GEOLOGICAL SURVEY OF CANADA

OPEN FILE 1681

A FORTRAN 77 PROGRAM FOR POISSON REGRESSION

C.F. Chung

Canada

1988

GEOLOGICAL SURVEY OF CANADA

OPEN FILE 1681

A FORTRAN 77 PROGRAM FOR POISSON REGRESSION

C.F. Chung

1988

Appendix A. A listing of program POISSON.

```
      program Poisson
c        *         *         *         *         *
c                  Poisson Regression Analysis

c                         C.F. Chung
c                   Geological Survey of Canada
c                601 Booth Street, Ottawa, K1A 0E8
c
c                         June, 1987
c        *         *         *         *         *
      common x(20),b(20),ns(20),d(20,20)
      dimension v(20),f(20)
      character*20 infile,outfil
      character*60  fori
      double precision d
      data evel/0.01/,iter/50/
c
c        *         *         *         *         *
c     Enter description of input and output files
c        *         *         *         *         *
      write(*,*)' Enter input file name : '
      read(*,'(a)')infile
      open (30,file=infile,status='old')
      write(*,*)' Assign a name of output file for the results:'
      read(*,'(a)')outfil
      open (20,file=outfil,status='new')
      open (10,file='tape',status='unknown',form='unformatted')
c
      write(*,*)' Enter no. of samples = '
      read(*,*)n
      write(*,*)' Enter no. of variables = '
      read(*,*)nvar
      write(*,*)' Enter the format of data in input file : '
      read (*,'(a)') fori
      m=nvar+1
c        *         *         *         *         *
c   print title and input, output control statements
c        *         *         *         *         *
      write (20,6) evel,iter
    6 format(' level of convergence =     ',f1?.10,/,1x,
     1       'maximum permissible no. of iterations =',i3)
      write (20,7) nvar,n
    7 format(' no of variables = ',i5,/,' no of data = ',i5)
      write (20,8) fori
    8 format(' input format = ',a60)
      iout=m+1
   10 continue
      rewind 10
c        *         *         *         *         *
c     read input data
c        *         *         *         *         *
      x(1)=1.0
```

```
      do 70 i=1,n
         read (30,fori) yy,(x(j),j=2,m)
         write(10)  (x(j),j=1,m),yy
   70 continue
      close(30)
      rewind 10
c        *         *         *         *         *
c     perform OLS
c        *         *         *         *         *
      write (20,14)
   14 format(' As a starting point, call ordinary regression.')
      call regres(m,n)
      rewind 10
      write (20,13) (b(i),i=1,m)
   13 format(5f12.5)
      istep=1
c        *         *         *         *         *
c   start iteration
c        *         *         *         *         *
  130 continue
c        *         *         *         *         *
c    call subroutine poison for computation
c        *         *         *         *         *
      write (20,16) istep
   16 format(i3,' th estimates of coefficients')
      call poison(m,n,v)
      rewind 10
      write (20,13) (v(i),i=1,m)
      ilevel=0
      do 140 i=1,m
         c=abs(v(i)-b(i))
         if(c.gt. evel)then
            ilevel=ilevel+1
         endif
  140 continue
      write (20,17) ilevel
   17 format(' Number of scores > level of convergence = ',i5)
      if(ilevel.ne.0)then
         istep=istep+1
         if(istep.lt.iter)then
            do 145 i=1,m
               b(i)=v(i)
  145       continue
            go to 130
         else
            write (20,18)
   18       format(' No solution is obtained.')
            go to 220
         endif
      else
c        *         *         *         *         *
c   stop iteration, print ML estimates and variances
c        *         *         *                   *
         write (20,27)
```

```
   27    format(' Final estimates of coefficients')
         write (20,13) (b(i),i=1,m)
         write (20,21)
   21    format(' Variances  of coefficients')
         write (20,22) (d(i,i),i=1,m)
   22    format(5d20.10)
         write (20,24)
   24    format(' Standard deviations of coefficients')
         do 170 i=1,m
            d(i,i)=dsqrt(d(i,i))
  170    continue
         write (20,13) (d(i,i),i=1,m)
c        *         *         *         *         *
c        estimate no. of occurrences and deviance
c        *         *         *         *         *
         rewind 10
         dev=0.
         chi=0.
         do 200 i=1,n
            read(10)(x(j),j=1,iout)
            xx=0.
            do 190 k=1,m
               xx=xx+x(k)*b(k)
  190       continue
         write (20,31) x(iout),xx
   31    format(f10.3, f10.4)
         dd=x(iout)-xx
         dev=dev-dd
         chi=chi+dd*dd/xx
         if(x(iout).gt.0.0.and.xx.gt.0.0)then
            dev=dev+x(iout)*alog(x(iout)/xx)
         endif
  200    continue
         write(20,32)dev
   32    format(' Deviance = ',f20.5)
         write(20,33)chi
   33    format(' Pearsons chi-square = ', f20.5)
         close(10)
         close(20)
      endif
  220 stop
      end
c-----------------------------
      subroutine poison(m,n,v)
      common x(20),b(20),ns(20),d(20,20)
      dimension v(20),f(20)
      double precision d
      iout=m+1
      do 2 j=1,m
        do 1 i=1,m
          d(i,j)=0.0
    1   continue
        f(j)=0.0
    2 continue
```

```
      do 6 k=1,n
        read(10)      (x(i),i=1,iout)
        c=0.0
        do 3 j=1,m
          c=c+b(j)*x(j)
    3   continue
        if(c.le.0.0)c=0.0001
        c=1.0/c
        do 4 j=1,m
          f(j)=f(j)+c*x(iout)*x(j)
    4   continue
        do 5  i=1,m
        do 5  j=1,m
          d(i,j)=d(i,j)+x(i)*x(j)*c
    5   continue
    6 continue
      rewind 10
      call geninv(m)
      do 10 j=1,m
        c=0.0
        if(ns(j).ne.-1)then
          do 9 i=1,m
             if(ns(i).ne.-1)then
                c=c +d(i,j)*f(i)
             endif
    9     continue
        endif
        v(j)=c
   10 continue
      return
      end
c-------------------------------
      subroutine geninv(m)
      common c(20),f(20),ns(20),a(20,20)
      dimension b(20,20)
      double precision a,xx
      do 2 i=1,m
        ns(i)=-1
        do 1 j=1,m
          b(i,j)=a(i,j)
    1   continue
    2 continue
c
    3 continue
      do 9 i=1,m
         do 4 il=1,m
            c(il)=0.0
    4    continue
         xx=dabs(a(i,i))
         if((xx.gt.1.0d-7).and.(ns(i).ne.1))then
            k=i
            xx=a(k,k)
            ns(k)=1
            c(k)=1.0/xx
```

```
            do 5 j=1,m
               a(k,j)=a(k,j)/xx
    5       continue
            do 7 i2=1,m
               if(i2.ne.k)then
                  xx=a(i2,k)
                  c(i2)=-xx*c(k)
                  do 6 j=1,m
                     a(i2,j)=a(i2,j)-a(k,j)*xx
    6             continue
               endif
    7       continue
            do 8 i3=1,m
               a(i3,k)=c(i3)
    8       continue
         go to 3
         endif
    9 continue
c
      k=0
      do 12 i=1,m
         if(ns(i).ne.-1)then
            k=k+1
         endif
   12 continue
      if(k.eq. m)then
         write (20,13) m
   13    format(' The input matrix has full rank',i10)
      else
         write (20,14) k
   14    format(' The input matrix has rank',i10)
      endif
      do 21 k=1,m
         if(ns(k).ne.-1)then
            do 18 i=1,m
               cc=0.0
               if(ns(i).ne.-1)then
                  do 17 j=1,m
                     if(ns(j).ne.-1)then
                        cc=cc+a(k,j)*b(j,i)
                     endif
   17                continue
               endif
               c(i)=cc
   18       continue
            do 20 l=1,m
               if(l.ne.k)then
                  cc=abs(c(l))
                  if(cc.gt.1.0e-5)then
                     write (20,19)
   19                format(' warning - erroneous inverse matrix')
                     return
                  endif
               endif
```

```
   20       continue
         endif
   21 continue
      return
      end
c-----------------------------
      subroutine regres(m,n)
      common x(20),b(20),ns(20),d(20,20)
      dimension f(20)
      double precision d
      iout=m+1
      do 2 j=1,m
         do 1 i=1,m
            d(i,j)=0.0
    1    continue
         f(j)=0.0
    2 continue
      do 6 k=1,n
         read(10) (x(i),i=1,iout)
         do 4 j=1,m
            f(j)=f(j)+x(iout)*x(j)
    4    continue
         do 5  i=1,m
         do 5  j=1,m
            d(i,j)=d(i,j)+x(i)*x(j)
    5    continue
    6 continue
      rewind 10
      call geninv(m)
      do 10 j=1,m
         c=0.0
         if(ns(j).ne.-1)then
            do 9 i=1,m
               if(ns(i).ne.-1)then
                  c=c +d(i,j)*f(i)
               endif
    9       continue
            b(j)=c
         else
            b(j)=0.
         endif
   10 continue
      return
      end
```

Appendix C. Listing of the test data set D40.DAT.

```
1     2.312   0.263 0.60806 0.104040.000000 6.064380.000000
2     2.354   1.074 2.52820 1.445360.129470 3.587500.000000
1     2.010   1.565 3.14565 0.158790.673350 0.142710.000000
2     2.482   0.918 2.27848 0.263090.851326 4.633890.000000
2     1.191   0.000 0.00000 0.038110.003573 2.318880.000000
2     3.283   0.127 0.41694 0.952070.679581 2.787270.000000
1     4.499   0.469 2.11003 2.065040.026994 1.376690.000000
3     1.527   0.258 0.39397 0.265700.038175 5.217760.000000
2     1.350   0.283 0.38205 0.278100.008100 4.761450.005562
5     3.514   0.902 3.16963 1.971350.000000 0.593870.144177
1     0.465   0.274 0.12741 0.052080.016275 0.060450.058464
1     2.345   0.066 0.15477 0.241530.021105 6.561310.000000
2     3.268   0.287 0.93792 0.081700.189544 6.552340.000000
7     1.969   1.109 2.18362 0.675360.013783 1.187310.393421
1     1.444   0.098 0.14151 0.031760.229596 1.594180.033374
3     3.514   1.286 4.51900 0.449790.256522 2.178680.000000
2     1.486   0.356 0.52902 0.285310.627092 3.499530.054144
1     0.071   0.057 0.00405 0.000000.000000 0.005400.000000
1     2.660   0.906 2.40996 0.069160.311220 4.740120.000000
1     0.866   0.154 0.13336 0.046760.009526 0.170600.112320
0     0.056   0.000 0.00000 0.000000.000000 0.000000.000000
0     1.556   0.508 0.79045 0.767110.000000 2.220410.000000
0     1.556   0.257 0.39989 0.186720.040456 5.206380.000000
0     1.904   0.270 0.51408 0.009520.188496 1.281390.000000
0     0.930   0.443 0.41199 0.343170.027900 3.475410.000000
0     0.051   0.000 0.00000 0.000000.000000 0.297480.000000
0     2.060   0.105 0.21630 0.061800.611820 1.577960.000000
0     2.147   1.280 2.74816 0.261930.139555 1.069210.000000
0     1.474   0.000 0.00000 0.051590.166562 1.124660.000000
0     4.072   0.340 1.38448 0.464210.574152 1.180880.0
0     2.589   0.079 0.20453 0.189000.038835 7.777360.000000
0     0.635   0.215 0.13652 0.002540.019050 0.275590.000000
0     3.021   0.145 0.43804 0.141990.048336 3.969590.000000
0     0.000   0.000 0.00000 0.000000.000000 0.000000.000000
0     4.074   0.022 0.08963 0.032590.020370 7.141720.000000
0     3.416   0.348 1.18877 0.0     0.20496  5.632980.0
0     0.000   0.000 0.00000 0.000000.000000 0.000000.000000
0     1.344   0.177 0.23789 0.068540.000000 4.276610.000000
0     2.728   0.038 0.10366 0.147310.000000 6.503550.000000
0     1.909   0.259 0.49443 0.045810.032453 0.683420.000000
```