

NAG Library Routine Document

G02JDF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

1 Purpose

G02JDF fits a multi-level linear mixed effects regression model using restricted maximum likelihood (REML). Prior to calling G02JDF the initialization routine G02JCF must be called.

2 Specification

```

SUBROUTINE G02JDF (LVPR, VPR, NVPR, GAMMA, EFFN, RNKX, NCOV, LNLIKE, LB,      &
                  ID, LDID, B, SE, CZZ, LDCZZ, CXX, LDCXX, CXZ, LDCXZ,      &
                  RCOMM, ICOMM, IOPT, LIOPT, ROPT, LROPT, IFAIL)
INTEGER          LVPR, VPR(LVPR), NVPR, EFFN, RNKX, NCOV, LB,              &
                  ID(LDID, LB), LDID, LDCZZ, LDCXX, LDCXZ, ICOMM(*),      &
                  IOPT(LIOPT), LIOPT, LROPT, IFAIL
REAL (KIND=nag_wp) GAMMA(NVPR+1), LNLIKE, B(LB), SE(LB), CZZ(LDCZZ,*),  &
                  CXX(LDCXX,*), CXZ(LDCXZ,*), RCOMM(*), ROPT(LROPT)

```

3 Description

G02JDF fits a model of the form:

$$y = X\beta + Z\nu + \epsilon$$

where y is a vector of n observations on the dependent variable,

X is a known n by p design matrix for the *fixed* independent variables,

β is a vector of length p of unknown *fixed effects*,

Z is a known n by q design matrix for the *random* independent variables,

ν is a vector of length q of unknown *random effects*,

and ϵ is a vector of length n of unknown random errors.

Both ν and ϵ are assumed to have a Gaussian distribution with expectation zero and variance/covariance matrix defined by

$$\text{Var} \begin{bmatrix} \nu \\ \epsilon \end{bmatrix} = \begin{bmatrix} G & 0 \\ 0 & R \end{bmatrix}$$

where $R = \sigma_R^2 I$, I is the $n \times n$ identity matrix and G is a diagonal matrix. It is assumed that the random variables, Z , can be subdivided into $g \leq q$ groups with each group being identically distributed with expectation zero and variance σ_i^2 . The diagonal elements of matrix G therefore take one of the values $\{\sigma_i^2 : i = 1, 2, \dots, g\}$, depending on which group the associated random variable belongs to.

The model therefore contains three sets of unknowns: the fixed effects β , the random effects ν and a vector of $g + 1$ variance components γ , where $\gamma = \{\sigma_1^2, \sigma_2^2, \dots, \sigma_{g-1}^2, \sigma_g^2, \sigma_R^2\}$. Rather than working directly with γ , G02JDF uses an iterative process to estimate $\gamma^* = \{\sigma_1^2/\sigma_R^2, \sigma_2^2/\sigma_R^2, \dots, \sigma_{g-1}^2/\sigma_R^2, \sigma_g^2/\sigma_R^2, 1\}$. Due to the iterative nature of the estimation a set of initial values, γ_0 , for γ^* is required. G02JDF allows these initial values either to be supplied by you or calculated from the data using the minimum variance quadratic unbiased estimators (MIVQUE0) suggested by Rao (1972).

G02JDF fits the model by maximizing the restricted log-likelihood function:

$$-2l_R = \log(|V|) + (n-p)\log(r^T V^{-1} r) + \log|X^T V^{-1} X| + (n-p)(1 + \log(2\pi/(n-p)))$$

where

$$V = ZGZ^T + R, \quad r = y - Xb \quad \text{and} \quad b = (X^T V^{-1} X)^{-1} X^T V^{-1} y.$$

Once the final estimates for γ^* have been obtained, the value of σ_R^2 is given by

$$\sigma_R^2 = (r^T V^{-1} r)/(n-p).$$

Case weights, W_c , can be incorporated into the model by replacing $X^T X$ and $Z^T Z$ with $X^T W_c X$ and $Z^T W_c Z$ respectively, for a diagonal weight matrix W_c .

The log-likelihood, l_R , is calculated using the sweep algorithm detailed in Wolfinger *et al.* (1994).

4 References

- Goodnight J H (1979) A tutorial on the SWEEP operator *The American Statistician* **33(3)** 149–158
- Harville D A (1977) Maximum likelihood approaches to variance component estimation and to related problems *JASA* **72** 320–340
- Rao C R (1972) Estimation of variance and covariance components in a linear model *J. Am. Stat. Assoc.* **67** 112–115
- Stroup W W (1989) Predictable functions and prediction space in the mixed model procedure *Applications of Mixed Models in Agriculture and Related Disciplines Southern Cooperative Series Bulletin No. 343* 39–48
- Wolfinger R, Tobias R and Sall J (1994) Computing Gaussian likelihoods and their derivatives for general linear mixed models *SIAM Sci. Statist. Comput.* **15** 1294–1310

5 Arguments

Note: prior to calling G02JDF the initialization routine G02JCF must be called, therefore this documentation should be read in conjunction with the document for G02JCF.

In particular some argument names and conventions described in that document are also relevant here, but their definition has not been repeated. Specifically, RNDM, WEIGHT, N, NFF, NRF, NLSV, LEVELS, FIXED, DAT, LICOMM and LRCOMM should be interpreted identically in both routines.

- 1: LVPR – INTEGER *Input*
On entry: the sum of the number of random parameters and the random intercept flags specified in the call to G02JCF.
Constraint: $LVPR = \sum_i RNDM(1, i) + RNDM(2, i)$.
- 2: VPR(LVPR) – INTEGER array *Input*
On entry: a vector of flags indicating the mapping between the random variables specified in RNDM and the variance components, σ_i^2 . See Section 9 for more details.
Constraint: $1 \leq VPR(i) \leq NVPR$, for $i = 1, 2, \dots, LVPR$.
- 3: NVPR – INTEGER *Input*
On entry: g , the number of variance components being estimated (excluding the overall variance, σ_R^2).
Constraint: $1 \leq NVPR \leq LVPR$.

- 4: GAMMA(NVPR + 1) – REAL (KIND=nag_wp) array *Input/Output*
On entry: holds the initial values of the variance components, γ_0 , with GAMMA(i) the initial value for σ_i^2/σ_R^2 , for $i = 1, 2, \dots, NVPR$.
 If GAMMA(1) = -1.0, the remaining elements of GAMMA are ignored and the initial values for the variance components are estimated from the data using MIVQUE0.
On exit: GAMMA(i), for $i = 1, 2, \dots, NVPR$, holds the final estimate of σ_i^2 and GAMMA(NVPR + 1) holds the final estimate for σ_R^2 .
Constraint: GAMMA(1) = -1.0 or GAMMA(i) \geq 0.0, for $i = 1, 2, \dots, g$.
- 5: EFFN – INTEGER *Output*
On exit: effective number of observations. If there are no weights (i.e., WEIGHT = 'U'), or all weights are nonzero, then EFFN = N.
- 6: RNKX – INTEGER *Output*
On exit: the rank of the design matrix, X , for the fixed effects.
- 7: NCOV – INTEGER *Output*
On exit: number of variance components not estimated to be zero. If none of the variance components are estimated to be zero, then NCOV = NVPR.
- 8: LNLIKE – REAL (KIND=nag_wp) *Output*
On exit: $-2l_R(\hat{\gamma})$ where l_R is the log of the restricted maximum likelihood calculated at $\hat{\gamma}$, the estimated variance components returned in GAMMA.
- 9: LB – INTEGER *Input*
On entry: the dimension of the arrays B and SE and the second dimension of the array ID as declared in the (sub)program from which G02JDF is called.
Constraint: LB \geq NFF + NRF \times NLSV.
- 10: ID(LDID, LB) – INTEGER array *Output*
On exit: an array describing the parameter estimates returned in B. The first NLSV \times NRF columns of ID describe the parameter estimates for the random effects and the last NFF columns the parameter estimates for the fixed effects.
 The example program for this routine includes a demonstration of decoding the parameter estimates given in B using information from ID.
 For fixed effects:
 for $l = \text{NRF} \times \text{NLSV} + 1, \dots, \text{NRF} \times \text{NLSV} + \text{NFF}$
 if B(l) contains the parameter estimate for the intercept then

$$\text{ID}(1, l) = \text{ID}(2, l) = \text{ID}(3, l) = 0;$$
 if B(l) contains the parameter estimate for the i th level of the j th fixed variable, that is the vector of values held in the k th column of DAT when FIXED($j + 2$) = k then

$$\begin{aligned} \text{ID}(1, l) &= 0, \\ \text{ID}(2, l) &= j, \\ \text{ID}(3, l) &= i; \end{aligned}$$
 if the j th variable is continuous or binary, that is LEVELS(FIXED($j + 2$)) = 1, then

$$\text{ID}(3, l) = 0;$$
 any remaining rows of the l th column of ID are set to 0.

For random effects:

let

N_{R_b} denote the number of random variables in the b th random statement, that is $N_{R_b} = \text{RNDM}(1, b)$;

R_{jb} denote the j th random variable from the b th random statement, that is the vector of values held in the k th column of DAT when $\text{RNDM}(2 + j, b) = k$;

N_{S_b} denote the number of subject variables in the b th random statement, that is $N_{S_b} = \text{RNDM}(3 + N_{R_b}, b)$;

S_{jb} denote the j th subject variable from the b th random statement, that is the vector of values held in the k th column of DAT when $\text{RNDM}(3 + N_{R_b} + j, b) = k$;

$L(S_{jb})$ denote the number of levels for S_{jb} , that is $L(S_{jb}) = \text{LEVELS}(\text{RNDM}(3 + N_{R_b} + j, b))$;

then

for $l = 1, 2, \dots, \text{NRF} \times \text{NLSV}$, if $B(l)$ contains the parameter estimate for the i th level of R_{jb} when $S_{kb} = s_k$, for $k = 1, 2, \dots, N_{S_b}$ and $1 \leq s_k \leq L(S_{jb})$, i.e., s_k is a valid value for the k th subject variable, then

$$\begin{aligned} \text{ID}(1, l) &= b, \\ \text{ID}(2, l) &= j, \\ \text{ID}(3, l) &= i, \\ \text{ID}(3 + k, l) &= s_k, \quad k = 1, 2, \dots, N_{S_b}; \end{aligned}$$

if the parameter being estimated is for the intercept then $\text{ID}(2, l) = \text{ID}(3, l) = 0$;

if the j th variable is continuous, or binary, that is $L(S_{jb}) = 1$, then $\text{ID}(3, l) = 0$;

the remaining rows of the l th column of ID are set to 0.

In some situations, certain combinations of variables are never observed. In such circumstances all elements of the l th row of ID are set to -999 .

11: LDID – INTEGER *Input*

On entry: the first dimension of the array ID as declared in the (sub)program from which G02JDF is called.

Constraint: $\text{LDID} \geq 3 + \max_j (\text{RNDM}(3 + \text{RNDM}(1, j), j))$, i.e., $3 +$ maximum number of subject variables (see G02JCF).

12: B(LB) – REAL (KIND=nag_wp) array *Output*

On exit: the parameter estimates, with the first $\text{NRF} \times \text{NLSV}$ elements of B containing the parameter estimates for the random effects, ν , and the remaining NFF elements containing the parameter estimates for the fixed effects, β . The order of these estimates are described by the ID argument.

13: SE(LB) – REAL (KIND=nag_wp) array *Output*

On exit: the standard errors of the parameter estimates given in B.

14: CZZ(LDCZZ,*) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array CZZ must be at least $\text{NRF} \times \text{NLSV}$ (see G02JCF).

On exit: if $\text{NLSV} = 1$, then CZZ holds the lower triangular portion of the matrix $(1/\sigma^2) \left(Z^T \hat{R}^{-1} Z + \hat{G}^{-1} \right)$, where \hat{R} and \hat{G} are the estimates of R and G respectively. If $\text{NLSV} > 1$ then CZZ holds this matrix in compressed form, with the first NRF columns holding

the part of the matrix corresponding to the first level of the overall subject variable, the next NRF columns the part corresponding to the second level of the overall subject variable etc.

- 15: LDCZZ – INTEGER *Input*
On entry: the first dimension of the array CZZ as declared in the (sub)program from which G02JDF is called.
Constraint: LDCZZ \geq NRF.
- 16: CXX(LDCXX, *) – REAL (KIND=nag_wp) array *Output*
Note: the second dimension of the array CXX must be at least NFF (see G02JCF).
On exit: CXX holds the lower triangular portion of the matrix $(1/\sigma^2)X^T\hat{V}^{-1}X$, where \hat{V} is the estimated value of V .
- 17: LDCXX – INTEGER *Input*
On entry: the first dimension of the array CXX as declared in the (sub)program from which G02JDF is called.
Constraint: LDCXX \geq NFF.
- 18: CXZ(LDCXZ, *) – REAL (KIND=nag_wp) array *Output*
Note: the second dimension of the array CXZ must be at least NLSV \times NRF (see G02JCF).
On exit: if NLSV = 1, then CXZ holds the matrix $(1/\sigma^2)(X^T\hat{V}^{-1}Z)\hat{G}$, where \hat{V} and \hat{G} are the estimates of V and G respectively. If NLSV > 1 then CXZ holds this matrix in compressed form, with the first NRF columns holding the part of the matrix corresponding to the first level of the overall subject variable, the next NRF columns the part corresponding to the second level of the overall subject variable etc.
- 19: LDCXZ – INTEGER *Input*
On entry: the first dimension of the array CXZ as declared in the (sub)program from which G02JDF is called.
Constraint: LDCXZ \geq NFF.
- 20: RCOMM(*) – REAL (KIND=nag_wp) array *Communication Array*
Note: the dimension of the array RCOMM must be at least LRCOMM (see G02JCF).
On entry: communication array initialized by a call to G02JCF.
- 21: ICOMM(*) – INTEGER array *Communication Array*
Note: the dimension of the array ICOMM must be at least LICOMM (see G02JCF).
On entry: communication array initialized by a call to G02JCF.
- 22: IOPT(LIOPT) – INTEGER array *Input*
On entry: optional parameters passed to the optimization routine.
 By default G02JDF fits the specified model using a modified Newton optimization algorithm as implemented in E04LBF. In some cases, where the calculation of the derivatives is computationally expensive it may be more efficient to use a sequential QP algorithm. The sequential QP algorithm as implemented in E04UCA can be chosen by setting IOPT(5) = 1. If LIOPT < 5 or IOPT(5) \neq 1 then E04LBF will be used.
 Different optional parameters are available depending on the optimization routine used. In all cases, using a value of -1 will cause the default value to be used. In addition only the first

LIOPT values of IOPT are used, so for example, if only the first element of IOPT needs changing and default values for all other optional parameters are sufficient LIOPT can be set to 1.

E04LBF is being used.

<i>i</i>	Description	Equivalent E04LBF argument	Default Value
1	Number of iterations	MAXCAL	1000
2	Unit number for monitoring information	n/a	As returned by X04ABF
3	Print optional parameters (1 = print)	n/a	-1 (no printing performed)
4	Frequency that monitoring information is printed	IPRINT	-1
5	Optimizer used	n/a	n/a

If requested, monitoring information is displayed in a similar format to that given by E04LBF.

E04UCA is being used.

<i>i</i>	Description	Equivalent E04UCA argument	Default Value
1	Number of iterations	Major Iteration Limit	max (50, 3 × NVPR)
2	Unit number for monitoring information	n/a	As returned by X04ABF
3	Print optional parameters (1 = print, otherwise no print)	List/Nolist	-1 (no printing performed)
4	Frequency that monitoring information is printed	Major Print Level	0
5	Optimizer used	n/a	n/a
6	Number of minor iterations	Minor Iteration Limit	max (50, 3 × NVPR)
7	Frequency that additional monitoring information is printed	Minor Print Level	0

If LIOPT ≤ 0 then default values are used for all optional parameters and IOPT is not referenced.

23: LIOPT – INTEGER *Input*
On entry: length of the options array IOPT.

24: ROPT(LROPT) – REAL (KIND=nag_wp) array *Input*
On entry: optional parameters passed to the optimization routine.

Different optional parameters are available depending on the optimization routine used. In all cases, using a value of -1.0 will cause the default value to be used. In addition only the first LROPT values of ROPT are used, so for example, if only the first element of ROPT needs changing and default values for all other optional parameters are sufficient LROPT can be set to 1.

E04LBF is being used.

<i>i</i>	Description	Equivalent E04LBF argument	Default Value
1	Sweep tolerance	n/a	$\max\left(\sqrt{\text{eps}}, \sqrt{\text{eps}} \times \max_i(\text{zz}_{ii})\right)$
2	Lower bound for γ^*	n/a	eps/100
3	Upper bound for γ^*	n/a	10 ²⁰
4	Accuracy of linear minimizations	ETA	0.9
5	Accuracy to which solution is required	XTOL	0.0
6	Initial distance from solution	STEPMX	100000.0

E04UCA is being used.

<i>i</i>	Description	Equivalent E04UCA argument	Default Value
1	Sweep tolerance	n/a	$\max\left(\sqrt{\text{eps}}, \sqrt{\text{eps}} \times \max_i(\text{zz}_{ii})\right)$
2	Lower bound for γ^*	n/a	eps/100
3	Upper bound for γ^*	n/a	10^{20}
4	Line search tolerance	Line Search Tolerance	0.9
5	Optimality tolerance	Optimality Tolerance	eps ^{0.72}

where eps is the *machine precision* returned by X02AJF and zz_{ii} denotes the i diagonal element of $Z^T Z$.

If $\text{LROPT} \leq 0$ then default values are used for all optional parameters and ROPT is not referenced.

25: LROPT – INTEGER *Input*

On entry: length of the options array ROPT.

26: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, LVPR = $\langle \text{value} \rangle$.
Constraint: LVPR $\geq \langle \text{value} \rangle$.

IFAIL = 2

On entry, VPR($\langle \text{value} \rangle$) = $\langle \text{value} \rangle$ and NVPR = $\langle \text{value} \rangle$.
Constraint: $1 \leq \text{VPR}(i) \leq \text{NVPR}$.

IFAIL = 3

On entry, NVPR = $\langle \text{value} \rangle$.
Constraint: $1 \leq \text{NVPR} \leq \langle \text{value} \rangle$.

IFAIL = 4

On entry, GAMMA($\langle \text{value} \rangle$) = $\langle \text{value} \rangle$.
Constraint: GAMMA(1) = -1.0 or GAMMA(i) ≥ 0.0 .

IFAIL = 9

On entry, LB = $\langle value \rangle$.
Constraint: LB \geq $\langle value \rangle$.

IFAIL = 11

On entry, LDID = $\langle value \rangle$.
Constraint: LDID \geq $\langle value \rangle$.

IFAIL = 15

On entry, LDCZZ = $\langle value \rangle$.
Constraint: LDCZZ \geq $\langle value \rangle$.

IFAIL = 17

On entry, LDCXX = $\langle value \rangle$.
Constraint: LDCXX \geq $\langle value \rangle$.

IFAIL = 19

On entry, LDCXZ = $\langle value \rangle$.
Constraint: LDCXZ \geq $\langle value \rangle$.

IFAIL = 21

On entry, ICOMM has not been initialized correctly.

IFAIL = 32

On entry, at least one value of i , for $i = 1, 2, \dots, NVPR$, does not appear in VPR.

IFAIL = 101

Optimal solution found, but requested accuracy not achieved.

IFAIL = 102

Too many major iterations.

IFAIL = 103

Current point cannot be improved upon.

IFAIL = 104

At least one negative estimate for GAMMA was obtained. All negative estimates have been set to zero.

IFAIL = -99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -399

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

G02JDF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G02JDF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

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

9 Further Comments

The argument VPR gives the mapping between the random variables and the variance components. In most cases $VPR(i) = i$, for $i = 1, 2, \dots, \sum_i \text{RNDM}(1, i) + \text{RNDM}(2, i)$. However, in some cases it might be necessary to associate more than one random variable with a single variance component, for example, when the columns of DAT hold dummy variables.

Consider a dataset with three variables:

$$\text{DAT} = \begin{pmatrix} 1 & 1 & 3.6 \\ 2 & 1 & 4.5 \\ 3 & 1 & 1.1 \\ 1 & 2 & 8.3 \\ 2 & 2 & 7.2 \\ 3 & 2 & 6.1 \end{pmatrix}$$

where the first column corresponds to a categorical variable with three levels, the next to a categorical variable with two levels and the last column to a continuous variable. So in a call to G02JCF

$$\text{LEVELS} = (3 \quad 2 \quad 1)$$

also assume a model with no fixed effects, no random intercept, no nesting and all three variables being included as random effects, then

$$\begin{aligned} \text{FIXED} &= (0 \quad 0); \\ \text{RNDM} &= (3 \quad 0 \quad 1 \quad 2 \quad 3)^T. \end{aligned}$$

Each of the three columns in DAT therefore correspond to a single variable and hence there are three variance components, one for each random variable included in the model, so

$$\text{VPR} = (1 \quad 2 \quad 3).$$

This is the recommended way of supplying the data to G02JDF, however it is possible to reformat the above dataset by replacing each of the categorical variables with a series of dummy variables, one for each level. The dataset then becomes

$$\text{DAT} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 3.6 \\ 0 & 1 & 0 & 1 & 0 & 4.5 \\ 0 & 0 & 1 & 1 & 0 & 1.1 \\ 1 & 0 & 0 & 0 & 1 & 8.3 \\ 0 & 1 & 0 & 0 & 1 & 7.2 \\ 0 & 0 & 1 & 0 & 1 & 6.1 \end{pmatrix}$$

where each column only has one level

$$\text{LEVELS} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1).$$

Again a model with no fixed effects, no random intercept, no nesting and all variables being included as random effects is required, so

$$\begin{aligned} \text{FIXED} &= (0 \ 0); \\ \text{RNDM} &= (6 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6)^T. \end{aligned}$$

With the data entered in this manner, the first three columns of DAT correspond to a single variable (the first column of the original dataset) as do the next two columns (the second column of the original dataset). Therefore VPR must reflect this

$$\text{VPR} = (1 \ 1 \ 1 \ 2 \ 2 \ 3).$$

In most situations it is more efficient to supply the data to G02JCF in terms of categorical variables rather than transform them into dummy variables.

10 Example

This example fits a random effects model with three levels of nesting to a simulated dataset with 90 observations and 12 variables.

10.1 Program Text

```
! G02JDF Example Program Text
! Mark 26 Release. NAG Copyright 2016.

Module g02jdfc_mod

! G02JDF Example Program Module:
! Parameters and User-defined Routines

! .. Use Statements ..
Use nag_library, Only: nag_wp
! .. Implicit None Statement ..
Implicit None
! .. Accessibility Statements ..
Private
Public :: print_results
! .. Parameters ..
Integer, Parameter, Public :: nin = 5, nout = 6
Contains
Subroutine print_results(n,nff,nlsv,nrf,fixed,lfixed,nrndm,rdm,lrdm, &
    nvpr,vpr,lvpr,gamma,effn,rnkx,ncov,lnlike,lb,id,ldid,b,se)

! .. Scalar Arguments ..
Real (Kind=nag_wp), Intent (In) :: lnlike
Integer, Intent (In) :: effn, lb, ldid, ldrndm, lfixed, &
    lvpr, n, ncov, nff, nlsv, nrf, &
    nrndm, nvpr, rnkx

! .. Array Arguments ..
Real (Kind=nag_wp), Intent (In) :: b(lb), gamma(nvpr+1), se(lb)
Integer, Intent (In) :: fixed(lfixed), id(ldid,lb), &
    rdm(lrdm,nrndm), vpr(lvpr)

! .. Local Scalars ..
Integer :: aid, i, k, l, ns, nv, p, pb, tb, &
    tdid, vid
Character (120) :: pfmt, tfmt

! .. Executable Statements ..
! Display the output
Write (nout,*) 'Number of observations (N) = ', n
Write (nout,*) 'Number of random factors (NRF) = ', nrf
Write (nout,*) 'Number of fixed factors (NFF) = ', nff
Write (nout,*) 'Number of subject levels (NLSV) = ', &
    nlsv
Write (nout,*) 'Rank of X (RNKX) = ', &
    rnkx
Write (nout,*) 'Effective N (EFFN) = ', &
    effn
Write (nout,*) 'Number of nonzero variance components (NCOV) = ', ncov
```

```

Write (nout,99990) 'Parameter Estimates'
tdid = nff + nrf*nlsv

If (nrf>0) Then
  Write (nout,*)
  Write (nout,99990) 'Random Effects'
End If
pb = -999
pfmt = ' '
Do k = 1, nrf*nlsv
  tb = id(1,k)
  If (tb/=-999) Then
    vid = id(2,k)
    nv = rndm(1,tb)
    ns = rndm(3+nv,tb)
    Write (tfmt,*)(id(3+1,k),l=1,ns)
    If (pb/=tb .Or. tfmt/=pfmt) Then
      If (k/=1) Then
        Write (nout,*)
      End If
      Write (nout,99991) ' Subject: ', ('Variable ',rndm(3+nv+1,tb), &
        ' (Level ',id(3+1,k),' )',l=1,ns)
    End If
    If (vid==0) Then
      Intercept
      Write (nout,99994) b(k), se(k)
    Else
      !
      VID'th variable specified in RNDM
      aid = rndm(2+vid,tb)
      If (id(3,k)==0) Then
        Write (nout,99992) aid, b(k), se(k)
      Else
        Write (nout,99993) aid, id(3,k), b(k), se(k)
      End If
    End If
    pfmt = tfmt
  End If
  pb = tb
End Do

If (nff>0) Then
  Write (nout,*)
  Write (nout,99990) 'Fixed Effects'
End If
Do k = nrf*nlsv + 1, tdid
  If (vid/=-999) Then
    vid = id(2,k)
    If (vid==0) Then
      !
      Intercept
      Write (nout,99997) b(k), se(k)
    Else
      !
      VID'th variable specified in FIXED
      aid = fixed(2+vid)
      If (id(3,k)==0) Then
        Write (nout,99995) aid, b(k), se(k)
      Else
        Write (nout,99996) aid, id(3,k), b(k), se(k)
      End If
    End If
  End If
End Do

Write (nout,*)
Write (nout,*) 'Variance Components'
Write (nout,*) ' Estimate      Parameter      Subject'
Do k = 1, nvpr
  Write (nout,99999,Advance='NO') gamma(k)
  p = 0
  Do tb = 1, nrndm
    nv = rndm(1,tb)

```

```

      ns = rndm(3+nv,tb)
      If (rndm(2,tb)==1) Then
        p = p + 1
        If (vpr(p)==k) Then
          Write (nout,99988,Advance='NO')(rndm(3+nv+1,tb),l=1,ns)
        End If
      End If
      Do i = 1, nv
        p = p + 1
        If (vpr(p)==k) Then
          Write (nout,99989,Advance='NO') rndm(2+i,tb),
            (rndm(3+nv+1,tb),l=1,ns)
        End If
      End Do
    End Do
  Write (nout,*)
End Do
Write (nout,*)
Write (nout,99998) 'SIGMA**2          = ', gamma(nvpr+1)
Write (nout,99998) '-2LOG LIKELIHOOD = ', lnlike

Return
99999 Format (1X,F10.5,5X)
99998 Format (1X,A,F15.5)
99997 Format (3X,'Intercept',20X,F10.4,1X,F10.4)
99996 Format (3X,'Variable ',I2,' (Level ',I2,')',7X,F10.4,1X,F10.4)
99995 Format (3X,'Variable ',I2,18X,F10.4,1X,F10.4)
99994 Format (5X,'Intercept',18X,F10.4,1X,F10.4)
99993 Format (5X,'Variable ',I2,' (Level ',I2,')',5X,F10.4,1X,F10.4)
99992 Format (5X,'Variable ',I2,16X,F10.4,1X,F10.4)
99991 Format (1X,A,4(A,I2,A,I2,A,1X))
99990 Format (1X,A)
99989 Format (1X,'Variable',1X,I2,5X,'Variables',1X,100(I2,1X))
99988 Format (1X,'Intercept',7X,'Variables',1X,100(I2,1X))
End Subroutine print_results
End Module g02jdf_mod
Program g02jdf

!      G02JDF Example Main Program

!      .. Use Statements ..
Use nag_library, Only: g02jcf, g02jdf, nag_wp
Use g02jdf_mod, Only: nin, nout, print_results
!      .. Implicit None Statement ..
Implicit None
!      .. Local Scalars ..
Real (Kind=nag_wp)          :: lnlike
Integer                    :: effn, i, ifail, j, lb, ldcxx, ldcxz, &
                          ldczz, lddat, ldid, ldrndm, lfixed, &
                          licomm, liopt, lrcomm, lropt, lvpr, &
                          lwt, n, ncol, ncov, nff, nl, nlsv, &
                          nrf, nrndm, nv, nvpr, nzz, rnkx

Character (1)              :: weight
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: b(:), cxx(:,,:), cxz(:,,:), czz(:,,:), &
                          dat(:,,:), gamma(:), rcomm(:), &
                          ropt(:), se(:), wt(:), y(:)
Integer, Allocatable       :: fixed(:), icomm(:), id(:,,:), &
                          iopt(:), levels(:), rndm(:,,:), &
                          vpr(:)

!      .. Intrinsic Procedures ..
Intrinsic                  :: max
!      .. Executable Statements ..
Write (nout,*) 'G02JDF Example Program Results'
Write (nout,*)

!      Skip the heading in data file
Read (nin,*)

!      Read in the problem size
Read (nin,*) weight, n, ncol, nrndm, nvpr

```

```

!      Set LFIXED and LDRNDM to maximum value they could
!      be for this dataset
      lfixed = ncol + 2
      ldrndm = 3 + 2*ncol

      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      lddat = n
      Allocate (dat(lddat,ncol),levels(ncol),y(n),wt(lwt),fixed(lfixed),      &
        rndm(ldrndm,nrndm))

!      Read in the number of levels associated with each of the
!      independent variables
      Read (nin,*) levels(1:ncol)

!      Read in the fixed part of the model
      Read (nin,*)

!      Number of variables
      Read (nin,*) fixed(1)
      nv = fixed(1)

!      Intercept
      Read (nin,*) fixed(2)

!      Variable IDs
      If (nv>0) Then
        Read (nin,*) fixed(3:(nv+2))
      End If

!      Read in the random part of the model
      lvpr = 0
      Do j = 1, nrndm
!      Skip header
        Read (nin,*)

!      Number of variables and intercept
        Read (nin,*) rndm(1,j)
        Read (nin,*) rndm(2,j)
        nv = rndm(1,j)

!      Variable IDs
        If (nv>0) Then
          Read (nin,*)(rndm(i,j),i=3,nv+2)
        End If

!      Number of subject variables
        Read (nin,*) rndm(nv+3,j)
        nl = rndm(nv+3,j)

!      Subject variable IDs
        If (nl>0) Then
          Read (nin,*)(rndm(i,j),i=nv+4,nv+nl+3)
        End If
        lvpr = lvpr + rndm(2,j) + nv
      End Do

!      Read in the dependent and independent data
      If (lwt>0) Then
        Read (nin,*)(y(i),dat(i,1:ncol),wt(i),i=1,n)
      Else
        Read (nin,*)(y(i),dat(i,1:ncol),i=1,n)
      End If

      licomm = 2
      lrcomm = 0
      Allocate (icomm(licomm),rcomm(lrcomm))

```

```

!      Call the initialization routine once to get LRCOMM and LICOMM
      ifail = 0
      Call g02jcf(weight,n,ncol,dat,lddat,levels,y,wt,fixed,lfixed,nrndm,rndm, &
        ldrndm,nff,nlsv,nrf,rcomm,lrcomm,icomm,licomm,ifail)

!      Reallocate ICOMM and RCOMM
      licomm = icomm(1)
      lrcomm = icomm(2)
      Deallocate (icomm,rcomm)
      Allocate (icomm(licomm),rcomm(lrcomm))

!      Pre-process the data
      ifail = 0
      Call g02jcf(weight,n,ncol,dat,lddat,levels,y,wt,fixed,lfixed,nrndm,rndm, &
        ldrndm,nff,nlsv,nrf,rcomm,lrcomm,icomm,licomm,ifail)

!      Use the default options
      liopt = 0
      lropt = 0

!      Calculate LDID
      ldid = 0
      Do i = 1, nrndm
        nv = rndm(1,i)
        ldid = max(rndm(3+nv,i),ldid)
      End Do
      ldid = ldid + 3

      lb = nff + nrf*nlsv
      nzz = nrf*nlsv
      ldczz = nzz
      ldcxx = nff
      ldcxz = nff
      Allocate (vpr(lvpr),gamma(nvpr+1),id(ldid,lb),b(lb),se(lb), &
        czz(ldczz,nzz),cxx(ldcxx,nff),cxz(ldcxz,nzz),iopt(liopt),ropt(lropt))

!      Read in VPR
      Read (nin,*) vpr(1:lvpr)

!      Read in GAMMA
      Read (nin,*) gamma(1:nvpr)

!      Perform the analysis
      ifail = -1
      Call g02jdf(lvpr,vpr,nvpr,gamma,efn,rnkx,ncov,lnlike,lb,id,ldid,b,se, &
        czz,ldczz,cxx,ldcxx,cxz,ldcxz,rcomm,icomm,iopt,liopt,ropt,lropt,ifail)
      If (ifail/=0 .And. ifail<100) Then
        Go To 100
      End If

!      Display results
      Call print_results(n,nff,nlsv,nrf,fixed,lfixed,nrndm,rndm,ldrndm,nvpr, &
        vpr,lvpr,gamma,efn,rnkx,ncov,lnlike,lb,id,ldid,b,se)

100    Continue

      End Program g02jdf

```

10.2 Program Data

G02JDF Example Program Data

```

U 90 12 3 7           :: WEIGHT,N,NCOL,NRNDM,NVPR
2 3 2 3 2 3 1 4 5 2 3 3 :: LEVELS(1:NCOL)
## FIXED
2                   :: number of variables
1                   :: intercept
1 2                 :: variable IDs
## RANDOM 1
2                   :: number of variables

```

```

0                :: intercept
3 4             :: variable IDs
3              :: number of subject variables
10 11 12       :: subject variable IDs
## RANDOM 2
2              :: number of variables
0              :: intercept
5 6            :: variable IDs
2              :: number of subject variables
11 12         :: subject variable IDs
## RANDOM 3
3              :: number of variables
0              :: intercept
7 8 9         :: variable IDs
1              :: number of subject variables
12            :: subject variable IDs
 3.1100 1.0 3.0 2.0 1.0 2.0 2.0 -0.3160 4.0 2.0 1.0 1.0 1.0
 2.8226 1.0 1.0 1.0 3.0 1.0 2.0 -1.3377 1.0 4.0 1.0 1.0 1.0
 7.4543 1.0 3.0 1.0 3.0 1.0 3.0 -0.7610 4.0 2.0 1.0 1.0 1.0
 4.4313 2.0 3.0 2.0 1.0 1.0 3.0 -2.2976 4.0 2.0 1.0 1.0 1.0
 6.1543 2.0 2.0 1.0 3.0 2.0 3.0 -0.4263 2.0 1.0 1.0 1.0 1.0
-0.1783 2.0 1.0 2.0 3.0 1.0 3.0  1.4067 3.0 3.0 2.0 1.0 1.0
 4.6748 2.0 3.0 2.0 1.0 2.0 1.0 -1.4669 1.0 2.0 2.0 1.0 1.0
 7.0667 1.0 1.0 1.0 3.0 2.0 3.0  0.4717 2.0 4.0 2.0 1.0 1.0
 1.4262 1.0 3.0 2.0 3.0 2.0 1.0  0.4436 1.0 3.0 2.0 1.0 1.0
 7.7290 1.0 1.0 1.0 2.0 2.0 3.0 -0.5950 3.0 4.0 2.0 1.0 1.0
-2.1806 1.0 3.0 1.0 3.0 1.0 1.0 -1.7981 4.0 2.0 1.0 2.0 1.0
 6.8419 2.0 3.0 1.0 2.0 1.0 1.0  0.2397 1.0 4.0 1.0 2.0 1.0
 1.2590 1.0 2.0 2.0 1.0 2.0 3.0  0.4742 1.0 1.0 1.0 2.0 1.0
 8.8405 2.0 2.0 2.0 2.0 2.0 3.0  0.6888 3.0 1.0 1.0 2.0 1.0
 6.1657 2.0 1.0 2.0 3.0 1.0 3.0 -1.0616 3.0 5.0 1.0 2.0 1.0
-4.5605 1.0 2.0 2.0 2.0 2.0 1.0 -0.5356 1.0 3.0 2.0 2.0 1.0
-1.2367 1.0 3.0 2.0 2.0 1.0 1.0 -1.2963 2.0 5.0 2.0 2.0 1.0
-12.2932 1.0 2.0 2.0 1.0 2.0 2.0 -1.5389 3.0 2.0 2.0 2.0 1.0
-2.3374 2.0 3.0 1.0 1.0 2.0 2.0 -0.6408 2.0 1.0 2.0 2.0 1.0
 0.0716 1.0 2.0 2.0 2.0 1.0 1.0  0.6574 1.0 1.0 2.0 2.0 1.0
 0.1895 2.0 1.0 1.0 1.0 1.0 3.0  0.9259 1.0 2.0 1.0 3.0 1.0
 1.5608 2.0 2.0 2.0 1.0 2.0 2.0  1.5080 3.0 1.0 1.0 3.0 1.0
-0.8529 2.0 3.0 1.0 1.0 1.0 3.0  2.5821 2.0 3.0 1.0 3.0 1.0
-4.1169 1.0 2.0 2.0 1.0 2.0 3.0  0.4102 1.0 4.0 1.0 3.0 1.0
 3.9977 2.0 1.0 2.0 3.0 2.0 2.0  0.7839 2.0 5.0 1.0 3.0 1.0
-8.1277 1.0 2.0 2.0 3.0 2.0 1.0 -1.8812 4.0 2.0 2.0 3.0 1.0
-4.9656 1.0 2.0 1.0 3.0 2.0 3.0  0.7770 4.0 1.0 2.0 3.0 1.0
-0.6428 2.0 2.0 1.0 2.0 1.0 3.0  0.2590 3.0 1.0 2.0 3.0 1.0
-5.5152 2.0 3.0 2.0 2.0 2.0 3.0 -0.9250 3.0 3.0 2.0 3.0 1.0
-5.5657 2.0 2.0 1.0 3.0 2.0 3.0 -0.4831 1.0 5.0 2.0 3.0 1.0
14.8177 2.0 2.0 1.0 3.0 1.0 3.0  0.5046 3.0 3.0 1.0 1.0 2.0
16.9783 2.0 1.0 1.0 2.0 2.0 1.0 -0.6903 2.0 1.0 1.0 1.0 2.0
13.8966 1.0 3.0 2.0 2.0 2.0 1.0  1.6166 2.0 5.0 1.0 1.0 2.0
14.8166 2.0 2.0 2.0 2.0 1.0 3.0  0.2778 2.0 3.0 1.0 1.0 2.0
19.3640 2.0 3.0 2.0 2.0 1.0 2.0  1.9586 4.0 2.0 1.0 1.0 2.0
 9.5299 1.0 3.0 1.0 1.0 1.0 3.0  1.0506 2.0 5.0 2.0 1.0 2.0
12.0102 2.0 1.0 1.0 3.0 2.0 3.0  0.4871 1.0 1.0 2.0 1.0 2.0
 6.1551 2.0 1.0 2.0 3.0 2.0 1.0  2.0891 4.0 4.0 2.0 1.0 2.0
-1.7048 1.0 2.0 1.0 1.0 2.0 2.0  1.4338 4.0 3.0 2.0 1.0 2.0
 2.7640 1.0 1.0 2.0 3.0 1.0 2.0 -1.1196 3.0 4.0 2.0 1.0 2.0
 2.8065 1.0 3.0 1.0 1.0 2.0 1.0  0.3367 3.0 2.0 1.0 2.0 2.0
 0.0974 2.0 2.0 1.0 3.0 1.0 1.0  0.1092 2.0 2.0 1.0 2.0 2.0
-7.8080 1.0 1.0 1.0 2.0 2.0 2.0  0.4007 4.0 1.0 1.0 2.0 2.0
-18.0450 2.0 3.0 1.0 1.0 1.0 2.0  0.1460 3.0 5.0 1.0 2.0 2.0
-2.8199 2.0 1.0 2.0 3.0 1.0 3.0 -0.3877 3.0 4.0 1.0 2.0 2.0
 8.9893 1.0 1.0 1.0 2.0 2.0 1.0  0.6957 4.0 3.0 2.0 2.0 2.0
 3.7978 2.0 1.0 1.0 1.0 2.0 1.0 -0.4664 3.0 3.0 2.0 2.0 2.0
-6.3493 1.0 1.0 1.0 1.0 2.0 3.0  0.2067 2.0 4.0 2.0 2.0 2.0
 8.1411 2.0 1.0 2.0 1.0 1.0 2.0  0.4112 1.0 4.0 2.0 2.0 2.0
-7.5483 2.0 2.0 1.0 1.0 1.0 2.0 -1.3734 3.0 3.0 2.0 2.0 2.0
-0.4600 2.0 1.0 2.0 3.0 1.0 3.0  0.7065 1.0 3.0 1.0 3.0 2.0
-3.2135 1.0 2.0 2.0 2.0 1.0 2.0  1.3628 4.0 2.0 1.0 3.0 2.0
-6.6562 2.0 1.0 2.0 2.0 2.0 3.0 -0.5052 4.0 5.0 1.0 3.0 2.0
 5.1267 2.0 1.0 1.0 1.0 2.0 1.0 -1.3457 2.0 5.0 1.0 3.0 2.0
 3.5592 1.0 1.0 2.0 1.0 2.0 3.0 -1.8022 3.0 4.0 1.0 3.0 2.0

```

```

-4.4420 2.0 3.0 1.0 2.0 1.0 1.0 0.0116 2.0 4.0 2.0 3.0 2.0
-8.5965 2.0 2.0 1.0 3.0 2.0 3.0 -0.9075 1.0 3.0 2.0 3.0 2.0
-6.3187 2.0 2.0 2.0 2.0 2.0 3.0 -1.4707 1.0 1.0 2.0 3.0 2.0
-7.8953 2.0 2.0 1.0 1.0 2.0 1.0 -1.2938 2.0 3.0 2.0 3.0 2.0
-10.1383 1.0 3.0 1.0 3.0 2.0 2.0 -1.1660 4.0 4.0 2.0 3.0 2.0
-7.8850 1.0 2.0 1.0 1.0 2.0 3.0 0.0397 4.0 4.0 1.0 1.0 3.0
23.2001 1.0 3.0 1.0 2.0 1.0 3.0 -0.5987 3.0 2.0 1.0 1.0 3.0
5.5829 2.0 3.0 2.0 2.0 1.0 1.0 0.6683 3.0 3.0 1.0 1.0 3.0
-4.3698 2.0 2.0 1.0 1.0 2.0 2.0 -0.0106 1.0 3.0 1.0 1.0 3.0
2.1274 1.0 2.0 1.0 3.0 2.0 2.0 0.5885 1.0 3.0 1.0 1.0 3.0
-2.7184 1.0 1.0 1.0 1.0 1.0 2.0 0.4555 1.0 5.0 2.0 1.0 3.0
-17.9128 2.0 2.0 2.0 1.0 1.0 2.0 0.6502 4.0 3.0 2.0 1.0 3.0
-1.2708 1.0 1.0 1.0 3.0 1.0 1.0 -0.1601 1.0 3.0 2.0 1.0 3.0
-24.2735 2.0 2.0 1.0 3.0 2.0 3.0 1.6910 1.0 1.0 2.0 1.0 3.0
-14.7374 2.0 2.0 2.0 3.0 1.0 2.0 0.1053 4.0 4.0 2.0 1.0 3.0
0.1713 2.0 1.0 2.0 3.0 2.0 2.0 -0.4037 3.0 4.0 1.0 2.0 3.0
8.0006 1.0 3.0 2.0 3.0 1.0 3.0 -0.5853 3.0 2.0 1.0 2.0 3.0
1.2100 2.0 3.0 2.0 1.0 1.0 1.0 -0.3037 1.0 3.0 1.0 2.0 3.0
3.3307 1.0 3.0 1.0 1.0 2.0 2.0 -0.0774 1.0 4.0 1.0 2.0 3.0
-22.6713 2.0 3.0 1.0 2.0 2.0 1.0 0.4733 4.0 5.0 1.0 2.0 3.0
7.5562 1.0 3.0 2.0 2.0 1.0 2.0 -0.0354 4.0 2.0 2.0 2.0 3.0
-7.0694 1.0 3.0 2.0 2.0 1.0 1.0 -0.6640 2.0 1.0 2.0 2.0 3.0
3.7159 2.0 3.0 1.0 3.0 1.0 1.0 0.0335 4.0 4.0 2.0 2.0 3.0
-4.3135 1.0 2.0 2.0 2.0 1.0 3.0 0.1351 1.0 1.0 2.0 2.0 3.0
-14.5577 1.0 1.0 2.0 1.0 2.0 3.0 -0.5951 3.0 4.0 2.0 2.0 3.0
-12.5107 2.0 2.0 2.0 3.0 1.0 3.0 0.2735 3.0 2.0 1.0 3.0 3.0
4.7708 2.0 2.0 1.0 1.0 1.0 3.0 0.3157 1.0 2.0 1.0 3.0 3.0
13.2797 2.0 2.0 2.0 1.0 1.0 1.0 -1.0843 2.0 3.0 1.0 3.0 3.0
-6.3243 1.0 2.0 2.0 1.0 2.0 2.0 -0.0836 4.0 2.0 1.0 3.0 3.0
-7.0549 2.0 1.0 2.0 1.0 1.0 2.0 -0.2884 2.0 1.0 1.0 3.0 3.0
-9.2713 2.0 3.0 2.0 3.0 2.0 3.0 -0.1006 1.0 2.0 2.0 3.0 3.0
-18.7788 1.0 3.0 1.0 2.0 2.0 3.0 0.5710 1.0 3.0 2.0 3.0 3.0
-7.7230 1.0 1.0 2.0 1.0 1.0 2.0 0.2776 2.0 3.0 2.0 3.0 3.0
-22.7230 2.0 3.0 2.0 2.0 1.0 3.0 -0.7561 4.0 4.0 2.0 3.0 3.0
-11.6609 1.0 2.0 2.0 2.0 1.0 2.0 1.5549 1.0 4.0 2.0 3.0 3.0
1 2 3 4 5 6 7
-1.0 -1.0 -1.0 -1.0 -1.0 -1.0

```

```

:: Y, X
:: VPR
:: GAMMA(1:NVPR)

```

10.3 Program Results

G02JDF Example Program Results

```

Number of observations (N)           = 90
Number of random factors (NRF)      = 55
Number of fixed factors (NFF)       = 4
Number of subject levels (NLSV)     = 3
Rank of X (RNKX)                    = 4
Effective N (EFFN)                   = 90
Number of nonzero variance components (NCOV) = 7
Parameter Estimates

```

Random Effects

```

Subject: Variable 10 (Level 1) Variable 11 (Level 1) Variable 12 (Level 1)
Variable 3 (Level 1)           2.1561    3.7946
Variable 3 (Level 2)           1.8951    3.9284
Variable 4 (Level 1)           0.6496    3.1617

```

```

Subject: Variable 10 (Level 1) Variable 11 (Level 1) Variable 12 (Level 1)
Variable 4 (Level 3)           0.7390    3.1424

```

```

Subject: Variable 10 (Level 2) Variable 11 (Level 1) Variable 12 (Level 1)
Variable 3 (Level 1)           1.4216    3.3773
Variable 3 (Level 2)           -2.8921   3.3953
Variable 4 (Level 1)           3.6789    2.3162
Variable 4 (Level 2)           -1.9742   2.3887
Variable 4 (Level 3)           -2.2088   2.0697

```

```

Subject: Variable 10 (Level 1) Variable 11 (Level 2) Variable 12 (Level 1)
Variable 3 (Level 1)           -2.9659   3.9127
Variable 3 (Level 2)           2.7951    4.7183
Variable 4 (Level 1)           -4.7330   2.3094
Variable 4 (Level 2)           5.5161    2.2330
Variable 4 (Level 3)           -0.8417   2.3826

```

Subject: Variable 10 (Level 2)	Variable 11 (Level 2)	Variable 12 (Level 1)
Variable 3 (Level 1)	4.2202	3.6675
Variable 3 (Level 2)	-4.3883	3.4424
Variable 4 (Level 1)	-1.1391	3.2187
Variable 4 (Level 2)	1.0814	3.0654
Subject: Variable 10 (Level 1)	Variable 11 (Level 3)	Variable 12 (Level 1)
Variable 3 (Level 1)	0.3391	4.0647
Variable 3 (Level 2)	0.1502	3.4787
Variable 4 (Level 1)	-1.0026	2.4363
Subject: Variable 10 (Level 1)	Variable 11 (Level 3)	Variable 12 (Level 1)
Variable 4 (Level 3)	1.1703	2.6365
Subject: Variable 10 (Level 2)	Variable 11 (Level 3)	Variable 12 (Level 1)
Variable 3 (Level 1)	1.2658	3.4819
Variable 3 (Level 2)	-1.5356	3.9097
Subject: Variable 10 (Level 2)	Variable 11 (Level 3)	Variable 12 (Level 1)
Variable 4 (Level 2)	0.7992	2.7902
Variable 4 (Level 3)	-0.8916	2.8763
Subject: Variable 11 (Level 1)	Variable 12 (Level 1)	
Variable 5 (Level 1)	-0.4885	2.8206
Variable 5 (Level 2)	1.8829	2.7530
Variable 6 (Level 1)	0.9249	3.7747
Variable 6 (Level 2)	-2.3568	3.1624
Variable 6 (Level 3)	4.3117	3.1474
Subject: Variable 11 (Level 2)	Variable 12 (Level 1)	
Variable 5 (Level 1)	1.3898	2.9362
Variable 5 (Level 2)	-1.5729	2.8909
Variable 6 (Level 1)	0.2111	3.9967
Variable 6 (Level 2)	-3.7083	4.2866
Variable 6 (Level 3)	3.1190	4.7983
Subject: Variable 11 (Level 3)	Variable 12 (Level 1)	
Variable 5 (Level 1)	1.7352	3.1370
Variable 5 (Level 2)	-1.6165	3.1713
Variable 6 (Level 1)	-1.1102	3.9374
Variable 6 (Level 2)	4.4877	3.6980
Variable 6 (Level 3)	-3.1325	3.1966
Subject: Variable 12 (Level 1)		
Variable 7	0.6827	0.5060
Variable 8 (Level 1)	1.5964	1.3206
Variable 8 (Level 2)	-0.7533	1.5663
Variable 8 (Level 3)	0.4035	1.6840
Variable 8 (Level 4)	-0.8523	1.7518
Variable 9 (Level 1)	0.5699	1.6236
Variable 9 (Level 2)	0.0012	1.9111
Variable 9 (Level 3)	-0.2850	1.9245
Variable 9 (Level 4)	0.4468	2.0329
Variable 9 (Level 5)	0.0030	2.1390
Subject: Variable 10 (Level 1)	Variable 11 (Level 1)	Variable 12 (Level 2)
Variable 3 (Level 1)	6.2551	3.3595
Variable 3 (Level 2)	5.6085	3.4127
Subject: Variable 10 (Level 1)	Variable 11 (Level 1)	Variable 12 (Level 2)
Variable 4 (Level 2)	2.6922	2.7542
Variable 4 (Level 3)	1.3742	2.8068
Subject: Variable 10 (Level 2)	Variable 11 (Level 1)	Variable 12 (Level 2)
Variable 3 (Level 1)	1.5647	3.8353
Variable 3 (Level 2)	-2.7565	3.9041
Variable 4 (Level 1)	-0.8621	2.8257
Subject: Variable 10 (Level 2)	Variable 11 (Level 1)	Variable 12 (Level 2)
Variable 4 (Level 3)	0.4536	2.8070
Subject: Variable 10 (Level 1)	Variable 11 (Level 2)	Variable 12 (Level 2)
Variable 3 (Level 1)	-10.1544	3.3433
Variable 3 (Level 2)	3.2446	4.1221
Variable 4 (Level 1)	-2.9419	2.3508
Variable 4 (Level 2)	0.2510	3.0675
Variable 4 (Level 3)	0.3224	2.9710

Subject: Variable 10 (Level 2)	Variable 11 (Level 2)	Variable 12 (Level 2)
Variable 3 (Level 1)	-1.3577	3.1925
Variable 3 (Level 2)	8.1277	3.9975
Variable 4 (Level 1)	-0.4290	2.4578
Variable 4 (Level 2)	2.7495	2.5821
Subject: Variable 10 (Level 1)	Variable 11 (Level 3)	Variable 12 (Level 2)
Variable 3 (Level 1)	4.8432	4.0069
Variable 3 (Level 2)	0.0370	3.6006
Variable 4 (Level 1)	3.0713	2.2706
Variable 4 (Level 2)	-1.8899	2.4756
Variable 4 (Level 3)	0.4914	2.2914
Subject: Variable 10 (Level 2)	Variable 11 (Level 3)	Variable 12 (Level 2)
Variable 3 (Level 1)	-4.4766	3.3355
Variable 3 (Level 2)	-3.7936	4.0759
Variable 4 (Level 1)	-0.5459	2.7097
Variable 4 (Level 2)	-1.5619	2.7412
Variable 4 (Level 3)	-0.7269	2.9735
Subject: Variable 11 (Level 1)	Variable 12 (Level 2)	
Variable 5 (Level 1)	4.8653	3.0706
Variable 5 (Level 2)	0.9011	3.0696
Variable 6 (Level 1)	6.9277	3.8411
Variable 6 (Level 2)	-1.3108	3.1667
Variable 6 (Level 3)	6.2916	3.5327
Subject: Variable 11 (Level 2)	Variable 12 (Level 2)	
Variable 5 (Level 1)	-0.4047	3.0956
Variable 5 (Level 2)	0.3291	3.0784
Variable 6 (Level 1)	6.9096	3.3073
Variable 6 (Level 2)	-1.0680	3.6213
Variable 6 (Level 3)	-5.9977	3.7299
Subject: Variable 11 (Level 3)	Variable 12 (Level 2)	
Variable 5 (Level 1)	-1.0925	3.0994
Variable 5 (Level 2)	-0.7392	2.9900
Variable 6 (Level 1)	2.7758	3.8748
Variable 6 (Level 2)	-6.3526	3.3014
Variable 6 (Level 3)	-0.2060	3.6481
Subject: Variable 12 (Level 2)		
Variable 7	0.1711	0.5785
Variable 8 (Level 1)	1.7186	1.9143
Variable 8 (Level 2)	-0.6768	1.7352
Variable 8 (Level 3)	-0.0439	1.6395
Variable 8 (Level 4)	0.1463	1.5358
Variable 9 (Level 1)	0.9761	2.3930
Variable 9 (Level 2)	6.5436	1.8193
Variable 9 (Level 3)	-1.5504	1.8527
Variable 9 (Level 4)	0.1047	2.0244
Variable 9 (Level 5)	-3.9386	1.7937
Subject: Variable 10 (Level 1)	Variable 11 (Level 1)	Variable 12 (Level 3)
Variable 3 (Level 1)	10.6802	3.2596
Variable 3 (Level 2)	-1.0290	3.7842
Variable 4 (Level 1)	-2.8612	2.2917
Variable 4 (Level 2)	3.9265	2.8934
Variable 4 (Level 3)	2.2427	2.3737
Subject: Variable 10 (Level 2)	Variable 11 (Level 1)	Variable 12 (Level 3)
Variable 3 (Level 1)	-6.2076	3.3642
Variable 3 (Level 2)	-8.7670	3.8463
Variable 4 (Level 1)	-2.9251	2.4657
Subject: Variable 10 (Level 2)	Variable 11 (Level 1)	Variable 12 (Level 3)
Variable 4 (Level 3)	-2.2077	2.3743
Subject: Variable 10 (Level 1)	Variable 11 (Level 2)	Variable 12 (Level 3)
Variable 3 (Level 1)	-3.3334	3.4665
Variable 3 (Level 2)	-0.3111	3.2650
Variable 4 (Level 1)	1.5131	2.4890
Variable 4 (Level 2)	-3.0345	3.0562
Variable 4 (Level 3)	0.2722	2.8300
Subject: Variable 10 (Level 2)	Variable 11 (Level 2)	Variable 12 (Level 3)
Variable 3 (Level 1)	6.5905	4.0386

Variable 3 (Level 2)	-5.3168	3.4549
Variable 4 (Level 1)	-3.5280	2.9663
Variable 4 (Level 2)	1.7056	2.9293
Variable 4 (Level 3)	2.2590	3.1780
Subject: Variable 10 (Level 1) Variable 11 (Level 3) Variable 12 (Level 3)		
Variable 3 (Level 1)	8.1889	4.1429
Variable 3 (Level 2)	-1.5388	3.3333
Variable 4 (Level 1)	3.4338	2.6376
Subject: Variable 10 (Level 1) Variable 11 (Level 3) Variable 12 (Level 3)		
Variable 4 (Level 3)	-1.1544	2.9885
Subject: Variable 10 (Level 2) Variable 11 (Level 3) Variable 12 (Level 3)		
Variable 3 (Level 1)	-4.4243	4.0049
Variable 3 (Level 2)	-4.1349	3.1248
Variable 4 (Level 1)	1.0460	2.6550
Variable 4 (Level 2)	-4.4844	2.2843
Variable 4 (Level 3)	0.5046	2.6926
Subject: Variable 11 (Level 1) Variable 12 (Level 3)		
Variable 5 (Level 1)	5.3030	3.0278
Variable 5 (Level 2)	-8.1794	3.1335
Variable 6 (Level 1)	-0.8188	3.7810
Variable 6 (Level 2)	-2.5078	3.1514
Variable 6 (Level 3)	-2.6138	3.4600
Subject: Variable 11 (Level 2) Variable 12 (Level 3)		
Variable 5 (Level 1)	4.3331	3.1489
Variable 5 (Level 2)	-5.6142	3.1649
Variable 6 (Level 1)	-5.8804	3.1770
Variable 6 (Level 2)	5.4265	3.3006
Variable 6 (Level 3)	-2.1917	3.2156
Subject: Variable 11 (Level 3) Variable 12 (Level 3)		
Variable 5 (Level 1)	0.4305	2.9144
Variable 5 (Level 2)	-1.4620	3.0119
Variable 6 (Level 1)	14.3595	3.9254
Variable 6 (Level 2)	-5.2399	3.3099
Variable 6 (Level 3)	-11.2498	3.2212
Subject: Variable 12 (Level 3)		
Variable 7	-0.3839	0.6755
Variable 8 (Level 1)	2.7549	1.6017
Variable 8 (Level 2)	0.4377	1.8826
Variable 8 (Level 3)	-0.2261	1.9909
Variable 8 (Level 4)	-4.5051	1.5398
Variable 9 (Level 1)	-4.7091	2.1458
Variable 9 (Level 2)	3.7940	1.9872
Variable 9 (Level 3)	-1.7994	1.8614
Variable 9 (Level 4)	0.4480	1.9016
Variable 9 (Level 5)	-0.6047	2.4729
Fixed Effects		
Intercept	1.6433	2.4596
Variable 1 (Level 2)	-1.6224	0.8549
Variable 2 (Level 2)	-2.4817	1.1414
Variable 2 (Level 3)	0.4624	1.2133
Variance Components		
Estimate	Parameter	Subject
36.32491	Variable 3	Variables 10 11 12
12.45090	Variable 4	Variables 10 11 12
19.62767	Variable 5	Variables 11 12
40.53480	Variable 6	Variables 11 12
0.56320	Variable 7	Variables 12
5.81968	Variable 8	Variables 12
10.86069	Variable 9	Variables 12
SIGMA**2 = 0.00239		
-2LOG LIKELIHOOD = 608.19449		