F77 Program 17

```
c       Fortran 77 program to do a simulation of the Weibull distribution
c       of the uncensored model.
c       This program may be adapted for censored models as well as for
c       uncensored models.
c       ****************************************************************
c                By Derek Dhammaloka FDX3 - 28th 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                sumln is the sum of the logs to the base e
c                sumpower is the sum of the powers
c                sumpowerln is the sum of product of (t**k) and ln t
c                spln2 is the same as sumpowerln, but has (ln pt)**2
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 difference between the new and old
c                values of kappa
c                ipp,ipk and ikk are the elements of the info. matrix
c                sep and sek are the standard errors of rho and kappa
c                ciplb95 and cipub95 are the lower and upper bounds of
c                rho with 95% confidence
c                ciklb95 and cikub95 are the lower and upper bounds of
c                kappa with 95% confidence
c                rho is the rate to be entered by the user
c                l0 and l1 are the values of the likelihood function
c                under H0 and H1 respectively, where l1 must be greater
c                than l0, but how much bigger?
c                lambda is the difference between l1 and l0
c                cvalchi5 is the critical value of chi-square at the 5
c                per cent level
c                netscaledev is the change in scaled deviance
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 is the seed to be entered by the user.
c       ****************************************************************
        real cdf(5000),t(5000)
        real sumln,sumpower,sumpowerln,spln2,kold,knew,kdiff
        real rho,tol,ipp,ipk,ikk
        integer loop,n,iy
        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'
      read*,iy
      print*,'Enter the index parameter'
      read*,knew
      print*,'Enter the rate parameter'
      read*,rho
c     *****************************************************************
c             Simulate the Weibull distribution
c             using the two parameters to obtain the remission times
c     *****************************************************************
      do 20 loop=1,n
              cdf(loop)=urand(iy)
              t(loop)=(-log(1-cdf(loop))/(rho**knew)**(1/knew)
20    continue
c     *****************************************************************
c             Print the headings
c     *****************************************************************
      print*
      print*,'Simulation of the Weibull distribution with
      print*,'Index = ',knew,' and rate = ',rho
      print*
      write(*,25)
25    format(t3,'cdf',t10,'time')
c     *****************************************************************
c             Output the cdf and remission time values
c     *****************************************************************
      do 30 loop=1,n
              write(*,40)cdf(loop),t(loop)
40            format(f7.3,t8,f7.3,t37)
30    continue
      print*
c             Update the statistics
      do 35 loop=1,n
              sumln=sumln+(log(t(loop))
35    continue
c             Print statistics
      print*,'Sum of logs to the base e = ',sumln
      print*
c             Print headings
      write(*,45)
45    format(t9,'kappa',t20,'rho')
c             Iteration loop
50    kold=knew
c             Update the statistics
      do 60 loop=1,n
              sumpower=sumpower+(t(loop)**kold)
              sumpowerln=sumpowerln+((t(loop)**kold)*log(t(loop)))
60    continue
      knew=1/((sumpowerln/sumpower)-(sumln/n))
      kdiff=knew-kold
c     Use the new value of kappa to give the new rho`value
      rho=(n/sumpower)**(1/knew)
c             Obtain sumpowerln2 using this new rho value
      do 70 loop=1,n
              spln2=spln2+(sumpower*(log(rho**t(loop)))**2)
```

```
70    continue
c     Use the new value of rho to obtain the elements of the
c     information matrix. ipk will have to appear in 2 lines, as it
c     is difficult to get this whole expression in 66 characters.
      ipp=(knew*n/(rho**2))+((knew*(knew-1)*sumpower*(rho**(knew-2))))
      ipk=-((n/rho)-((rho**(knew-1)*(1+(knew*log(rho)))+sumpower))
      ipk=ipk+(knew*(rho**(knew-1))*sumpowerln)
      ikk=-((-n/(knew*knew))-((rho**knew)*spln2))
      write(*,80)knew,rho
80    format(f12.5,f12.5)
c     ****************************************************************
c             If the old and new values of kappa do not agree to a
c             certain no. of decimal places, zero the sums (not sumln)
c             and carry on with the iteration.
c     ****************************************************************
      if(abs(kdiff).gt.tol) then
                          sumpower=0
                          sumpowerln=0
                          spln2=0
                          goto 50
      endif
c     ****************************************************************
c             Print the elements of the information matrix
c     ****************************************************************
      print*
      print*,'Elements of the information matrix -'
      print*,'ipp =  ',ipp
      print*,'ipk =  ',ipk
      print*,'ikk =  ',ikk
c     ****************************************************************
c             Assign and print the standard errors of rho and kappa
c     ****************************************************************
      print*
      sep=1/(sqrt(ipp))
      sek=1/(sqrt(ikk))
      print*,'Standard errors -'
      print*,'rho   =  ',sep
      print*,'kappa =  ',sek
c     ****************************************************************
c             Assign and print the 95% confidence intervals of rho and
c             kappa. Assume that the Asymptotic Theorem for MLEs hold.
c     ****************************************************************
      print*
      ciplb95=(rho)-(1.96*sep)
      cipub95=(rho)+(1.96*sep)
      ciklb95=(knew)-(1.96*sek)
      cikub95=(knew)+(1.96*sek)
      print*,'With 95% confidence -'
      print*,'rho   is between  ',ciplb95,'  and  ',cipub95
      print*,'kappa is between  ',ciklb95,'  and  ',cikub95
c
c             Perform log likelihood ratio test and use it to test
c             for exponentiality, ie by putting kappa = 1
c
      print*
c
```

```
c        Print null and alternative hypotheses
c
      print*,'H0 : kappa  =  1, for exponentiality'
      print*,'H1 : kappa  <> 1, for non-exponentiality'
      print*
c
c        The following statements give the likelihood
c        function of the Weibull distribution under the null
c        and alternative hypotheses
c
c        For l0 use the examined values of kappa and rho
c        For l1 use the estimated values of kappa and rho
c
      l0=(n*log(rho))-n
      l1=(n*log(knew))+(knew*n*log(rho))+((knew-1)*sumln)-n
c
c        Print the values of the likelihood function
c        under the null and alternative hypotheses
c
      print*,'Likelihood function under H0 = ',l0
      print*,'Likelihood function under H1 = ',l1
c
c        Assume that the large sample method holds and use
c        2 ln constant = chi-square, where ln constant is
c        equal to l1-l0 and 2 ln constant represents the
c        change in scaled deviance.
c
      lambda=l1-l0
      print*
      print*,'Value of l1 - l0 = ',lambda
      netscaledev=2*lambda
      print*
c
c        Do not proceed further with this test, unless
c        l1 is greater than or equal to l0
c
      if(lambda.lt.0) then
        print*,'Cannot use the likelihood ratio test, since'
        print*,'l1 is less than l0'
      else
        print*,'Change in scaled deviance  = ',netscaledev
c
c        The following statement represents the critical
c        value of chi-square at the 5 per cent level on
c        1 degree of freedom
c
        cvalchi5=3.841
        print*,'Chi-sq at 5% level on 1 df = ',cvalchi5
        print*
c
c        Compare the change in scaled deviance with the
c        critical value of chi-square at the 5% level
c        on 1 degree of freedom
c
c        If the change in scaled deviance is less than
c        this critical value of chi-square, then we
```

```
c           accept this null hypothesis. Otherwise we
c           reject this null hypothesis.
c
        if(netscaledev.lt.cvalchi5) then
         print*,'It is reasonable to assume exponentiality'
        else
         print*,'It is not reasonable to assume exponentiality'
        endif
      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 17

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

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

   cdf     time
   .997   1.339
   .260    .412
   .329    .461
   .636    .670
   .399    .509
   .199    .365
   .157    .329
   .684    .706
   .577    .628
   .263    .415
   .378    .495
   .493    .571
   .070    .233
   .456    .547
   .589    .636
   .988   1.207
   .554    .612
   .953   1.043
   .630    .665
   .649    .679
   .078    .244
   .422    .524
   .642    .674
   .433    .531
   .842    .852
   .979   1.147
   .909    .946
   .866    .882
   .377    .494
   .729    .742
   .082    .249
   .033    .172
   .464    .552
   .369    .489
   .882    .903
   .662    .689
   .641    .673
   .822    .830
   .864    .878
   .045    .194
   .628    .663
```

```
 .368   .488
 .350   .476
 .996  1.312
 .533   .598
 .239   .397
 .115   .287
 .581   .630
 .170   .341
 .156   .328
 .919   .964
 .823   .831
 .336   .467
 .762   .770
 .491   .569
 .113   .285
 .388   .502
 .435   .533
 .039   .184
 .535   .599
 .970  1.101
 .442   .537
 .905   .939
 .838   .847
 .231   .390
 .897   .926
 .427   .527
 .261   .413
 .373   .491
 .507   .580
 .635   .669
 .929   .984
 .880   .900
 .560   .616
 .853   .865

Sum of logs to the base e =    -.4189541E+02

       kappa         rho
     2.60583      1.39483
     2.52887      1.41903
     2.58419      1.40144
     2.54409      1.41409
     2.57298      1.40492
     2.55207      1.41152
     2.56715      1.40674
     2.55625      1.41019
     2.56412      1.40770
     2.55843      1.40950
     2.56254      1.40820
     2.55957      1.40914
     2.56171      1.40846
     2.56017      1.40895
     2.56128      1.40859
     2.56048      1.40885
     2.56106      1.40867
     2.56064      1.40880
```

```
2.56095     1.40870
2.56072     1.40877
2.56088     1.40872
2.56077     1.40876
2.56085     1.40873
2.56079     1.40875
2.56083     1.40874
2.56080     1.40875
2.56083     1.40874
2.56081     1.40875
2.56082     1.40874
2.56081     1.40874
2.56082     1.40874
2.56081     1.40874

Elements of the information matrix -
ipp =      .2478295E+03
ipk =      .2380329E+02
ikk =      .1443312E+04

Standard errors -
rho   =      .6352191E-01
kappa =      .2632206E-01

With 95% confidence -
rho   is between      .1284241E+01 and      .1533247E+01
kappa is between      .2509222E+01 and      .2612405E+01

H0 : kappa  =  1, for exponentiality
H1 : kappa  <> 1, for non-exponentiality

Likelihood function under H0 =     -.4929761E+02
Likelihood function under H1 =     -.4047501E+01

Value of l1 - l0 =      .4525011E+02

Change in scaled deviance  =      .9050021E+02
Chi-sq at 5% level on 1 df =      .3841000E+01

It is not reasonable to assume exponentiality
```