F77 Program 24

```
c       Fortran 77 program to do a simulation of the Weibull distribution
c       with a variable rate parameter of the uncensored model.
c       This program performs Single regression.
c       This program may be adapted for censored models as well as for
c       uncensored models.
c       ***************************************************************
c               By Derek Dhammaloka FDX3 - 11th Mar. 1991
c       ***************************************************************
c               Define the following variables
c
c               cdf is the cumulative density function of the probability
c               function and is between 0 and 1. The function urand will
c               generate the random numbers between 0 and 1. It has 1
c               parameter iy, the seed to initialise the generator.
c               t is the remission time in arbitary units
c               b0old and b1old are the old values of b0 and b1 in the
c               iteration loop.
c               b0new and b1new are the new values of b0 and b1 to be
c               entered by the user, but is changed in the iteration
c               loop until they are equal to b0old and b1old.
c               b0diff and b1diff are the absolute differences between
c               the new and old values of b0 and b1.
c               sumln is the sum of the logs to the base e
c               powers has the form (t**k)
c               sr has the form ln(rho*t)*((rho*t)**k)
c               spexp has the form (t**k)*exp(.)
c               spexpic is as spexp but it also has z1(.)
c               kold is the old value of kappa in the iteration loop
c               knew is the new value of kappa to be entered by the
c               user, but is changed in the iteration loop until it is
c               almost equal to kold.
c               kdiff is the absolute diff. between the new and old
c               values of kappa
c               gptype gives a random number and uses it to classify
c               the type of group (control,treated) and this info. is
c               stored in the indicator variable z1.
c               treated is the no. of individuals in the treated group.
c               rho is the non-constant rate parameter
c               tol is the tolerance value
c               loop is used in loop counters
c               n is the no. of uncensored individuals to be entered by
c               the user.
c               iy and iy2 are the seeds to be entered by the user.
c       ***************************************************************
      real cdf(5000),t(5000),rho(5000)
      real sr,sumln,spexp,spexpic,powers,knew,kold,kdiff
      real tol,b0old,b0new,b1old,b1new,b0diff,b1diff,gptype
      integer loop,n,iy,iy2,z1(5000),treated
      tol=0.000005
c       ***************************************************************
c               Input the no. of individuals
c               Also the index (kappa) and the rate (rho)
c       ***************************************************************
      print*,'How many individuals'
```

```
      read*,n
      print*,'Seed for the Weibull distribution'
      read*,iy
      print*,'Enter the index parameter'
      read*,knew
      print*,'Enter b0'
      read*,b0new
      print*,'Enter b1'
      read*,b1new
      print*,'Enter seed for the group type'
      read*,iy2
c     ***************************************************************
c             Simulate the Weibull distribution
c             using the regression constants to obtain the rate
c             parameter. This rate parameter (non-constant) and the
c             index parameter will be used to obtain the remission
c             times.
c     ***************************************************************
      do 20 loop=1,n
             cdf(loop)=urand(iy)
c
c            Generate a random number for the group type.
c            Assume that it is likely for the individual to
c            be in the control or treated group.
c
c            If random number is between 0 and 0.5, the individual is
c            in the control group and has indicator variable=0. Else
c            it is in the treated group and has indicator variable=1.
c
             gptype=urand(iy2)
             if(gptype.ge.0.and.gptype.le.0.500) z1(loop)=0
             if(gptype.ge.0.500.and.gptype.le.1) z1(loop)=1
             rho(loop)=exp((b0new)+(b1new*z1(loop)))
             t(loop)=(-log(1-cdf(loop))/(rho(loop)**knew)**(1/knew)
20    continue
c     ***************************************************************
c             Print the headings
c     ***************************************************************
      print*
      print*,'Simulation of the Weibull distribution with '
      print*,'Index = ',knew
      print*,'b0    = ',b0new
      print*,'b1    = ',b1new
      print*
      write(*,25)
25    format(t3,'cdf',t10,'time',t40,'rate',t53,'treat?')
c     ***************************************************************
c             Output the cdf and remission time values
c             Also the rate parameter and indicator variable for
c             type of group (control,treated)
c     ***************************************************************
      do 30 loop=1,n
             write(*,40)cdf(loop),t(loop),rho(loop),z1(loop)
40           format(f7.3,t8,f7.3,t37,f7.3,t50,i7.0,t60)
30    continue
      print*
```

```
c               Update the statistics
      do 35 loop=1,n
               sumln=sumln+(log(t(loop))
               treated=treated+z1(loop)
35    continue
c               Print statistics
      print*,'Sum of logs to the base e =   ',sumln
      print*,'No. in treated group       =  ',treated
      print*
c               Print headings
      write(*,45)
45    format(t8,'kappa',t20,'b0',t30,'b1')
c               Iteration loop
50    kold=knew
      b0old=b0new
      b1old=b1new
c               Update the statistics
      do 60 loop=1,n
      sr=sr+((log(rho(loop))*(t(loop)))*((rho(loop)*t(loop))**kold))
60    continue
      knew=(sr/n)-(b0old)-(b1old*treated/n)-(sumln/n)
      knew=1/knew
      kdiff=abs(knew-kold)
c     Use the new value of kappa to obtain sumpowerexp
      do 68 loop=1,n
               powers=t(loop)**knew
               spexp=spexp+(exp(z1(loop)*knew*b1old)*powers)
68    continue
c
c     Use the new value of kappa to obtain b0
c
      b0new=(log(n/spexp))/knew
      b0diff=abs(b0new-b0old)
c
c     Use the new value of kappa to obtain sumpowerexpic
c
      do 71 loop=1,n
        powers=t(loop)**knew
        spexpic=spexpic+(exp(z1(loop)*knew*b1old)*powers*z1(loop))
71    continue
c
c     Use the new values of kappa and b0 to obtain b1
c
      b1new=log(exp(b1old)*treated)/(exp(knew*b0new)*spexpic)
      b1diff=abs(b1new-b1old)
      write(*,80)knew,b0new,b1new
80    format(f12.5,f12.5,f12.5)
c     ***************************************************************
c               If the old and new values of kappa,b0 and/or b1 do not
c               agree to a certain no. of decimal places, zero the sums
c               (not sumln,treated) and carry on with the iteration.
c     ***************************************************************
```

```
      if((kdiff).gt.tol.or.(b0diff).gt.tol.or.(b1diff).gt.tol) then
                                    spexp=0
                                    spexpic=0
                                    sr=0
                                    goto 50
      endif
      stop
      end

      real function urand(iy)
      integer iy
c     ************************************************************************
c              Urand is a uniform random number generator based on
c              theory and suggestions given by KNUTH (1969). The
c              integer iy should be initialised to an arbitary integer
c              prior to the first call to urand. The calling program
c              should not alter the value of iy between subsequent
c              calls to urand. Values of urand will be returned in the
c              interval (0,1).
c     ************************************************************************
c              Reference - Problem solving with Fortran 77
c                          Brian D.Hahn 1987
c     ************************************************************************
      integer ia,ic,itwo,m2,m,mic
      double precision halfm
      real s
      data m2/0/,itwo/2/

c     If first entry, compute machine integer word length
      if(m2.eq.0) then
                  m=1
10    if(m.gt.m2) then
                  m2=m
                  m=itwo*m2
                  goto 10
      endif
      halfm=m2

c     Compute multiplier and increment for linear congruential method
      ia=8*int(halfm*atan(1.d0)/8.d0)+5
      ic=2*int(halfm*(0.5d0-sqrt(3.d0)/6.d0))+1
      mic=(m2-ic)+m2

c     s is the scale factor for converting to floating point
      s=0.5/halfm
      endif

c     Compute next random number
      iy=iy*ia

c     The following statement is for computers which do not allow
c     integer overflow on addition
      if(iy.gt.mic) iy=(iy-m2)-m2
      iy=iy+ic
```

```
c       The following statement is for computers where the word length
c       is greater than for multiplication
        if(iy/2.gt.m2) iy=(iy-m2)-m2
c
c       The following statement is for computers where integer overflow
c       affects sign bit
        if(iy.lt.0) iy=(iy+m2)+m2
        urand=float(iy)*s
        return
        end
```

Output from F77 Program 24

```
How many individuals
100
Seed for the Weibull distribution
2
Enter the index parameter
1.5
Enter b0
0.4
Enter b1
0.25
Enter seed for the group type
1

Simulation of the Weibull distribution with
Index =      .1500000E+01
b0    =      .4000000E+00
b1    =      .2500000E+00
```

cdf	time	rate	treat?
.997	1.670	1.916	1
.260	.301	1.492	
.329	.283	1.916	1
.636	.526	1.916	1
.399	.333	1.916	1
.199	.246	1.492	
.157	.161	1.916	1
.684	.574	1.916	1
.577	.607	1.492	
.263	.237	1.916	1
.378	.318	1.916	1
.493	.518	1.492	
.070	.090	1.916	1
.456	.375	1.916	1
.589	.619	1.492	
.988	1.803	1.492	
.554	.453	1.916	1
.953	1.413	1.492	
.630	.668	1.492	
.649	.538	1.916	1
.078	.098	1.916	1
.422	.350	1.916	1
.642	.683	1.492	
.433	.459	1.492	
.842	1.008	1.492	
.979	1.290	1.916	1
.909	.935	1.916	1
.866	.831	1.916	1
.377	.317	1.916	1
.729	.801	1.492	
.082	.130	1.492	
.033	.070	1.492	
.464	.489	1.492	
.369	.400	1.492	
.882	1.111	1.492	

.662	.708	1.492	
.641	.530	1.916	1
.822	.752	1.916	1
.864	.827	1.916	1
.045	.086	1.492	
.628	.665	1.492	
.368	.311	1.916	1
.350	.382	1.492	
.996	1.614	1.916	1
.533	.435	1.916	1
.239	.282	1.492	
.115	.165	1.492	
.581	.611	1.492	
.170	.171	1.916	1
.156	.205	1.492	
.919	1.239	1.492	
.823	.753	1.916	1
.336	.288	1.916	1
.762	.664	1.916	1
.491	.401	1.916	1
.113	.127	1.916	1
.388	.417	1.492	
.435	.461	1.492	
.039	.079	1.492	
.535	.437	1.916	1
.970	1.547	1.492	
.442	.468	1.492	
.905	1.186	1.492	
.838	.778	1.916	1
.231	.214	1.916	1
.897	.903	1.916	1
.427	.453	1.492	
.261	.302	1.492	
.373	.403	1.492	
.507	.414	1.916	1
.635	.674	1.492	
.929	.999	1.916	1
.880	.861	1.916	1
.560	.458	1.916	1
.853	1.034	1.492	
.057	.079	1.916	1
.266	.307	1.492	
.134	.184	1.492	
.802	.720	1.916	1
.652	.541	1.916	1
.694	.584	1.916	1
.987	1.784	1.492	
.314	.273	1.916	1
.962	1.473	1.492	

.292	.257	1.916	1
.105	.120	1.916	1
.228	.272	1.492	
.364	.307	1.916	1
.537	.438	1.916	1
.371	.313	1.916	1
.577	.472	1.916	1
.931	1.290	1.492	
.522	.547	1.492	
.330	.364	1.492	
.662	.551	1.916	1
.383	.412	1.492	
.473	.498	1.492	
.506	.531	1.492	
.091	.140	1.492	
.645	.686	1.492	

Sum of logs to the base e = -.8107597E+02
No. in treated group = 50

kappa	b0	b1
1.18761	.39780	.08030
1.19738	.47946	.08537
1.32728	.45004	.08600
1.22186	.47163	.08533
1.30262	.45513	.08584
1.24034	.46783	.08544
1.28804	.45809	.08575
1.25131	.46558	.08551
1.27947	.45983	.08569
1.25781	.46425	.08556
1.27443	.46086	.08566
1.26166	.46347	.08558
1.27146	.46146	.08564
1.26393	.46300	.08559
1.26971	.46182	.08563
1.26527	.46273	.08560
1.26528	.46203	.08562
1.26606	.46257	.08561
1.26807	.46216	.08562
1.26653	.46247	.08561
1.26772	.46223	.08562
1.26680	.46242	.08561
1.26750	.46227	.08562
1.26696	.46238	.08561
1.26738	.46230	.08562
1.26706	.46236	.08561
1.26731	.46231	.08562
1.26712	.46235	.08561
1.26726	.46232	.08562
1.26715	.46234	.08561
1.26724	.46233	.08562
1.26717	.46234	.08562
1.26722	.46233	.08562
1.26718	.46234	.08562
1.26721	.46233	.08562
1.26719	.46234	.08562
1.26721	.46233	.08562
1.26719	.46234	.08562
1.26720	.46233	.08562
1.26720	.46234	.08562
1.26720	.46233	.08562
1.26720	.46234	.08562