NAG Library Routine Document

g22ydf

Note: please be advised that this routine is classed as 'experimental' and its interface may be developed further in the future. Please see Section 3.1.1 in How to Use the NAG Library and its Documentation for further information.

1 Purpose

g22ydf produces labels for the columns of a design matrix, model parameters and a vector of column inclusion flags suitable for use with routines in Chapter G02. Thus allowing for submodels to be fit using the same design matrix.

2 Specification

Fortran Interface

```
Subroutine g22ydf (hform, hxdesc, intcpt, ip, lisx, isx, lplab, plab,      &
                   lvinfo, vinfo, ifail)
Integer, Intent (In)         :: lisx, lplab, lvinfo
Integer, Intent (Inout)      :: ifail
Integer, Intent (Out)        :: ip, isx(lisx), vinfo(lvinfo)
Character (*), Intent (Out) :: intcpt, plab(lplab)
Type (c_ptr), Intent (In)    :: hform, hxdesc
```

3 Description

g22ydf is a utility routine for use with **g22yaf**, **g22ybf** and **g22ycf**. It can be used to construct labels for the columns for an $n \times m_x$ design matrix, X, created by **g22ycf** and return additional input vectors and flags required by a number of NAG Library model fitting routines.

Many of the analysis routines that require a design matrix to be supplied allow submodels to be defined through the use of a vector of ones or zeros indicating whether a column of X should be included or excluded from the analyses (see for example **isx** in **g02daf** or **g02gaf**). This allows nested models to be fit without having to reconstructed the design matrix for each analysis.

Let $\mathcal{M}$ denote a model constructed by **g22yaf**, D a data matrix as described by **g22ybf** and X be the corresponding design matrix constructed by **g22ycf** from $\mathcal{M}$ and D. A different model, $\mathcal{M}_S$ is a submodel of $\mathcal{M}$ if each term in $\mathcal{M}_S$, including the mean effect (intercept term) is also present in $\mathcal{M}$.

If $\mathcal{M}_S$ is a submodel of $\mathcal{M}$, you can fit $\mathcal{M}_S$ to D using a design matrix whose columns are a subset of the columns of X.

4 References

None.

5 Arguments

1: **hform** – Type (c_ptr) *Input*

On entry: a G22 handle to the internal data structure containing a description of the required (sub)model $\mathcal{M}_S$, as returned in **hform** by **g22yaf**.

2: **hxdesc** – Type (c_ptr) *Input*

On entry: a G22 handle to the internal data structure containing a description of the design matrix, D, as returned in **hxdesc** by **g22ycf**.

3: **intcpt** – Character(*) *Output*

On exit: if **intcpt** $=$ 'M', in order to fit the model $\mathcal{M}_S$ to D using X, any analysis routine should include an implicit mean effect (intercept term).

intcpt $=$ 'Z', if $\mathcal{M}_S$ does not include a mean effect or the mean effect has been explicitly included in the design matrix.

4: **ip** – Integer *Output*

On exit: p, the number of parameters in the model specified in **hform**, including the intercept if one is present.

If **lisx** $\neq 0$, if **intcpt** $=$ 'Z', $p = \sum_{i=1}^{m_x} \mathbf{isx}(i)$, otherwise $p = \sum_{i=1}^{m_x} \mathbf{isx}(i) + 1$.

5: **lisx** – Integer *Input*

On entry: length of **isx**.

Constraint: **lisx** $= 0$ or **lisx** $\geq m_x$, where m_x is the number of columns in the design matrix X.

6: **isx**(**lisx**) – Integer array *Output*

On exit: if **lisx** $\neq 0$, an array indicating which columns of the design matrix form the model specified in **hform**.

isx$(j) = 0$
The jth column of the design matrix, X, should not be included in the analysis.

isx$(j) = 1$
The jth column of the design matrix, X, should be included in the analysis.

If **lisx** $= 0$, **isx** is not referenced.

7: **lplab** – Integer *Input*

On entry: the length of **plab**.

As $p \leq m_x + 1$, if labels are required, using **lplab** $= m_x + 1$ will always be sufficient.

Constraint: **lplab** $= 0$ or **lplab** $\geq p$.

8: **plab**(**lplab**) – Character(*) array *Output*

On exit: if **lplab** $\neq 0$, the names associated with the p parameters in the model.

If **intcpt** $=$ 'Z', the labels in **plab** are also the labels for the columns of design matrix used in the analysis.

If **intcpt** $=$ 'M', columns **plab**(2) to **plab**(p) are the corresponding column labels.

If a mean effect is present in M_S, the corresponding label is always in **plab**(1).

If **lplab** $= 0$, **plab** is not referenced.

9: **lvinfo** – Integer *Input*

On entry: the length of **vinfo**.

Let n_T denote the number of terms in M_S, n_{Tt} denote the number of variables in the tth term and m_{xt} denote the number of columns of X corresponding to the tth term. The required size of **vinfo**, denoted a is given by:

$$a = \sum_{t=1}^{n_T} m_{xt}(1 + 3n_{Tt}).$$

If the model includes a mean effect, a should be incremented by one.

The values n_T, n_{Tt} and m_{xt} are not trivial to calculate as they require the formula describing the model to be fully expanded and the contrast / dummy variable encoding to be known. Therefore, if **lisx**, **lplab** or **lvinfo** are too small and **lvinfo** ≥ 3, **ifail** $= 92$ is returned and the required sizes for these arrays are returned in **vinfo**(1), **vinfo**(2) and **vinfo**(3) respectively.

Constraint: **lvinfo** $= 0$ or **lvinfo** $\geq a$.

10: **vinfo**(**lvinfo**) – Integer array *Output*

On exit: if **lvinfo** $\neq 0$, information encoding a description of the parameters in the model.

The encoding information can be extracted as follows:

(i) Set $k = 1$.

(ii) Iterate j from 1 to p.

1. Set $b =$ **vinfo**(k).
2. Increment k.
3. Iterate i from 1 to b.
   (a) Set $v_i =$ **vinfo**(k).
   (b) Set $l_i =$ **vinfo**$(k+1)$.
   (c) Set $c_i =$ **vinfo**$(k+2)$.
   (d) Increment k by 3.
4. The jth model parameter corresponds to the interaction between the b variables held in columns $v_1, v_2, \dots, v_b$ of D. Therefore, $b = 1$ indicates a main effect, $b = 2$ a two-way interaction, etc..

   If $b = 0$, the jth model parameter corresponds to the mean effect.

   If $l_i = 0$, the corresponding variable v_i is binary, ordinal or continuous. Otherwise, l_i is the level for the corresponding variable for model parameter j.

   c_i is a numeric flag indicating the contrast used in the case of a categorical variable. With $c_i = 0$ indicating that dummy variables were used for variable v_i in this term. The remaining six types of contrast; treatment contrasts (with respect to the first and last levels), sum contrasts (with respect to the first and last levels), Helmert contrasts and polynomial contrasts, as described in **g22ycf**, are identified by the integers one to six respectively.

If **lvinfo** $= 0$, **vinfo** is not referenced.

11: **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 $= 11$

hform has not been initialized or is corrupt.

ifail $= 12$

hform is not a G22 handle as generated by **g22yaf**.

ifail $= 13$

A variable name used when creating **hform** is not present in **hxdesc**.
Variable name: $\langle$*value*$\rangle$.

ifail $= 14$

The model and the design matrix are not consistent. The design matrix was constructed in the presence of a mean effect and the model does not include a mean effect.

ifail $= 15$

The model and the design matrix are not consistent. The model includes a term not present in the design matrix.
Term: $\langle$*value*$\rangle$.

ifail $= 16$

The model and the design matrix are not consistent.
Term: $\langle$*value*$\rangle$.
This is likely due to the design matrix being constructed in the presence of either a mean effect or main effect that is not present in the model.

ifail $= 17$

The model and the design matrix are not consistent. The model specifies different contrasts to those used when the design matrix was constructed. The contrasts specified in **hform** will be ignored.

ifail $= 21$

hxdesc has not been initialized or is corrupt.

ifail $= 22$

hxdesc is not a G22 handle as generated by **g22ycf**.

ifail $= 41$

On entry, **lisx** $= \langle$*value*$\rangle$ and $m_x = \langle$*value*$\rangle$.
Constraint: **lisx** $= 0$ or **lisx** $\geq m_x$.

ifail $= 71$

On entry, **lplab** $= \langle$*value*$\rangle$ and $p = \langle$*value*$\rangle$.
Constraint: **lplab** $= 0$ or **lplab** $\geq p$.

ifail $= 81$

On entry, **plab** is too short to hold the parameter labels. Long labels will be truncated.
The longest parameter label is $\langle value \rangle$.

ifail $= 91$

On entry, **lvinfo** is too small.
lvinfo $= \langle value \rangle$.
Constraint: **lvinfo** $= 0$ or **lvinfo** $\geq \langle value \rangle$.

ifail $= 92$

On entry, one or more of **lisx**, **lplab** or **lvinfo** are nonzero, but too small.
Minimum values are zero, or $\langle value \rangle$, $\langle value \rangle$ and $\langle value \rangle$ respectively.
The minimum values are returned in the first three elements of **vinfo**.

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

g22ydf 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 routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

This example performs a linear regression using **g02daf**. The linear regression model is defined via a text string which is parsed using **g22yaf** and the design matrix associated with the model is generated using **g22ycf**. A submodel is then fit using the same design matrix.

Default parameter labels, as returned in **plab** are used for both models. An example of using the information returned in **vinfo** to construct more verbose parameter labels is given in Section 10 in **g22ybf**.

See also the examples for **g22yaf** and **g22ycf**.

10.1 Program Text

```
!   G22YDF Example Program Text
!   Mark 26.1 Release. NAG Copyright 2016.
    Module g22ydfe_mod
!     G22YDF Example Program Module:
!            Parameters and User-defined Routines

!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                           :: fit_lm, read_line
!     .. Parameters ..
      Integer, Parameter, Public       :: nin = 5, nout = 6

    Contains
      Subroutine read_line(v1)
!       Read in a line from NIN and remove any comments

!       .. Scalar Arguments ..
        Character (*), Intent (Out)    :: v1
!       .. Local Scalars ..
        Integer                        :: pend
!       .. Intrinsic Procedures ..
        Intrinsic                      :: adjustl, index
!       .. Executable Statements ..
        Continue

        Read (nin,'(A200)') v1
        pend = index(v1,'::')
        If (pend/=0) Then
          v1 = v1(1:pend-1)
        End If
        v1 = adjustl(v1)

        Return
      End Subroutine read_line
      Subroutine fit_lm(hform,intcpt,nobs,mx,x,isx,ip,y,plab)
!       Perform a multiple linear regression using G02DAF

!       .. Use Statements ..
        Use, Intrinsic                 :: iso_c_binding, Only: c_ptr
        Use nag_library, Only: g02daf, g22znf, nag_wp
!       .. Scalar Arguments ..
        Type (c_ptr), Intent (In)      :: hform
        Integer, Intent (In)           :: ip, mx, nobs
        Character (*), Intent (In)     :: intcpt
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In) :: x(:,:), y(:)
        Integer, Intent (In)           :: isx(:)
        Character (*), Intent (In)     :: plab(:)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: rss, rvalue, tol
        Integer                        :: i, idf, ifail, irank, ivalue, ldq,   &
                                          ldx, lwt, optype
        Logical                        :: svd
        Character (200)                :: cvalue
        Character (1)                  :: weight
!       .. Local Arrays ..
        Real (Kind=nag_wp), Allocatable :: b(:), cov(:), h(:), p(:), q(:,:),   &
                                          res(:), se(:), wk(:), wt(:)
!       .. Intrinsic Procedures ..
        Intrinsic                      :: repeat, size, trim
!       .. Executable Statements ..
        Continue

        ldx = size(x,1)

!       We are assuming un-weighted data
        weight = 'U'
        lwt = 0
        ldq = nobs
        Allocate (b(ip),cov((ip*ip+ip)/2),h(nobs),p(ip*(ip+                  &
          2)),q(ldq,ip+1),res(nobs),se(ip),wk(ip*ip+5*(ip-1)),wt(lwt))

!       Use suggested value for tolerance
        tol = 0.000001E0_nag_wp

!       Fit a regression model
        ifail = 0
        Call g02daf(intcpt,weight,nobs,x,ldx,mx,isx,ip,y,wt,rss,idf,b,se,cov,  &
          res,h,q,ldq,svd,irank,p,tol,wk,ifail)

!       Get the formula for the model being fit
        ifail = 0
        Call g22znf(hform,'Formula',ivalue,rvalue,cvalue,optype,ifail)

!       Display the results
        Write (nout,*) 'Model: ', trim(cvalue)
        Write (nout,*) '                                   Parameter   Standard'
        Write (nout,*) 'Coefficients                       Estimate      Error'
        Write (nout,*) repeat('-',51)
        Do i = 1, ip
          Write (nout,99997) plab(i), b(i), se(i)
        End Do
        Write (nout,*) repeat('-',51)
        Write (nout,99998) 'Residual sum of squares = ', rss
        Write (nout,99999) 'Degrees of freedom      = ', idf

        Return
99999   Format (1X,A,I9)
99998   Format (1X,A,F9.4)
99997   Format (1X,A30,1X,F7.3,5X,F7.3)
      End Subroutine fit_lm
    End Module g22ydfe_mod

    Program g22ydfe
!     G22YDF Example Program Text

!     Mark 25 Release. NAG Copyright 2015.

!     .. Use Statements ..
      Use g22ydfe_mod, Only: fit_lm, nin, nout, read_line
      Use, Intrinsic                   :: iso_c_binding, Only: c_null_ptr,     &
                                          c_ptr
      Use nag_library, Only: g22yaf, g22ybf, g22ycf, g22ydf, g22zaf
      Use nag_precisions, Only: wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Type (c_ptr)                     :: hddesc, hform, hxdesc
      Integer                          :: i, ifail, ip, lddat, ldx, lisx,      &
                                          lplab, lvinfo, lvnames, mx, nobs,    &
                                          nvar, sddat, sdx
      Character (200)                  :: formula
      Character (1)                    :: intcpt
!     .. Local Arrays ..
      Real (Kind=wp), Allocatable      :: dat(:,:), x(:,:), y(:)
      Real (Kind=wp)                   :: tx(0,0)
      Integer, Allocatable             :: isx(:), levels(:), vinfo(:)
      Integer                          :: tisx(0), tvinfo(3)
      Character (50), Allocatable      :: plab(:), vnames(:)
      Character (1)                    :: tplab(0)
!     .. Executable Statements ..

!     .. Executable Statements ..
      Write (nout,*) 'G22YDF Example Program Results'
      Write (nout,*)

      hform = c_null_ptr
      hddesc = c_null_ptr
      hxdesc = c_null_ptr

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

!     Read in size of the data matrix and number of variable labels supplied
      Read (nin,*) nobs, nvar, lvnames

!     Read in number of levels and names for the variables
      Allocate (levels(nvar),vnames(lvnames))
      Read (nin,*) levels(1:nvar)
      If (lvnames>0) Then
        Read (nin,*) vnames(1:lvnames)
      End If

!     Create a description of the data matrix
      ifail = 0
      Call g22ybf(hddesc,nobs,nvar,levels,lvnames,vnames,ifail)

!     Read in the data matrix and response variable
      lddat = nobs
      sddat = nvar
      Allocate (dat(lddat,sddat),y(nobs))
      Read (nin,*)(dat(i,1:nvar),y(i),i=1,nobs)

!     Read in the formula for the full model, remove comments and parse it
      Call read_line(formula)
      ifail = 0
      Call g22yaf(hform,formula,ifail)

!     Calculate the size of the design matrix
      ldx = 0
      sdx = 0
      ifail = 1
      Call g22ycf(hform,hddesc,dat,lddat,sddat,hxdesc,tx,ldx,sdx,mx,ifail)
      If (ifail/=91) Then
!       redisplay the error message
        ifail = 0
        Call g22ycf(hform,hddesc,dat,lddat,sddat,hxdesc,tx,ldx,sdx,mx,ifail)
      End If

!     Generate the design matrix
      ldx = nobs
      sdx = mx
      Allocate (x(ldx,sdx))
      ifail = 0
      Call g22ycf(hform,hddesc,dat,lddat,sddat,hxdesc,x,ldx,sdx,mx,ifail)

!     Get size of output arrays
      lvinfo = 3
      lisx = 0
      lplab = 0
      ifail = 1
      Call g22ydf(hform,hxdesc,intcpt,ip,lisx,tisx,lplab,tplab,lvinfo,tvinfo,  &
        ifail)
      If (ifail/=92) Then
!       redisplay the error message
        ifail = 0
        Call g22ydf(hform,hxdesc,intcpt,ip,lisx,tisx,lplab,tplab,lvinfo,       &
          tvinfo,ifail)
      End If

!     Allocate output arrays
      lisx = tvinfo(1)
      lplab = tvinfo(2)
!     We don't need VINFO as we are using labels in PLAB
      lvinfo = 0
      Allocate (isx(lisx),plab(lplab),vinfo(lvinfo))

!     Get the ISX flag and parameter labels
      ifail = 0
      Call g22ydf(hform,hxdesc,intcpt,ip,lisx,isx,lplab,plab,lvinfo,vinfo,     &
        ifail)

!     Fit the full model and print the results
      Call fit_lm(hform,intcpt,nobs,mx,x,isx,ip,y,plab)

!     Read in the formula for the sub-model, remove comments and parse it
      Call read_line(formula)
      ifail = 0
      Call g22yaf(hform,formula,ifail)

!     Get the ISX flag and parameter labels, as the new model has to be
!     a sub-model of the original one, the output arrays, ISX, PLAB and VINFO
!     can be reused as they will be of sufficient size
      ifail = 0
      Call g22ydf(hform,hxdesc,intcpt,ip,lisx,isx,lplab,plab,lvinfo,vinfo,     &
        ifail)

      Write (nout,*)
      Write (nout,*)

!     Fit the sub-model and print the results
      Call fit_lm(hform,intcpt,nobs,mx,x,isx,ip,y,plab)

!     Clean-up the G22 handles
      ifail = 0
      Call g22zaf(hform,ifail)
      Call g22zaf(hddesc,ifail)
      Call g22zaf(hxdesc,ifail)

      Deallocate (dat,x,y)
      Deallocate (isx,levels,vinfo)
      Deallocate (plab,vnames)

    End Program g22ydfe
```

10.2 Program Data

```
G22YDF Example Program Data
25 3 3                 :: NOBS,NVAR,LVNAMES
3 4 1                  :: LEVELS
Fact1 Fact2 Con        :: VNAMES
3 1 -2.4   1.16
3 4  0.2   4.96
1 4 -1.4  -1.67
2 1 -5.4 -11.80
3 3  0.2   6.03
3 2  1.4  11.70
1 2  6.8  33.34
1 4  6.7  31.97
1 1  5.3  23.93
2 3 -1.3   3.17
3 2 -3.6   1.68
3 2 -0.7   8.01
1 4  5.7  26.14
3 3  2.3  11.04
1 2  3.3  20.32
2 3 -0.5   5.62
1 1 -2.6  -6.21
1 2  3.7  22.45
1 2  0.9  10.93
3 4 -1.1   1.59
2 2  2.1  13.55
1 3  4.6  24.16
2 3  4.6  20.70
1 2  5.1  28.30
1 3  0.9   9.69    :: DAT, Y
(Fact2 + Con + Fact1)^2 :: Model FORMULA
Fact1 + Fact2 + Con     :: Sub-Model FORMULA
```

10.3 Program Results

```
 G22YDF Example Program Results

 Model: FACT2+CON+FACT1+FACT2.CON+FACT2.FACT1+CON.FACT1
                               Parameter   Standard
 Coefficients                   Estimate      Error
 ---------------------------------------------------
 Intercept                        3.709       0.490
 FACT2_2                          4.200       0.801
 FACT2_3                          2.302       0.913
 FACT2_4                          0.174       0.727
 CON                              3.815       0.117
 FACT1_2                         -0.321       1.828
 FACT1_3                          2.377       1.041
 FACT2_2.CON                      0.013       0.184
 FACT2_3.CON                      0.153       0.253
 FACT2_4.CON                      0.257       0.157
 FACT2_2.FACT1_2                  0.029       2.442
 FACT2_2.FACT1_3                 -1.160       1.372
 FACT2_3.FACT1_2                  1.372       2.407
 FACT2_3.FACT1_3                 -2.611       1.435
 FACT2_4.FACT1_2                 -0.000       0.000
 FACT2_4.FACT1_3                 -1.946       1.247
 CON.FACT1_2                     -1.003       0.266
 CON.FACT1_3                     -1.763       0.206
 ---------------------------------------------------
 Residual sum of squares =    3.4371
 Degrees of freedom      =         8


 Model: FACT1+FACT2+CON
                               Parameter   Standard
 Coefficients                   Estimate      Error
 ---------------------------------------------------
 Intercept                        6.222       1.322
 FACT2_2                          3.225       1.599
 FACT2_3                          0.074       1.694
 FACT2_4                         -0.601       1.762
 CON                              3.442       0.196
 FACT1_2                         -0.319       1.561
 FACT1_3                          0.064       1.346
 ---------------------------------------------------
 Residual sum of squares =  105.6953
 Degrees of freedom      =        18
```