NAG Library Function Document

nag_estim_weibull (g07bec)

1 Purpose

nag_estim_weibull (g07bec) computes maximum likelihood estimates for arguments of the Weibull distribution from data which may be right-censored.

2 Specification

```
#include <nag.h>
#include <nagg07.h>
```

```
void nag_estim_weibull (Nag_CestMethod cens, Integer n, const double x[],
     const Integer ic[], double *beta, double *gamma, double tol,
     Integer maxit, double *sebeta, double *segam, double *corr, double *dev,
     Integer *nit, NagError *fail)
```

3 Description

nag_estim_weibull (g07bec) computes maximum likelihood estimates of the arguments of the Weibull distribution from exact or right-censored data.

For n realizations, y_i, from a Weibull distribution a value x_i is observed such that

$$x_i \leq y_i.$$

There are two situations:

(a) exactly specified observations, when $x_i = y_i$

(b) right-censored observations, known by a lower bound, when $x_i < y_i$.

The probability density function of the Weibull distribution, and hence the contribution of an exactly specified observation to the likelihood, is given by:

$$f(x;\lambda,\gamma) = \lambda\gamma x^{\gamma-1}\exp(-\lambda x^{\gamma}), \quad x > 0, \quad \text{for } \lambda,\gamma > 0;$$

while the survival function of the Weibull distribution, and hence the contribution of a right-censored observation to the likelihood, is given by:

$$S(x;\lambda,\gamma) = \exp(-\lambda x^{\gamma}), \quad x > 0, \quad \text{for } \lambda,\gamma > 0.$$

If d of the n observations are exactly specified and indicated by $i \in D$ and the remaining $(n-d)$ are right-censored, then the likelihood function, Like (λ,γ) is given by

$$\text{Like}(\lambda,\gamma) \propto (\lambda\gamma)^d \left(\prod_{i\in D} x_i^{\gamma-1}\right) \exp\left(-\lambda\sum_{i=1}^{n} x_i^{\gamma}\right).$$

To avoid possible numerical instability a different parameterisation β,γ is used, with $\beta = \log(\lambda)$. The kernel log-likelihood function, $L(\beta,\gamma)$, is then:

$$L(\beta,\gamma) = d\log(\gamma) + d\beta + (\gamma-1)\sum_{i\in D}\log(x_i) - e^{\beta}\sum_{i=1}^{n} x_i^{\gamma}.$$

If the derivatives $\frac{\partial L}{\partial\beta}$, $\frac{\partial L}{\partial\gamma}$, $\frac{\partial^2 L}{\partial\beta^2}$, $\frac{\partial^2 L}{\partial\beta\partial\gamma}$ and $\frac{\partial^2 L}{\partial\gamma^2}$ are denoted by L_1, L_2, L_{11}, L_{12} and L_{22}, respectively, then the maximum likelihood estimates, $\hat{\beta}$ and $\hat{\gamma}$, are the solution to the equations:

$$L_1\left(\hat{\beta},\hat{\gamma}\right)=0 \quad (1)$$

and

$$L_2\left(\hat{\beta},\hat{\gamma}\right)=0 \quad (2)$$

Estimates of the asymptotic standard errors of $\hat{\beta}$ and $\hat{\gamma}$ are given by:

$$\mathrm{se}\left(\hat{\beta}\right)=\sqrt{\frac{-L_{22}}{L_{11}L_{22}-L_{12}^2}}, \quad \mathrm{se}(\hat{\gamma})=\sqrt{\frac{-L_{11}}{L_{11}L_{22}-L_{12}^2}}.$$

An estimate of the correlation coefficient of $\hat{\beta}$ and $\hat{\gamma}$ is given by:

$$\frac{L_{12}}{\sqrt{L_{12}L_{22}}}.$$

Note: if an estimate of the original argument λ is required, then

$$\hat{\lambda}=\exp\left(\hat{\beta}\right) \quad \text{and} \quad \mathrm{se}\left(\hat{\lambda}\right)=\hat{\lambda}\,\mathrm{se}\left(\hat{\beta}\right).$$

The equations (1) and (2) are solved by the Newton–Raphson iterative method with adjustments made to ensure that $\hat{\gamma}>0.0$.

4 References

Gross A J and Clark V A (1975) *Survival Distributions: Reliability Applications in the Biomedical Sciences* Wiley

Kalbfleisch J D and Prentice R L (1980) *The Statistical Analysis of Failure Time Data* Wiley

5 Arguments

1: **cens** – Nag_CestMethod *Input*

On entry: indicates whether the data is censored or non-censored.

cens = Nag_NoCensored
: Each observation is assumed to be exactly specified. **ic** is not referenced.

cens = Nag_Censored
: Each observation is censored according to the value contained in **ic**$[i-1]$, for $i=1,2,\ldots,n$.

Constraint: **cens** = Nag_NoCensored or Nag_Censored.

2: **n** – Integer *Input*

On entry: n, the number of observations.

Constraint: **n** ≥ 1.

3: **x**[**n**] – const double *Input*

On entry: **x**$[i-1]$ contains the ith observation, x_i, for $i=1,2,\ldots,n$.

Constraint: **x**$[i-1]>0.0$, for $i=1,2,\ldots,n$.

4: **ic**[*dim*] – const Integer *Input*

Note: the dimension, *dim*, of the array **ic** must be at least

n when **cens** = Nag_Censored;
1 otherwise.

On entry: if **cens** = Nag_Censored, then **ic**$[i-1]$ contains the censoring codes for the ith observation, for $i = 1, 2, \ldots, n$.

If **ic**$[i-1] = 0$, the ith observation is exactly specified.

If **ic**$[i-1] = 1$, the ith observation is right-censored.

If **cens** = Nag_NoCensored, then **ic** is not referenced.

Constraint: if **cens** = Nag_Censored, then **ic**$[i-1] = 0$ or 1, for $i = 1, 2, \ldots, n$.

5: **beta** – double * *Output*

On exit: the maximum likelihood estimate, $\hat{\beta}$, of β.

6: **gamma** – double * *Input/Output*

On entry: indicates whether an initial estimate of γ is provided.

If **gamma** > 0.0, it is taken as the initial estimate of γ and an initial estimate of β is calculated from this value of γ.

If **gamma** ≤ 0.0, then initial estimates of γ and β are calculated, internally, providing the data contains at least two distinct exact observations. (If there are only two distinct exact observations, then the largest observation must not be exactly specified.) See Section 9 for further details.

On exit: contains the maximum likelihood estimate, $\hat{\gamma}$, of γ.

7: **tol** – double *Input*

On entry: the relative precision required for the final estimates of β and γ. Convergence is assumed when the absolute relative changes in the estimates of both β and γ are less than **tol**.

If **tol** $= 0.0$, then a relative precision of 0.000005 is used.

Constraint: ***machine precision*** $\leq$ **tol** ≤ 1.0 or **tol** $= 0.0$.

8: **maxit** – Integer *Input*

On entry: the maximum number of iterations allowed.

If **maxit** ≤ 0, then a value of 25 is used.

9: **sebeta** – double * *Output*

On exit: an estimate of the standard error of $\hat{\beta}$.

10: **segam** – double * *Output*

On exit: an estimate of the standard error of $\hat{\gamma}$.

11: **corr** – double * *Output*

On exit: an estimate of the correlation between $\hat{\beta}$ and $\hat{\gamma}$.

12: **dev** – double * *Output*

On exit: the maximized kernel log-likelihood, $L\left(\hat{\beta}, \hat{\gamma}\right)$.

13: **nit** – Integer * *Output*

On exit: the number of iterations performed.

14: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_BAD_PARAM

On entry, argument $\langle value \rangle$ had an illegal value.

NE_CONVERGENCE

Iterations have failed to converge in $\langle value \rangle$ iterations.

NE_DIVERGENCE

Iterations have diverged.

NE_INITIALIZATION

Unable to calculate initial values.

NE_INT

On entry, $\mathbf{n} = \langle value \rangle$.
Constraint: $\mathbf{n} \geq 1$.

NE_INT_ARRAY_ELEM_CONS

On entry, element $\langle value \rangle$ of **ic** was not valid. $\mathbf{ic}[I] = \langle value \rangle$.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

NE_OBSERVATIONS

On entry, there are no exact observations.

NE_OVERFLOW

Potential overflow detected.

NE_REAL

On entry, **tol** is invalid: $\mathbf{tol} = \langle value \rangle$.

NE_REAL_ARRAY_ELEM_CONS

On entry, observation ⟨*value*⟩ is ≤ 0.0. $\mathbf{x}[I] = \langle value \rangle$.

NE_SINGULAR

Hessian matrix is singular.

7 Accuracy

Given that the Weibull distribution is a suitable model for the data and that the initial values are reasonable the convergence to the required accuracy, indicated by **tol**, should be achieved.

8 Parallelism and Performance

nag_estim_weibull (g07bec) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The initial estimate of γ is found by calculating a Kaplan–Meier estimate of the survival function, $\hat{S}(x)$, and estimating the gradient of the plot of $\log\left(-\log\left(\hat{S}(x)\right)\right)$ against x. This requires the Kaplan–Meier estimate to have at least two distinct points.

The initial estimate of $\hat{\beta}$, given a value of $\hat{\gamma}$, is calculated as

$$\hat{\beta} = \log\left(\frac{d}{\sum_{i=1}^{n} x_i^{\hat{\gamma}}}\right).$$

10 Example

In a study, 20 patients receiving an analgesic to relieve headache pain had the following recorded relief times (in hours):

1.1 1.4 1.3 1.7 1.9 1.8 1.6 2.2 1.7 2.7 4.1 1.8 1.5 1.2 1.4 3.0 1.7 2.3 1.6 2.0

(See Gross and Clark (1975).) This data is read in and a Weibull distribution fitted assuming no censoring; the parameter estimates and their standard errors are printed.

10.1 Program Text

```
/* nag_estim_weibull (g07bec) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg07.h>

int main(void)
{
```

```
  /* Scalars */
  double   beta, corr, dev, gamma, sebeta, segam, tol;
  Integer  exit_status, i, maxit, n, nit;
  NagError fail;

  /* Arrays */
  double   *x = 0;
  Integer  *ic = 0;

  INIT_FAIL(fail);

  exit_status = 0;
  printf("nag_estim_weibull (g07bec) Example Program Results\n");

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif
#ifdef _WIN32
  scanf_s("%"NAG_IFMT"%*[^\n] ", &n);
#else
  scanf("%"NAG_IFMT"%*[^\n] ", &n);
#endif

  /* Allocate memory */
  if (!(x = NAG_ALLOC(n, double)) ||
      !(ic = NAG_ALLOC(n, Integer)))
    {
      printf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }

  for (i = 1; i <= n; ++i)
#ifdef _WIN32
    scanf_s("%lf", &x[i - 1]);
#else
    scanf("%lf", &x[i - 1]);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif

  /* If data were censored then ic would also be read in.
   *  Leave nag_estim_weibull (g07bec) to calculate initial values
   */
  gamma = 0.0;
  /* Use default values for tol and maxit */
  tol = 0.0;
  maxit = 0;
  /* nag_estim_weibull (g07bec).
   * Computes maximum likelihood estimates for parameters of
   * the Weibull distribution
   */
  nag_estim_weibull(Nag_NoCensored, n, x, ic, &beta, &gamma, tol, maxit,
                    &sebeta, &segam, &corr, &dev, &nit, &fail);
  if (fail.code != NE_NOERROR)
    {
      printf("Error from nag_estim_weibull (g07bec).\n%s\n",
              fail.message);
      exit_status = 1;
      goto END;
    }

  printf("\n");
  printf("Beta  = %10.4f Standard error = %10.4f\n", beta, sebeta);
  printf("Gamma = %10.4f Standard error = %10.4f\n", gamma, segam);
```

```
 END:
  NAG_FREE(x);
  NAG_FREE(ic);
  return exit_status;
}
```

10.2 Program Data

```
nag_estim_weibull (g07bec) Example Program Data
20
1.1 1.4 1.3 1.7 1.9 1.8 1.6 2.2 1.7 2.7
4.1 1.8 1.5 1.2 1.4 3.0 1.7 2.3 1.6 2.0
```

10.3 Program Results

```
nag_estim_weibull (g07bec) Example Program Results

Beta  =    -2.1073 Standard error =     0.4627
Gamma =     2.7870 Standard error =     0.4273
```