F77 Program 11

```
c       Fortran 77 program to do a simulation of the Weibull distribution
c       of the censored model. Realistic case.
c       This program may be adapted for other forms of censored models as
c       well as the simplest censoring model.
c       ***************************************************************
c               By Derek Dhammaloka FDX3 - 21st Jan. 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               knew is the new value of kappa to be entered by the user
c               rho is the rate to be entered by the user
c               loop is used in loop counters
c               n is the no. of individuals to be entered by the user
c               d is the no. of uncensored individuals
c               c is the censor time to be obtained using the equation
c               c=censorstart+((n-loop)/(n-1)) where censorstart is
c               the initial censor time to be entered by the user
c               i is the indicator variable (1 if censored, 0 otherwise)
c               x is equal to t if t is less than C, C otherwise
c               swops is the no. of swops
c               sorts is the no. of sorts
c               temp is used in sorting data values
c               list assigns the values 1,n so that it can be used in
c               sorting 1 variable in ascending or descending order and
c               the other variables have the same values as the original
c               data, but is being sorted
c               psn is the no. of trials in view at a certain time
c               ple is the product limit estimator
c               prcensor is the probability of censoring
c               iy is the seed to be entered by the user
c       ***************************************************************
c               Find the product limit estimator
c       ***************************************************************
      integer i(5000),list(5000)
      real cdf(5000),t(5000),c(5000),x(5000)
      real knew,censorstart
      real rho,ple,prcensor
      integer loop,temp,swops,sorts,n,d,psn,iy
      ple=1
      swops=1
      sorts=0
c       ***************************************************************
c               Input the no. of individuals
c               Also the index (kappa) and the rate (rho)
c       ***************************************************************
      print*, How many individuals'
      read*,n
c
c               Set the no. of failures to the no. of observations
```

```
c
      d=n
c
c
c           Set the no. of trials in view at time=0 to n+1
c
      psn=n+1
      print*,'Seed for the Weibull distribution'
      read*,iy
      print*,'Enter the index parameter'
      read*,knew
      print*,'Enter the rate parameter'
      read*,rho
      print*
      print*,'For the censor times'
      print*,'Enter the initial censor time'
      print*
      read*,censorstart
c     ***************************************************************************
c           Simulate the Weibull distribution
c           using the two parameters to obtain the remission times.
c           However, use an equation to obtain the censor times.
c     ***************************************************************************
      do 20 loop=1,n
           cdf(loop)=urand(iy)
           t(loop)=(-log(1-cdf(loop))/(rho**knew)**(1/knew)
           c(loop)=(censorstart*(n-1))+(n-loop)
           c(loop)=(c(loop))/(n-1)
c
c          Decide whether the individual is to be censored
c          using the censoring rules
c
           if(t(loop).ge.c(loop)) then
                                  i(loop)=1
                                  d=d-1
                                  x(loop)=c(loop)
           endif
           if(t(loop).lt.c(loop)) then
                                  i(loop)=0
                                  x(loop)=t(loop)
           endif
           list(loop)=loop
20    continue
c     ***************************************************************************
c           Print the headings
c     ***************************************************************************
      print*
      print*,'Realistic case model I'
      print*
      print*,'Simulation of the Weibull distribution with '
      print*,'Index = ',knew,' and rate = ',rho
      print*
      print*,'Initial censor time = ',censorstart
      print*
      write(*,25)
25    format(t3,'T',t10,'C',t40,'I',t55,'x')
c     ***************************************************************************
```

```
c               Output the remission and censor times
c               Also the indicator variable
c     ************************************************************************
      do 30 loop=1,n
               write(*,40)t(loop),c(loop),i(loop),x(loop)
40             format(f7.3,t8,f7.3,t37,i7.0,t50,f7.3,t68)
30    continue
      print*
c
c               Sort the x values in ascending order
c
50    if(swops.ne.0.and.sorts.lt.n-1) then
        swops=0
        sorts=sorts+1
        do 60 loop=1,n-sorts
               if(x(list(loop)).gt.x(list(loop+1))) then
                 temp=list(loop)
                 list(loop)=list(loop+1)
                 list(loop+1)=temp
                 swops=swops+1
                 endif
60      continue
        goto 50
      endif
c
c               Print headings
c
      print*
      print*,'Calculation of the product limit estimator'
      print*,'(No. of failures assumed to be 1 for uncensored obs.)'
      print*
      write(*,63)
63    format(t3,'atom',t14,'r',t27,'(1-(1/r))',t42,'ple')
c
c               Calculate the product limit estimator using a do loop
c               Assume that the no. of failures for each x value is 1
c               providing that the individual is not censored since
c               the times are on a continuous scale and it is rare
c               for ties to occur.
c
      do 65 loop=1,n
               psn=psn-1
c
c               Only print the atom, position, censor probability and
c               the product limit if there is no censoring in the
c               individual.
c
               if(i(list(loop)).eq.0) then
                        prcensor=psn-1
                        prcensor=prcensor/psn
                        ple=ple*(prcensor)
                        write(*,75)x((list(loop))),psn,prcensor,ple
75                      format(f7.3,t8,i7.0,t25,f7.4,t40,f7.4,t55)
                        endif
65    continue
c
```

```
c
c               Print the final product limit estimator
c
                print*
                print*, 'Product limit estimator =  ',ple
      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 11

```
How many individuals
75
Seed for the Weibull distribution
2
Enter the index parameter
2.5
Enter the rate parameter
1.5

For the censor times
Enter the initial censor time
0.5

Realistic case model I

Simulation of the Weibull distribution with
Index =      .2500000E+01 and rate =      .1500000E+01

Initial censor time =      .5000000E+00
```

T	C	I	x
1.339	1.500		1.339
.412	1.486		.412
.461	1.473		.461
.670	1.459		.670
.509	1.446		.509
.365	1.432		.365
.329	1.419		.329
.706	1.405		.706
.628	1.392		.628
.415	1.378		.415
.495	1.365		.495
.571	1.351		.571
.233	1.338		.233
.547	1.324		.547
.636	1.311		.636
1.207	1.297		1.207
.612	1.284		.612
1.043	1.270		1.043
.665	1.257		.665
.679	1.243		.679
.244	1.230		.244
.524	1.216		.524
.674	1.203		.674
.531	1.189		.531
.852	1.176		.852
1.147	1.162		1.147
.946	1.149		.946
.882	1.135		.882
.494	1.122		.494
.742	1.108		.742
.249	1.095		.249
.172	1.081		.172
.552	1.068		.552

.489	1.054		.489
.903	1.041		.903
.689	1.027		.689
.673	1.014		.673
.830	1.000		.830
.878	.986		.878
.194	.973		.194
.663	.959		.663
.488	.946		.488
.476	.932		.476
1.312	.919	1	.919
.598	.905		.598
.397	.892		.397
.287	.878		.287
.630	.865		.630
.341	.851		.341
.328	.838		.328
.964	.824	1	.824
.831	.811	1	.811
.467	.797		.467
.770	.784		.770
.569	.770		.569
.285	.757		.285
.502	.743		.502
.533	.730		.533
.184	.716		.184
.599	.703		.599
1.101	.689	1	.689
.537	.676		.537
.939	.662		.939
.847	.649	1	.649
.390	.635		.390
.926	.622	1	.926
.527	.608		.527
.413	.595		.413
.491	.581		.491
.580	.568	1	.568
.669	.554	1	.554
.984	.541	1	.541
.900	.527	1	.527
.616	.514	1	.514
.865	.500	1	.500

Calculation of the product limit estimator
(No. of failures assumed to be 1 for uncensored obs.)

atom	r	(1-(1/r))	ple
.172	75	.9867	.9867
.184	74	.9865	.9733
.194	73	.9863	.9600
.233	72	.9861	.9467
.244	71	.9859	.9333
.249	70	.9857	.9200
.285	69	.9855	.9067
.287	68	.9853	.8933
.328	67	.9851	.8800
.329	66	.9848	.8667
.341	65	.9846	.8533
.365	64	.9844	.8400
.390	63	.9841	.8267
.397	62	.9839	.8133
.412	61	.9836	.8000
.413	60	.9833	.7867
.415	59	.9831	.7733
.461	58	.9828	.7600
.467	57	.9825	.7467
.476	56	.9821	.7333
.488	55	.9818	.7200
.489	54	.9815	.7067
.491	53	.9811	.6933
.494	52	.9808	.6800
.495	51	.9804	.6667
.502	49	.9796	.6531
.509	48	.9792	.6395
.524	46	.9783	.6256
.527	44	.9773	.6113
.531	43	.9767	.5971
.533	42	.9762	.5829
.537	41	.9756	.5687
.547	39	.9744	.5541
.552	38	.9737	.5395
.569	35	.9714	.5241
.571	34	.9706	.5087
.598	33	.9697	.4933
.599	32	.9687	.4779
.612	31	.9677	.4624
.628	29	.9655	.4465
.630	28	.9643	.4306
.636	27	.9630	.4146
.663	24	.9583	.3973
.665	23	.9565	.3801
.670	22	.9545	.3628
.673	21	.9524	.3455
.674	20	.9500	.3282
.679	19	.9474	.3110
.689	18	.9444	.2937
.706	16	.9375	.2753
.742	15	.9333	.2570
.770	14	.9286	.2386

.830	11	.9091	.2169
.852	10	.9000	.1952
.878	9	.8889	.1735
.882	8	.8750	.1518
.903	7	.8571	.1302
.946	5	.8000	.1041
1.043	4	.7500	.0781
1.147	3	.6667	.0521
1.207	2	.5000	.0260
1.339	1	.0000	.0000

Product limit estimator = .0000000E+00