     true_psi = 0.3
16     L_gamma0 = log(3/7)
17     L_gamma1 = 2
18     L_gamma2 = log(0.5)
19     L_gamma3 = log(1.5)
20     A_alpha0 = log(2/7)
21     A_alpha1 = 0.5
22     A_alpha2 = 0.5
23     A_alpha3 = log(4)
24     constant_point = 30
25     scale = 0.01
26     shape = 1
27
28     output = J(1, 19, .)
29     for (id=1; id<=subjects; id++)
30     {
31         u = uniform(1,1)
32         ran_exp = log(1-u)/(-1)
33         T0 = (ran_exp/(scale^shape))^(1/shape)
34         if (T0 < constant_point) {
35             IT0=1
36         }
37         else
38         {
39             IT0=0
40         }
41         max_T = tpoints
42         time_pointer1 = J(1,1,..)
43         time_pointer2 = J(1,1,..)
44
45         L = J(1,tpoints,0)
46         A = J(1,tpoints,0)
47         Y = J(1,tpoints,0)
48         Y_prev = J(1,tpoints,0)
49         pr_A = J(1,tpoints,0)
50         pr_L = J(1,tpoints,0)
51         int_factor = J(1,tpoints,0)
52         failure_time = J(1,tpoints,0)
53         Cense = J(1,tpoints,0)
54         Cens = J(1,tpoints,0)
55         pcens = J(1,tpoints,0)

```

```

56 pcense = J(1,tpoints,0)
57
58 j = 1
59 Y_prev[1]=0
60
61 logitL = L_gamma0 + L_gamma1 * IT0
62 pr_L[1] = 1/(1+(1/exp(logitL)))
63 if (pr_L[1]==1) {
64     L[1] = 1
65 }
66 else
67 {
68     successes = uniform(1,1) :&lt; pr_L[1]
69     L[1] = rowsum(successes)
70 }
71 logitA = A_alpha0 + A_alpha1 * L[1]
72 pr_A[1] = 1/(1+(1/exp(logitA)))
73 successes = uniform(1,1) :&lt; pr_A[1]
74 A[1] = rowsum(successes)
75
76 int_factor[1] = exp(true_psi*A[1])
77 if (T0 &gt; int_factor[1]) {
78     Y[1]=0
79     failure_time[1]=J(1,1,..)
80 }
81 else
82 {
83     Y[1]=1
84     failure_time[1] = T0*exp(-true_psi*A[1])
85 }
86 j = 1
87 Cense[1] = 0
88 Cens[1] = 0
89 pcens[1] = 0
90
91 for (j=2; j<=tpoints; j++){
92
93     if (time_pointer2 != J(1,1,..) ) {
94         if (j == time_pointer2) {
95             j = tpoints
96         }
97     }
98     if (time_pointer1 != J(1,1,..) ) {
99         if (j == time_pointer1) {
100             j = tpoints
101         }
102     }
103     Y_prev[j]=Y[j-1]
104
105     logitL = L_gamma0 + L_gamma1 * IT0 + L_gamma2 * A[j-1] + L_gamma3 * L[j-1]
106     pr_L[j]=exp(logitL)/(1+exp(logitL))
107
108     if (pr_L[j] == 1) {
109         L[j] = 1
110     }

```

```

111         else {
112             successes = uniform(1,1) :&lt; pr_L[j]
113             L[j] = rowsum(successes)
114         }
115
116     logitA = A_alpha0 + A_alpha1 * L[j] + A_alpha2 * L[j-1] + A_alpha3 * A[j-1]
117     pr_A[j]=exp(logitA)/(1+exp(logitA))
118     successes = uniform(1,1) :&lt; pr_A[j]
119     A[j] = rowsum(successes)
120
121     if (Y_prev[j]==1) {
122         A[j]=0
123         L[j]=0
124     }
125
126
127     int_factor[j] = int_factor[j-1] + exp(true_psi*A[j])
128
129     if (T0 &gt; int_factor[j]) {
130         Y[j]=0
131         failure_time[j]=J(1,1,..)
132     }
133     else
134     {
135         Y[j]=1
136         if (Y_prev[j]==1) {
137             failure_time[j]=failure_time[j-1]
138         }
139         else
140         {
141             failure_time[j]=(j-1)+((T0-int_factor[j-1])*exp(-true_psi*A[j]))
142         }
143     }
144
145
146     Cense[j] = Cense[1]
147     Cens[j] = 0
148     pcens[j] = 0
149     pcense[j] = pcense[1]
150 }
151
152 if (failure_time[tpoints] == J(1,1,..)) {
153     time_pointer1 = tpoints
154     time_pointer2 = tpoints
155 }
156 else
157 {
158     time_pointer1 = ceil(failure_time[tpoints])
159     time_pointer2 = ceil(failure_time[tpoints])
160 }
161
162 active_time = tpoints
163 if (time_pointer1 &lt; tpoints){
164     active_time = time_pointer1
165 }

```

```

166     if (time_pointer2 &lt; tpoints){
167         active_time = time_pointer2
168     }
169     if (time_pointer2 &lt; time_pointer1){
170         active_time = time_pointer2
171     }
172     else {
173         active_time = time_pointer1
174     }
175
176
177     idx = J(1,active_time,0)
178     Lx = J(1,active_time,0)
179     Ax = J(1,active_time,0)
180     Yx = J(1,active_time,0)
181     Y_prevx = J(1,active_time,0)
182     L_prev = J(1,active_time,0)
183     A_prev = J(1,active_time,0)
184     A_prevL = J(1,active_time,0)
185     pr_A_tx = J(1,active_time,0)
186     pr_Lx = J(1,active_time,0)
187     int_factorx = J(1,active_time,0)
188     Tx = J(1,active_time,0)
189     tpointx = J(1,active_time,0)
190     tpoint2x = J(1,active_time,0)
191     T0x = J(1,active_time,0)
192     IT0x = J(1,active_time,0)
193     max_Tx = J(1,active_time,0)
194     censor1 = J(1,active_time,0)
195     censor2 = J(1,active_time,0)
196     psi= J(1,active_time,0)
197     newxx= J(1,active_time,0)
198
199     for (tpoint=1; tpoint<=active_time; tpoint++)
200     {
201         tpoint2 = tpoint-1
202         T0x[tpoint]=T0
203         IT0x[tpoint]=IT0
204         tpointx[tpoint]=tpoint
205         tpoint2x[tpoint]=tpoint2
206         idx[tpoint]=id
207         Lx[tpoint]=L[tpoint]
208         int_factorx[tpoint]=int_factor[tpoint]
209         pr_Lx[tpoint] = pr_L[tpoint]
210         Ax[tpoint]=A[tpoint]
211         pr_A_tx[tpoint]=pr_A[tpoint]
212         if (tpoint==1) {
213             A_prev[tpoint]=0
214         }
215         else {
216             A_prev[tpoint]=A[tpoint-1]
217         }
218         if (tpoint==1) {
219             L_prev[tpoint]=0
220         }

```

```

221         else {
222             L_prev[tpoint]=L[tpoint-1]
223         }
224     A_prevL[tpoint] = A_prev[tpoint]*Lx[tpoint]
225
226     censor1[tpoint] = Cens[tpoint]
227     censor2[tpoint] = Cense[tpoint]
228
229     Y_prevx[tpoint] = Y_prev[tpoint]
230     psi[tpoint] = true_psi
231     newxx[tpoint] = newx
232
233     if (active_time < tpoints) {
234         if (tpoint < active_time) {
235             Yx[tpoint] = Y[tpoint]
236         }
237         else
238         {
239             Yx[tpoint] = 1
240         }
241     }
242     else
243     {
244         if (active_time == tpoints) {
245             Yx[tpoint] = Y[tpoint]
246         }
247     }
248
249     Tx[tpoint]=failure_time[max_T]
250     max_Tx[tpoint] = max_T
251 }
252
253
254
255     output = output\(idx', tpointx', tpoint2x', T0x', IT0x', int_factorx', Yx', Y_prevx', Ax', A_prev', Lx', L_prev',
A_prevL', pr_A_tx', Tx', max_Tx', pr_Lx', psi', newxx')
256 }
257
258
259 idx = rows(output)
260 outputx = output[2..idx,..]
261 st_matrix("outputx",outputx)
262 return(outputx)
263 }
264
265 end

```