

MISSMOD (Version 1.1): LANGUAGE REFERENCE MANUAL

**Diane J. Catellier, Keith E. Muller, Kevin B. Hunter, Ronald G. Helms,
James D. Hosking, David H. Christiansen, and Paul W. Stewart**

No warranty of any kind, with respect to accuracy or function, is given for MISSMOD.

The code and documentation may be shared freely, but may not be sold for profit.

Copyrighted by the authors.

1. INTRODUCTION.....	1
1.1 Background and Documentation	
1.2 Overview of Code	
1.3 User Programming Requirements	
1.4 Overview of IML Modules Invoked by User	
1.5 Examples of the Use of MISSMOD	
1.6 Sequential Dependencies in the Code	
1.7 References	
2. FITML.....	8
2.1 Function	
2.2 Computational Method	
2.3 User Inputs	
2.4 System Input Matrices	
2.5 Options and Defaults	
2.6 User Output (Including Matrices)	
2.7 System Output Matrices	
2.8 Error Conditions	
2.9 Examples	
2.10 Application Notes	
2.11 References	
3. PROCSSCP.....	12
3.1 Function	
3.2 Algorithm (Not applicable)	
3.3 User Supplied Inputs	
3.4 System Input Matrices (None)	
3.5 Options (None)	
3.6 User Oriented Output	
3.7 System Output Matrices	
3.8 Errors	
3.9 Examples	
3.10 Application Notes	
4. SETOPT.....	17
4.1 Function	
4.2 Algorithm (Not Applicable)	
4.3 User Supplied Inputs	
4.4 System Supplied Inputs	

4.5 Options and Defaults	
4.6 User Output	
4.7 System Output Matrices	
4.8 Error Conditions	
4.9 Examples	
4.10 Application Notes (All MISSMOD Options)	
5. TESTGLH Module.....	19
5.1 Purpose	
5.2 Algorithm	
5.3 User Supplied Matrices	
5.4 System Input Matrices	
5.5 Options and Defaults	
5.6 User Output (Including Matrices)	
5.7 System Outputs	
5.8 Errors	
5.9 Examples	
5.10 Application Notes	
5.11 References	
6. UTILITY Modules (Create Contrast Matrices and Names).....	24
6.1 NAMELIST	
6.2 UMEAN	
6.3 UPOLY1	
6.4 UPOLY2	
6.5 UTRENDS	
APPENDIX A: Notation and Formulae.....	26
APPENDIX B: Installation Guide.....	36

1. INTRODUCTION

1.1 Background and Documentation

MISSMOD implements the methods of Catellier and Muller (1998) for fitting a General Linear Multivariate Models (GLMM) with missing data in SAS IML (c). MISSMOD allows the analyst familiar with matrix algebra notation and IML syntax to efficiently compute tests, estimates, and all associated statistics, for any GLMM. Searle (1971) and Timm (1975) provided thorough treatments of univariate and multivariate versions, respectively, of the models of interest. Muller, LaVange, Ramey, and Ramey (Sections 2.1-2.3, 1992) presented a succinct statement of the model, the general linear hypothesis, and associated statistics.

The parameters of the models, estimated using the EM algorithm, are used to compute analogs of the hypothesis (\hat{H}) and error (\hat{E}) sums of squares matrices. In turn, analogs of the Hotelling-Lawley trace, the Pillai-Bartlett trace, Wilks' Lambda, the Geisser-Greenhouse corrected univariate test and corresponding uncorrected are computed. For example, for the Hotelling-Lawley analog, compute $\text{tr}(\hat{H}\hat{E}^{-1})$. Next an adjusted sample size (N_*) is computed to replace N (the total sample size) in calculation of error degrees of freedom for the F approximations commonly used. In all cases N_* equals a function of the number of non-missing pairs of responses. The best choice for N_* depends on the test statistic on the hypothesis of interest. By default, the MISSMOD program uses the N_* choices recommended by Catellier and Muller (1998). However, the user is free to override the default with various other choices for N_* . In all cases the methods reduce to standard ones for complete data.

MISSMOD uses matrix algebra formulation of models and tests. Thus, using MISSMOD requires the ability to define and analyze linear models in matrix notation. For example, MISSMOD does not add or in any way recognize an intercept term in any model. The user must code one if desired, and may choose to test it, if present. A clear advantage of this program over others lies in the fact that all of the computed values remain available to the user in matrices. This allows storing all results in a permanent SAS database (as distinct from the LST file), and exercising complete control of formatting.

In this document notation for matrices and equations is somewhat nonstandard. This was done to facilitate distribution of the documentation for MISSMOD. See Appendix A for a summary of the formulae behind the computations.

1.2 Overview of Code

The MISSMOD source code directory contains 16 files, in five categories.

1. The file MISSMOD.IML contains a block of IML code which causes all of the IML modules in MISSMOD to be defined and made available to the user.

2. Each of a collection of 15 files contains a single module internal to MISSMOD:

DEFOPT.IML, EREXIT.IML, FITMOD.IML, GETCSS.IML, INITL.IML,
MAKESS.IML, NMLIST.IML, PRPARM.IML, PVTTYPE.IML, SCAN.IML,
SETOPT.IML, TSTGLH.IML, UNIREP.IML, UNSCAN.IML, and ZPIVOT.IML.

Module names are file names with an underscore prefixed and post-fixed (e. g., _DEFOPT_).

3. MACROLIB.MAC contains SAS macro statements to simplify invoking MISSMOD.

4. ZUSER.IML contains five IML modules invoked by the user to fit and test models: FITMODEL, GETCORSS, MAKESS, SETOPT, TESTGLH. Each is a dummy module which invokes a corresponding (internal) module. For example, FITML invokes _FITMOD_, contained in the file FITMOD.IML. This approach means that the user specifies all parameters to MISSMOD by specifying matrices with particular names. This simplifies the task and improves the robustness of the code.

5. ZUTIL.IML contains six modules to help generate contrast matrices and labels:

NAMLIST, UMEAN, UPOLY1, UPOLY2, and UTRENDS. These modules are not essential to MISSMOD.

If this documentation is inadequate, the user may examine the MISSMOD source code, provided as ASCII text files. As always, work with a copy and not the original to avoid the risk of corrupting the code.

1.3 User Programming Requirements

A number of design features of MISSMOD help protect the MISSMOD code from the user's code, and vice versa. All internal (to MISSMOD) matrix names, macro variables, and module names begin and end with an underscore (the module `_FITMOD_`, the matrix `_SS_`, etc). Only thirteen (external) matrices provide all user input.

Input External Matrices	Module Using Input
INDVARS, DEPVARs	FITML
INCLUDE, EXCLUDE	GETCORSS
Z, ZNAMES	MAKESS
OPT_ON, OPT_OFF	SETOPT
C, U, THETA0, THETARNM, THETACNM	TESTGLH

Also note that GETCORSS requires that the file `_CORRDS_` exist (in the IML default library, usually WORK).

The programming rules for using MISSMOD may be summarized as follows.

1. Avoid using any name that both begins and ends with an underscore.
2. Avoid defining a module with any of the following names:
FITML, GETCORSS, MAKESS, SETOPT, TESTGLH,
NAMLIST, UMEAN, UPOLY1, UPOLY2, or UTRENDS.
3. Use the following matrices only as MISSMOD input: INDVARS, DEPVARs, INCLUDE, EXCLUDE, Y, Z, ZNAMES, OPT_ON, OPT_OFF, C, U, THETA0, THETARNM, and THETACNM.

1.4 Overview of IML Modules Invoked by User (Defined in ZUSER.IML)

RUN FITML; computes estimates of the parameters of a general linear multivariate model. It requires the MISSMOD system matrices as input. FITML does not use the raw data. Instead, it bases calculations on the SS matrix which allows it (and MISSMOD) to deal with very large files. Any analysis requires the use of FITML at least once.

RUN GETCORSS; retrieves and expands a SAS file created by &PROCSSCP into the matrices of information needed by MISSMOD to fit and test models.

RUN MAKESS; creates a sums of squares and cross products matrix (`_SS_`) and certain other system matrices. These matrices contain all information necessary to perform linear models computations. MAKESS takes as input the raw data matrix, Z. It is used when the Z and Z'Z matrices are small enough to fit comfortably in memory.

Alternately, invoking &PROCSSCP provides an easy to use, extremely time efficient method for the first computing steps of MISSMOD (this replaces the use of MAKESS). &PROCSSCP takes a SAS file as input (with observations as records) and reduces it to a file with variables as records. &PROCSSCP is especially recommended for large numbers of observations. &PROCSSCP creates a SAS file of TYPE=CORR which is input to the GETCORSS module.

RUN SETOPT; is used to set the options for MISSMOD modules. Besides simply turning particular options on and off, the module also provides a convenient way to reset all options to default values. Also if

desired, SETOPT will give the user a list of all valid options for all modules, and/or a list of output matrices, along with row and column label matrices, available upon completion of MISSMOD.

RUN TESTGLH; provides tests of the general linear hypothesis and estimation of the contrasts (secondary parameters). The secondary parameters are linear combinations of the primary parameter matrix. An important feature of MISSMOD, provided in TESTGLH, is easy to tests and estimation for contrasts involving linear combinations of response variables.

1.5 Examples of the Use of MISSMOD

Although this manual is not a tutorial document, examples of MISSMOD analyses demonstrate the ways in which the MISSMOD modules can be combined. The examples, while representative of many of the more common sequences of analyses which will be performed, are by no means exhaustive in demonstrating the ways in which the MISSMOD modules can be combined. Section 1.6 contains a more complete discussion.

```
TITLE1 "EXAMPLE1.SAS--Demonstrate MISSMOD using FITML";
```

```
* Data are from Zeisel, DaCosta, Franklin, et al, A.      *;  
* "Choline, an Essential Nutrient for Humans,"          *;  
* FASEB, 5, 1991, pp. 2093-2098.                      *;
```

```
LIBNAME IN01 "C:\MISSMOD\EXAMPLES\";
```

```
%LET LMDIRECT = C:\MISSMOD\SOURCE\;  
%INCLUDE "&LMDIRECT.MACROLIB.MAC" / NOSOURCE2 ;
```

```
PROC IML WORKSIZE=5000 SYMSIZE=5000;  
&MISSMOD ;
```

```
USE IN01.CHOLINE;
```

```
READ ALL VAR{DAY7 DAY14 DAY21 DAY28 DAY35} INTO Y [COLNAME=DEPVAR];  
READ ALL VAR{GROUP} INTO GROUP;  
READ ALL VAR{DAY0} INTO DAY0;
```

```
CELLMEAN = DESIGN(GROUP);    ** CELL MEAN CODING;  
X = CELLMEAN || DAY0;
```

```
INDVARS={"CONTROL" "DEFICIENT" "DAY0"};  
ZNAMES = DEPVAR || INDVARS ;  
Z = Y||X;  
P = NCOL(Y);
```

```
*-----*;  
* REPEATED MEASURES HYPOTHESIS *;  
*-----*;
```

```
C={-1 1 0};  
RUN UPOLY1((1:5), "TIME", U, UNAME);  
THETARNM={"DEFICIENT-CONTROL"};  
THETACNM={"LIN" "QUAD" "CUB" "QUAR"};
```

```

OPT_ON = {"NBEST" "SSFIT" "MSS"};
RUN SETOPT;
RUN FITML;
RUN TESTGLH;
QUIT;

```

The example demonstrates using MISSMOD to fit a multivariate general linear model, then estimate and test secondary parameters. Both the program and data are included in the EXAMPLES directory provided with MISSMOD. The data being analyzed are the "Payne data" described by Danford, Hughes, and McNee (1960). Data were collected on 45 individuals suffering from cancerous lesions. Of these subjects, 39 were assigned to one of 3 treatment groups and exposed to different amounts of whole-body X-irradiation, while 6 were used as controls (no irradiation). The number of subjects in each group, and the amounts of radiation used were:

Group	Radiation Amount	# Subjects
CONTROL	0	6
LOW	25r-50r	14
MODERATE	75r-100r	15
HIGH	125r-200r	10

After an initial training period the psychomotor test was administered to each subject four times a day. The average of the four trials for each subject each day was taken as the basic score for each day. Immediately following the training, each subject was given four pretreatment tests on the device, to obtain a baseline score, and then the indicated dose of radiation was administered. Observations were made on each of 10 consecutive days following irradiation; for the controls, observations were taken for 10 consecutive days after completion of initial training and baseline measurements. The

purpose of the experiment was to ascertain the extent to which whole-body X-irradiation affected psychomotor performance as measured by the device and whether the effect varied with dose. For this example, only the baseline measurement and measurements on even-numbered days were used.

At each time ($j=2, 4, 6, 8$, or 10), the model has four intercepts, one for each radiation group ($g=1, 2, 3$, or 4), and one covariate, the baseline measurement for a subject ($i(g)=1, 2, \dots, N(g)$):

$$E(Y(i(g), g, j)) = \beta_{00}(g, j) + \beta_{01}(j) \cdot \text{INIT}(i) .$$

Each subject's data is assumed independent of each other subject's. Within a subject, the covariance matrix is allowed to be unstructured, but assumed the same for all subjects (homogeneous between subjects):

$$\text{cov}(Y(i(g), g, j), Y(i(g), g, j')) = \{ \sigma_{jj'} \} \quad (5 \times 5 \text{ for the example}) .$$

For the example, $N(1)=6$, $N(2)=14$, $N(3)=15$, and $N(4)=10$.

In matrix form the model is: $E(Y) = X \cdot \text{BETA} .$

The five columns of Y (45×5) are the scores on days 2, 4, 6, 8, and 10 (labeled SCORE2, SCORE4,..., SCORE10 in the SAS program). The first four columns of X (45×5) are indicator variables for the four groups (corresponding to the variables named CONTROL, LOW, MODERATE, HIGH in the SAS program), and the fifth column of X is the initial score (INIT). BETA (5×5) has the structure:

```

beta0(Ctl, 2) beta0(Ctl, 4) beta0(Ctl, 6) beta0(Ctl, 8) beta0(Ctl, 10)
beta0(Low, 2) beta0(Low, 4) beta0(Low, 6) beta0(Low, 8) beta0(Low, 10)
beta0(Med, 2) beta0(Med, 4) beta0(Med, 6) beta0(Med, 8) beta0(Med, 10)
beta0(Hi , 2) beta0(Hi , 4) beta0(Hi , 6) beta0(Hi , 8) beta0(Hi , 10)
beta1( 2 )   beta1( 4 )   beta1( 6 )   beta1( 8 )   beta1( 10 )

```

MISSMOD stores the parameter estimates in the matrices `_BETA_` and `_SIGMA_` (available after RUN FITML). Secondary parameters and corresponding tests of hypotheses of interest are described in the example program.

1.6 Sequential Dependencies in the Code

The MISSMOD modules must be executed in the sequence demonstrated in the examples. For example, a secondary hypothesis cannot be tested without first fitting a model, nor can a model be fitted without first creating the proper sums of squares and cross products matrix. Consider the list of steps in the following table, with numbering consistent with the examples.

MISSMOD Steps			
Step	Task	&PROCSSCP (Example 1)	MAKES (Example 2)
0	Define raw file of data	LIBNAME or DATA step	LIBNAME or DATA step
1	Define source file path	%LET LMDIRECT=	%LET LMDIRECT=
2	Define macro variables	%INCLUDE MACROLIB	%INCLUDE MACROLIB
*3	Reduce data to SS etc.	&PROCSSCP	-(not done here)-
4	Initialize IML code	PROC IML -; &MISSMOD	PROC IML -; &MISSMOD
5	Change or list options	SETOPT	SETOPT
*6	Read raw data into IML	-(not done here)-	USE -;READ -;CLOSE;
*7	Create SS etc. in IML	GETCORSS	MAKES
8	Fit a GLMM	FITML	FITML
9	Conduct tests, estimation	TESTGLH	TESTGLH

The steps marked with an asterisk vary between the two possible paths (&PROCSSCP and MAKES). Within this required order, the user has great flexibility. Options can be changed by invoking SETOPT at any time after Step 4. Any number of models may be fitted, after Step 3. After a model has been fitted, any number of tests and estimates can be computed with TESTGLH.

In general, moving to back to an earlier step will clear errors detected, at least in terms of matrices and results internal to MISSMOD. None of the code ever creates, modifies, or FREEs any matrix whose name does not start and end with an underscore. Therefore any error in a matrix such as INDVARS or THETA0 will not be fixed. The general philosophy regarding error handling was to be conservative and avoid potentially incorrect calculations. Furthermore a significant amount of code is devoted to detecting errors and reporting sufficient information to allow the user to fix the problem in one try. Overall, the goal was to provide a quick, but graceful "death."

Example 1 uses &PROCSSCP to create a TYPE=CORR file from the raw data as the input. The output is stored as a temporary SAS data set named `_CORRDS_`. For greater efficiency, such as with very large files, or the desire to conduct many sets of MISSMOD calculations on a single set of data, the file may be stored by saying

```
PROC COPY IN=_CORRDS_ OUT=permanentlib.filename;
```

In turn, subsequent runs need to include (before the PROC IML step)

```
PROC COPY IN=permanentlib.filename OUT=_CORRDS_ .
```

This modest inefficiency stems from the inability of the IML READ statement to accept a matrix name as a file name (and limitations associated with using macro's in this context). Note that the VAR statement with &PROCSSCP causes only the listed variables to be used in creating the output data set, and causes them to appear in the order listed.

1.7 References

Muller, K. E., LaVange, L. M., Ramey, S. L., and Ramey, C. T. (1992). Power calculations for general linear multivariate models including repeated measures applications. *Journal of the American Statistical Association*, **87**, 1209-1226.

Searle, S. R. (1971). *Linear Models*. New York: John Wiley.

Timm, N. H. (1975). *Multivariate Analysis*. Monterey, California: Brooks/Cole.

2. FITML

2.1 Function

FITML performs the task of fitting the desired model. The model is defined by the user specifying the independent and dependent variables by name. FITML creates the uncorrected sums of squares and cross products matrix, `_SS_`, and associated parameter values, when `Z`, the raw data matrix, is READ into memory in PROC IML. If no missing values are found, `_SS_` is computed as $Z'Z$. When rows containing missing values are found, the EM algorithm is used to estimate the missing elements of `_Z_` and `_SS_` is computed as `_Z_'*_Z_`. The sweep operation (Goodnight, 1978) is then performed on the sums of square and cross products (`_SS_`) elements corresponding to the independent variables specified by the user. The resulting `_SS_` matrix can then be partitioned into `B`, the model parameter estimates; $(X'X)^{-1}$ the estimated parameter covariance; and `S`, the estimated error covariance matrix. These matrices can be displayed in various forms and significance tests can be performed on the primary parameters as described later in the Options Section.

Subsequent links to FITML to fit different models will first check the current status of the variables and sweep only those not already included in the model. Since the sweep operator is reversible, independent variables to be removed are simply "swept out" of the model by the same operator that "sweeps in" new independent variables. This method results in significant computational savings when fitting many different models, for example, in a stepwise regression procedure. Both full rank (FR) and less than full rank (LTFR) models may be fitted, with full rank being the default option. If a singularity occurs when a full rank model is desired, an error message is displayed and the structure of the linear dependency is printed.

2.2 Computational Method

The model fitting process consists of the following four steps.

2.2.1 Define the Model. The user specifies the names of dependent (Y) and independent (X) variables by defining the character row matrices `DEPVARs` and `INDVARs` respectively. The variable names must match those given to the variables when the `_SS_` matrix was formed. If either `DEPVARs` or `INDVARs` is not specified by the user, FITML will check the current status of all variables in `_SS_`. If any dependent or independent variables are already defined, they are used in the model. If not, an error message is printed and the model cannot be fitted.

2.2.2 Determine which variables need to be swept. The model defined by the user is compared with the current model defined by the status of the variables in `_SS_`, and a row matrix `_SWEEP1_` is created to indicate which variables need to be swept. A value of 1 indicates that an independent variable is to be added to the model, while -1 indicates an independent variable is to be removed, and 0 indicates no change. Note that a change from a dependent (Y) variable to a variable not in the model (W) requires no change in the sweep status.

2.2.3 Perform the Sweep. Since the sweep operator does not prescale and involves division by the pivot (Barr, Goodnight, Sall, and Helwig, 1976), some checking must be done in order to avoid division by zero or a number computationally near to zero. A row matrix, `_DELTA_ = _SUMSQ_ * _EPSL_`, is created to provide tolerances for each variable in `_SS_`. `_SUMSQ_` is a row matrix of the raw sums of squares of all variables in `_SS_`, and `_EPSL_ = 10**(-12)` unless otherwise specified by the user. (See application note 2.10.1).

The diagonal elements (pivots) of all variables to be swept are ranked, and the largest is chosen to be swept first. Before the sweep is performed, two computational accuracy checks must be passed. First the pivot must be greater than the proper element of `_DELTA_`. Second, the resulting sweep must not increase the approximate condition number more than a specified amount. If the pivot fails either of these checks, it is set to zero and the model is assumed to be LTFR. The row and column elements of a zero pivot are also set to zero if they correspond to an unswept variable, or if they are within \pm `_DELTA_` of zero for a swept variable.

After the selected pivot is swept, the matrix values indicating variable status, degrees of freedom and variables left to be swept are all updated. The largest remaining pivot is selected and the process is repeated until desired variables are either swept or have zero pivots. If a zero pivot exists in the swept (X) variables, a note is printed and the conditions of the LTFR option is checked. If a LTFR model is not allowed, an appropriate error condition is set. A zero pivot in the dependent (Y) variables causes a note and sets the error code to a warning level.

2.2.4 Extract Parameters and Perform Tests. The successfully swept `_SS_` can now be partitioned into `_BETA_`, `_XPXINV_` and `_SIGMA_`, which can later be printed and tested as described in the Options Section. It should be noted that the `_XPXINV_` printed in the LTFR case is not identical to the corresponding submatrix of `_SS_`. The $(X'X)^{-}$ generalized inverse must allow non-zero elements in the row and column of a zero pivot in order to preserve the reversibility of the sweeping operation. The inverse printed and used in testing, `_XPXINV_` has these rows and columns all set to zero. It has the desirable property of satisfying the equation `_BETA_ = _XPXINV_ * X'Y`.

2.3 User Inputs

2.3.1 Required

2.3.1.1 `Z`, the data matrix containing both X's and Y's, with columns as variables.

2.3.1.2 `ZNAMES`, a character matrix of one row of names for the columns of `Z` must be defined. It must conform to `Z`.

2.3.1.3 `DEPVARs`. A character matrix of one row of dependent (Y) variable names, which are character values. This vector is required when fitting a new model. If a previous model has been fitted and `DEPVARs` is omitted, all previously defined dependent variables will be included in the model. The order of the variable names in `DEPVARs` does not change the order of the variables in `_SS_`. Thus, when specifying a `U` matrix for `TESTGLH`, use the variable order reflected in `_SS_`, not the order in `DEPVARs`.

2.3.1.4 `INDVARs`. A character matrix with one row of independent (X) variable names. This vector is required when fitting a new model. If a previous model has been fitted and `INDVARs` is omitted, all previously defined independent variables will be included in the model. The order of the variable names in `INDVARs` does not change the order of the variables in `_SS_`. Thus, when specifying a `C` matrix for `TESTGLH`, use the variable order reflected in `_SS_`, not the order in `DEPVARs`.

2.3.2 Optional

2.3.2.1 `_EPSL_`. The constant used in calculating the tolerance value for each variable. The tolerance is the raw sum of squares times `_EPSL_`. Default value is $10^{(-12)}$. This value is computer system dependent, and should reflect the floating point accuracy of the platform.

2.4 System Input Matrices

The following system matrices are required by this module:

`_OPT_` - the option status matrix.

`_ECODE_` - the error condition matrix.

`_SS_` - the sums of squares and cross products matrix.

`_MODNMS_` - character matrix of module names.

`_PARM_`, `_SUMSQ_`, `_VTYPE_`, `_VNAME_` - the `_SS_` descriptors. See SETOPT and MAKESS for descriptions of these matrices.

2.5 Options and Defaults

The option names used to change the options status and the default values are shown below. See SETOPT for specific details.

2.5.1 "PARMIN" (ON) Prints the status of the variables (`_VTYPE_`) and parameter values (`_PARM_`) for input `_SS_`.

2.5.2 "SSIN" (OFF) Prints the input `_SS_`.

2.5.3 "BETA" (ON) Calculates and prints `_BETA_` in the submatrix of `_SS_` corresponding to the matrix of parameter estimates.

2.5.4 "XPXINV" (OFF) Extracts and prints `_XPXINV_`, the submatrix of `_SS_` corresponding to the covariance matrix of the model.

2.5.5 "UNIBETA" (OFF) Calculates and prints model parameters, F tests and $R^{**}(2)$. Prints the number of observations, degrees of freedom, value of the F statistic and its p value, and the multiple correlation $R^{**}(2)$ for each dependent variable. WARNING! This $R^{**}(2)$ value includes the intercept and thus is not the traditional $R^{**}(2)$ value, which can be computed with TESTGLH.

2.5.6 "EXBETA" (ON) Prints the expanded form of `_BETA_`, a column at a time. It gives the standard error, t statistic and two-tailed p value for each parameter. These matrices are also accessible in matrix form as `_BSE_`, `_BT_`, and `_BPVAL_` respectively.

2.5.7 "COVBETA" (OFF) Prints the covariance structure of `_BETA_`.

2.5.7 "SIGMA" (ON) Calculates and prints `_SIGMA_`. `_SIGMA_` is the submatrix of `_SS_` corresponding to the estimated error covariance matrix printed with ROWNAME=`_YNAME_`, COLNAME=`_YNAME_`.

2.5.9 "SCORR" (ON) Correlation matrix of `_SIGMA_`.

2.5.10 "SSSTEP" (OFF) Prints the `_SS_` matrix after each sweep.

2.5.11 "SSFIT" (OFF) Prints the fitted `_SS_` matrix.

2.5.12 "LTFR" (OFF) Allows less than full rank models.

2.5.13 "LINDEP" (ON) Prints the linear dependencies among the independent (X) variables.

2.5.14 "PARMOUT" (ON) Specifies the model that has been fitted. It will give the names of the X and Y variables, the number of observations, the rank of X, and the degrees of freedom.

2.5.15 "MPARM" (OFF) When this option is enabled, the parameters in `_PARM_` will be printed.

2.5.16 "MSS" (OFF) When this option is enabled, `_SS_`, the sum of squares and cross products matrix will be printed. Note that this matrix is NV x NV and can require a number of pages to print.

2.6 User Output

2.6.1 Printed Output. All printed output is described in the Options Section above.

2.6.2 User Accessible Matrices. Any of the optional matrices described in Section 2.5 can be accessed by the user directly after a link to FITML. This must be done before linking any other module since some of the matrices may be changed by another module. The following table gives the matrix name, option and row and column labels where appropriate.

Matrices Available from FITML

Matrix	Option	ROWNAME	COLNAME
<u>_BETA_</u>	BETA,EXBETA	<u>_XNAME_</u>	<u>_YNAME_</u>
<u>_BPVAL_</u>	EXBETA	<u>_XNAME_</u>	<u>_YNAME_</u>
<u>_BSE_</u>	EXBETA	<u>_XNAME_</u>	<u>_YNAME_</u>
<u>_BT_</u>	EXBETA	<u>_XNAME_</u>	<u>_YNAME_</u>
<u>_LINDEP_</u>	LINDEP	<u>_XNAME_</u>	<u>_VNAME_</u>
<u>_PARM1_</u>	UNIBETA	----	<u>_PM1CNM_</u>
<u>_SCORR_</u>	SCORR	<u>_YNAME_</u>	<u>_YNAME_</u>
<u>_SIGMA_</u>	SIGMA	<u>_YNAME_</u>	<u>_YNAME_</u>
<u>_SS_</u>	SSFIT	<u>_VNAME_</u>	<u>_VNAME_</u>
<u>_STAT_</u>	UNIBETA	<u>_YNAME_</u>	<u>_STRNM_</u>
<u>_XPXINV_</u>	XPXINV	<u>_XNAME_</u>	<u>_XNAME_</u>

2.7 System Output Matrices

The FITML module creates or modifies six system matrices.

2.7.1 _SS_ The matrix of sums of squares and cross products.

2.7.2 _PARM_ A row matrix containing N, NCOL(Y), RANK(X), NCOL(W), and TOLR.

2.7.3 _SUMSQ_ A row matrix containing the diagonal elements (sums of squares) of _SS_. This matrix is not modified by subsequent SWEEPS in FITML.

2.7.4 _VNAME_ A row matrix created from ZNAMES which contains the names of the columns (and rows) of _SS_.

2.7.5 _VTYPE_ A row matrix whose i-th element is -1 if the i-th variable is a dependent variable (Y), 0 if the i-th variable is not currently in the model (W) and 1 if the i-th element is an independent variable (X). Since independent and dependent variables are not defined until the FITML module is used, at the end of MAKESS execution none of the variables are currently in the model.

2.7.6 _ECODE_ The vector (column) of system error and warning codes. An element will be set to reflect any problems encountered.

2.8 Error Conditions

5001 (SEVERE) No DEPVARs specified and no dependent (Y) variables exist from a previously fitted model, or one or more variables specified in DEPVARs do not exist.

5002 (SEVERE) No INDVARs specified and no independent (X) variables exist from a previously fitted model, or one or more variables specified in INDVARs do not exist.

5003 (SEVERE) A valid _SS_ matrix has not been created.

5004 (SEVERE) A singularity in the X variables has been found and the less than full rank option (LTFR) was OFF.

5005 (SEVERE) A variable was declared in both INDVARs and DEPVARs.

5006 (SEVERE) Error in sweep status.

5007 (SEVERE) Error degrees of freedom was less than one: sample size is too small.

5010 (WARNING) A singularity exists in the Y variables.

1010 (WARNING) A singularity exists in the X variables. LTFR is allowed and has been fitted.

1110 (WARNING) A LTFR model fitted and a singularity exists in Y variables.

2.9 Examples

*FIT THE MODEL $Y_1 = (X_1 \ X_2 \ X_3) * BETA;$

```
INDVARS = { "X1" "X2" "X3" };
DEPVARs = { "Y1" };
RUN FITML;
```

```
*OMIT X3, MODEL IS NOW Y1 = X1 + X2;
INDVARS = { "X1" "X2" }; RUN FITML;
```

2.10 Application Notes

10.1 Modifying the Tolerance Limits. The tolerance for defining a zero sum of squares can be changed by the user. The current default is `_EPSL_ = 10**(- 12)`. The user can reduce (or increase) the tolerance, thus reducing (or increasing) `_EPSL_`. This may be done by including, for example, the statement

```
_EPSL_ = 10E-16;
```

before linking FITML. The value of `_EPSL_` must reflect the number of digits of accuracy provided by SAS in the particular operating system in which MISSMOD is run.

2.11 References

Goodnight, J. H. The Sweep Operator: Its Importance in Statistical Computing. Proceedings of the Computer Science and Statistics: Eleventh Annual Symposium on the INTERFACE, 1978, pp. 218-229.

Barr, A. J., Goodnight, J. H., Sall, J. P., and Helwig, J. T. A User's Guide to SAS 76, SAS Institute, Inc., Raleigh, NC, 1976.

3. PROCSSCP

3.1 Function

Creates a SSCP matrix from Z. By invoking PROC CORR, &PROCSSCP creates a SAS file containing the uncorrected sums of squares and cross products for a list of variables in the input data set. This data set is READ into memory in PROC IML and the GETCORSS module is used to create `_SS_`, `_PARM_`, `_VNAME_`, `_VTYPE_`, `_SUMSQ_`, and `_ECODE_` from it. At this point, other MISSMOD modules can be used just as if Z, the original data matrix, had been READ into memory and the MAKESS module had been used to create `_SS_`, etc. &PROCSSCP has several important advantages in comparison to the MAKESS module for creating these matrices. First, &PROCSSCP reads only one observation from the input data set into memory at a time; therefore the number of observations in the input data set has no effect on the memory requirements of &PROCSSCP. In contrast, the MAKESS module processes the entire Z matrix in memory. Thus &PROCSSCP is strongly preferred, and sometimes necessary, for the analysis of large files. In addition, &PROCSSCP has other advantages in efficiency and memory in comparison with MAKESS.

3.2 Algorithm (Not applicable)

3.3 User Supplied Inputs

3.3.1 `Input_data_set_name`. The `DATA= ---` parameter on the &PROCSSCP statement specifies the name of the SAS file containing the data matrix Z. If the parameter is omitted, the most recently created file will be used, as for any SAS procedure.

3.3.2 VAR statement. The syntax follows that for any SAS procedure. The statement specifies those variables to be included in the SSCP file. If it is omitted, all numeric variables in the input file will be used. The order of the variables in the VAR statement determines the resulting order of the variables in `_SS_`, and hence in MISSMOD. This characteristic is important in choosing contrast matrices, C, U, THETA0, and associated labels, for TESTGLH.

3.4 System Input Matrices (None)

3.5 Options (None)

3.6 User Oriented Output

Although no printed output is produced by &PROCSSCP, PROC PRINT DATA=_CORRDS_ may be used to list the entire output file.

3.7 System Output

3.7.1 _CORRDS_ is a particular version of a TYPE=CORR file created by PROC CORR. It consists of _SS_, with three rows appended from containing the means, standard deviations, and sample sizes. Listwise deletion (elimination of the entire observation, a row of data) is invoked so that sample size is the same for all variables.

3.8 Errors

ECODE is not set by this macro; if an error occurs the SAS system error switch is set to 1 and no output data set is created.

3.9 Examples

```
* Input dependent variables (Y1-Y5) and Group code (1,2,3);
DATA Z;
```

```
    INFILE RAWDATA;
    INPUT GROUP Y1-Y5;
X1 = (GROUP=1);
X2 = (GROUP=2);
X3 = (GROUP=3);
```

```
&LET LMDIRECT=C:\MISSMOD\SOURCE;
%INCLUDE "&LMDIRECT.MACROLIB.IML" / NOSOURCE2;
```

```
* Create _SS_ and related information as a SAS data set;
&PROCSSCP DATA=Z ;
VAR X1-X3 Y1-Y5;
```

```
* Read the data and analyze it;
PROC IML WORKSIZE=200 SYMSIZE=350;
&MISSMOD;
```

```
* Read the _SS_ file;
RUN GETCORSS;
```

```
* Fit a model;
INDVARS = { "X1" "X2" "X3" };
DEPVARs = { "Y1" "Y2" "Y3" "Y4" "Y5" };
RUN FITML;
```

3.10 Application Notes

&PROCSSCP uses listwise deletion of observations containing one or more missing values. Using any other approach for missing data is strongly discouraged as being of unknown or poor statistical quality.

4. SETOPT

4.1 Function

The SETOPT module is used to change the values of MISSMOD system options. These options are initialized to their default values when the INITIAL module is INCLUDED. The options may be changed at anytime during the MISSMOD analysis.

4.2 Algorithm (Not Applicable)

4.3 User Supplied Inputs

4.3.1 Required: none. If neither OPT_ON nor OPT_OFF is specified, the default options are restored.

4.3.2 Optional

OPT_ON A character row matrix of names of options to be turned on.

OPT_OFF A character row matrix of names of options to be turned off.

4.4 System Supplied Inputs

4.4.1 _ECODE_ The vector of MISSMOD system error conditions.

4.4.2 _MNEMON_ A matrix which contains the option names.

4.4.3 _MODNMS_ Character matrix containing module names.

4.4.4 _OPT_ A binary matrix with one row per MISSMOD module. If the jth element in a module's row is set to one, the corresponding option is on.

4.4.5 _PNAME_ A matrix which contains the option names.

4.5 Options and Defaults

The option name used to change an option's status and the default values are shown.

4.5.1 "AVAILOPT" (OFF) If this option is on, a printout of all available options in MISSMOD will be printed in a format similar to the table in Section 4.10.

4.5.2 "LISTINFO" (OFF) If this option is on then list all of the matrices that are produced by MISSMOD.

4.5.3 "CURROPTS" (OFF) If this option is on, the options currently turned on will be printed.

4.5.4 "NEWOPTS" (OFF) If this option is on, OPT_ON and OPT_OFF specified by the user will be printed.

4.5.5 "COMPRESS" (ON) If this option is on, certain titles, headings, and blank lines will be removed so as to present a more compact printout.

4.5.6 "NOPRINT" (OFF) If this option is on, all output is suppressed.

4.6 User Output

Nothing is printed by default. Setting options listed in 4.5 to ON prints information.

4.7 System Output Matrices

4.7.1 _OPT_ Changed to reflect the user's request.

4.7.2 _ECODE_ Will be set to reflect any errors encountered.

4.8 Error Conditions

9001 (FATAL) Bad option name specified.

4.9 Example

```
PROC IML WORKSIZE=200 SYMSIZE=300;
&MISSMOD ;
OPT_ON = { "LTFR" "ECORR" };
OPT_OFF = { "SIGMA" };
RUN SETOPT;
```

4.10 Application Notes

MISSMOD Options and Default Values

SETOPT /	GETCORSS	FITML	TESTGLH
----------	----------	-------	---------

AVAILOPT	CPARMS	PARMIN	C (ON)
LISTINFO	CSS	SSIN	U (ON)
CURROPTS		BETA (ON)	THETA0 (ON)
NEWOPTS		XPXINV	THETA (ON)
COMPRESS (ON)	UNIBETA	MID	
NOPRINT		CHECK	EXTHETA (ON)
		EXBETA (ON)	MATTHETA
		COVBETA	UNITHETA (ON)
		SIGMA (ON)	MSH
		SCORR (ON)	MSE
		SSSTEP	ECORR (ON)
		SSFIT	HEIVAL
		LTFR	CANVEC
		LINDEP (ON)	CANRSQ (ON)
		PARMOUT (ON)	
			MULTTEST (ON)
			RSQUARED (ON)
			UNIREP (ON)
			UNIRPRNT (ON)
			UNIRWARN
			UNIRFORC

Note: (ON) indicates the default for an option is ON.

The absence of (ON) indicates the default is OFF.

5. TESTGLH

5.1 Function

Estimate and test secondary parameters. The TESTGLH module computes THETA, the estimate of CBU, and performs tests of $H_0: CBU = THETA0$. C, U, and THETA0 are matrices of constants provided by the user. A variety of statistics associated with THETA may also be computed and printed.

5.2 Algorithm

The estimates of the primary parameters have been formed by FITML and are contained in `_SS_`. `_THETA_` is computed as $THETA = C * BETA * U - THETA0$. C (a x q), U (p x b), and THETA0 (a x b) are matrices of constants, with C required and U and THETA0 optional. In most cases, THETA0 is chosen to be the zero matrix. THETA is checked for estimability, using the technique described in Helms, Hosking, and Christiansen (1979). Provided THETA is estimable, the standard errors of each element of THETA, along with t-tests, and p-values for testing each element for departure from 0 are computed. Next, if univariate F tests (for the columns of THETA) or multivariate tests (for THETA as a whole) have been requested, $(C(X'X)^{-1}C')^{-1}$ is computed using the sweep operator (Goodnight, 1978). If multivariate tests have been specified, four test criteria are computed with corresponding significance tests. All are functions of the two b x b matrices H and E defined by: $H = THETA'(C(X'X)^{-1}C')^{-1}THETA$ and $E = U'SU$. The four statistics computed are:

1) Roy's largest root (Roy, 1939, 1945). Charts of the distribution of this statistic are available in many standard linear models texts (e.g. Morrison, 1976; Timm, 1975). Note that alternate definitions of the statistics are available. LINOD reports the largest eigenvalue of $H(H+E)^{-1}$, the largest squared canonical correlation, while SAS GLM reports the largest eigenvalue of $H(E)^{-1}$.

2) The likelihood ratio criterion (Wilks, 1936). Whenever $\text{MIN}(a, b) < 3$, an exact F test is computed based on Rao (1973, p. 555). In other cases, an asymptotic chi-squared approximation (Bartlett, 1947) and an asymptotic F approximation (Rao, 1951) are computed.

???

3) The Hotelling-Lawley trace (Lawley, 1938; Hotelling, 1947, 1951). When $\text{MIN}(a, b) > 1$, an asymptotic F-approximation per McKeon is computed (McKeon, 1974). When the McKeon method is not defined due to small sample size, the asymptotic F-approximation per Pillai (1960) will be computed instead. This substitution will be reported to the user if both the COMPRESS and NOPRINT options are turned off.

4) Pillai's trace (Pillai, 1955). An approximate F statistic for the significance of V is computed using Muller's (1998) approximation.

The formulas for these criteria and approximations are presented in Appendix A. When THETA has only one row or one column, these four criteria are equivalent and the one (exact) F-test is computed.

5.3 User Supplied Matrices

5.3.1 Required

C, a matrix defining linear combinations of the independent variables. Each row of C defines one linear combination of the independent variables (columns of X and rows of B) which will form the corresponding row of THETA. When specifying C, the order of the independent variables must be assumed to be as they were in `_SS_`, not as they were specified in INDVARS.

5.3.2 Optional

U, a matrix defining linear combinations of the dependent variables. U is a $p \times b$ matrix of constants. Each column of U defines one linear combination of the dependent variables (columns of Y and columns of B) which will form the corresponding column of THETA. If U is not specified, a $p \times p$ identity matrix is used for U (i.e. no linear combinations of the dependent variables are formed). When specifying U, the order of the dependent variables must be assumed to be as they were in `_SS_`, not as they were specified in DEPVARs.

THETA0, a matrix of constants to be subtracted from $C*B*U$ to form THETA. THETA0 is an $a \times b$ matrix whose (i, j) element is subtracted from the (i, j) element of $C*B*U$ to form THETA. If THETA0 is not specified, then an $a \times b$ zero matrix is used for THETA0 (e. g., nothing is subtracted from $C*B*U$). When specifying THETA0, the order of the independent and dependent variables must be assumed to be as they were in `_SS_`, not as they were specified in INDVARS and DEPVARs.

THETARNM, a $(1 \times a)$ row matrix of row names for THETA. If it is omitted, then TESTGLH will generate default names.

THETACNM, a $(1 \times b)$ row matrix of column names for THETA. If it is omitted, then TESTGLH will generate default names.

5.4 System Input Matrices

The following system matrices are required by this module:

`_OPT_` - the option status matrix.

`_ECODE_` - the error condition matrix.

`_EPSL_` - value used to test for numeric zero.

`_TGLHP1_` - indicator of first invocation of TESTGLH.

`_SS_` - the sums of squares and cross products matrix.

`_PARAM_`, `_SUMSQ_`, `_VTYPE_`, `_VNAME_` - the descriptors of `_SS_`.

See MAKESS and SETOPT for descriptions of these matrices.

5.5 Options and Defaults

The option names used to change the options status and the default values are shown below. See SETOPT for specific details.

5.5.1 "C" (ON) If this option is on, the C matrix specified by the user will be printed.

5.5.2 "U" (ON) If this option is on, then any user-defined U matrix will be printed. If U has not been defined, then this option has no effect.

5.5.3 "THETA0" (ON) If this option is on, then any user-defined THETA0 matrix will be printed. If THETA0 has not been defined, then this option has no effect.

5.5.4 "THETA" (ON) This option causes `_THETA_`, the estimate of THETA, to be printed in matrix form.

5.5.5 "MID" (OFF) When this option is on, $MID = C(X'X)^{-1}C'$ is printed.

5.5.6 "EXTHETA" (ON) This option, if on, causes THETA to be printed in "expanded form". For each column of THETA, a matrix `_STAT1_` is computed and printed. The first column of `_STAT1_` contains the column of THETA being expanded. The second column contains the estimated standard errors of the elements of the first column. The third column contains t-statistics for testing the corresponding element for departure from zero. The fourth column contains the degrees of freedom for the t-statistics and the fifth column contains two tailed P-values for the tests.

5.5.7 "MATTHETA" (OFF) When this option is on, TESTGLH creates three matrices containing statistics concerning the elements of THETA, which the user can then print or use in further computations. Each of these matrices is the same size as THETA (axb columns). Each element of `_SDTHETA_` is the estimated standard error of the corresponding element of THETA. Each element of `_TTHTA_` is the t-statistic for the corresponding element of THETA. Each element of `_PVTHTA_` contains the two tailed P-value for the corresponding element of `_TTHTA_`.

5.5.8 "UNITHETA" (ON) If this option is on, TESTGLH computes and prints univariate F tests, with associated degrees of freedom and two tailed P-values for each column of THETA.

5.5.9 "MSH" (OFF) This option causes H/a , the mean square hypothesis and cross products matrix to be printed.

5.5.10 "MSE" (OFF) This option causes $E/(N-r)$, the mean square error and cross products matrix to be printed. This is the residual covariance matrix, unbiased estimate, for the transformed scores, $Y*U$.

5.5.11 "ECORR" (ON) This option causes the error (residual) correlation matrix (based on E) to be computed and printed.

5.5.12 "HEIVAL" (OFF) This option prints the eigenvalues of $HE^{*}(-1)$.

5.5.13 "CANVEC" (OFF) If this option is on, the left eigenvectors of $H*INV(E)$ which equal the left eigenvectors of $H*INV(H+E)$ are printed. These eigenvectors, in matrix `_A_`, are normalized so that `_A_ * E * _A_ = I`. The j-th column of `_A_` is the j-th eigenvector, and its elements are proportional to the j-th canonical variate raw weights for the dependent variables.

5.5.14 "CANRSQ" (ON) This option causes the eigenvalues of $H*INV(H+E)$ to be computed and printed. These are the generalized squared canonical correlations.

5.5.15 "MULTTEST" (ON) If this option is specified, the four standard multivariate test criteria defined in Section 5.2 (this chapter) and their associated significance tests are computed and printed.

5.5.16 "RSQUARED" (ON) This option adds to the information printed by the options "EXTHETA", "UNITHETA", and "MULTTEST", and thus one or more of these three options must be on for "RSQUARED" to have any effect. For "EXTHETA", "RSQUARED" causes the simple correlation coefficient associated with each t test to be printed. For "UNITHETA", "RSQUARED" causes the multiple

$R^{**}(2)$ associated with each univariate F test to be printed. For "MULTTEST", it causes a multivariate measure of association to be printed for each of the four multivariate test statistics.

5.5.17 "UNIREP" (ON) If this option is specified, the uncorrected and the Geisser-Greenhouse corrected test for the univariate approach to repeated measures are computed.

5.5.18 "UNIRPRNT" (ON) This option causes the results computed if UNIREP is on to be printed.

5.5.19 "UNIRWARN" (OFF) If this option is on then whenever UNIREP is on and U^*U is not proportional to an identity matrix then a warning is printed.

5.5.20 "UNIRFORC" (OFF) If this option and UNIREP are on then the tests are computed even if U^*U is not proportional to an identity matrix, or $\text{ncol}(U) > \text{nrow}(U)$.

5.6 User Oriented Output

5.6.1 Printed Output

Printed output of TESTGLH is controlled by the options described in Section 5.5.

5.6.2 User Accessible Matrices

Any of the matrices described in Section 5.5 can be accessed by the user directly after a link to TESTGLH. This must be done before linking another module since all matrices are FREED at the beginning of the next module. The following table gives the matrix name, option used to create it, and the chapter section in which it is described.

Matrix	Option	Rowname	Colname	Section
<u>THETA</u>	THETA	<u>THRNM</u>	<u>THCNM</u>	5.2.2, 5.5.4
<u>MID</u>	MID	<u>THRNM</u>	<u>THRNM</u>	5.2.2, 5.5.5
<u>SDTHTA</u>	MATTHETA	<u>THRNM</u>	<u>THCNM</u>	5.5.7
<u>THTTA</u>	MATTHETA	<u>THRNM</u>	<u>THCNM</u>	5.5.7
<u>PVHTTA</u>	MATTHETA	<u>THRNM</u>	<u>THCNM</u>	5.5.7
<u>MSH</u>	MSH	<u>THCNM</u>	<u>THCNM</u>	5.2.2, 5.5.9
<u>MSE</u>	MSE	<u>THCNM</u>	<u>THCNM</u>	5.2.2, 5.5.10
<u>ECORR</u>	ECORR	<u>THCNM</u>	<u>THCNM</u>	5.5.11
<u>HEIVAL</u>	HEIVAL	<u>CANNM</u>	<u>NONM</u>	5.5.12
<u>CANVEC</u>	CANVEC	<u>THCNM</u>	<u>CANNM</u>	5.5.13
<u>CANRSQ</u>	CANRSQ	<u>CANNM</u>	<u>NONM</u>	5.5.14
<u>FSTATS</u>	UNITHETA	<u>THCNM</u>	<u>FSTRNM</u>	5.5.6, 5.5.17
<u>STMAT1</u>	MULTTEST	<u>STMRNM</u>	<u>STMCMN</u>	5.5.16, 5.5.17
<u>TPARM1</u>	MULTTEST	<u>NONM</u>	<u>TPCNM1</u>	5.5.16
<u>URESUL</u>	UNIREP	<u>UCOLNM</u>	<u>UROWNM</u>	5.5.18-5.5.21

Note that if THETARNM exists then THTRNM = THETARNM, and if THETACNM exists then THTCNM = THETACNM.

5.7 System Outputs

TESTGLH may modify ECODE, the error condition matrix.

5.8 Errors

5001 (SEVERE) Last invocation of FITML created an error of level 5000 or more.

3002 (TEMPORARY) C is not defined.

3003 (TEMPORARY) C and B do not conform.

3004 (TEMPORARY) U is specified and does not conform to B.

3005 (TEMPORARY) THETA0 is specified and does not conform to $C*B*U$.

3006 (TEMPORARY) THETA is not estimable.

3007 (TEMPORARY) TESTGLH was invoked without any model having been fitted.

3008 (TEMPORARY) TESTGLH is used, but no dependent variables have been specified in `_SS_`.

3009 (TEMPORARY) THETACNM and THETA do not conform.

3010 (TEMPORARY) THETARNM and THETA do not conform.

3011 (TEMPORARY) Denominator degrees of freedom are less than 1 for at least one of the four multivariate tests.

2001 (WARNING) THETA is singular, therefore estimable, but no multivariate tests can be performed.

5.9 Examples

See Examples 1 and 2 in Section 1.5.

5.10 Application Notes (None)

5.11 References

Bartlett, M. S. Multivariate analysis. *Journal of the Royal Statistical Society Supplement*, 1947, 9(B), 176-197.

Goodnight, J. H. The Sweep Operator: its Importance in Statistical Computing. *Proceedings of the Computer Science and Statistics: Eleventh Annual Symposium on the INTERFACE*, 1978, pp. 218-229.

Helms, R. W., Christiansen, D. H., and Hosking, J. H. The LINMOD Algorithms: Their Rhyme and Reason. The American Statistical Association, 1979, *Proceedings of Statistical Computing Section*.

Hotelling, H. Multivariate Quality Control, Illustrated by the Air Testing of Sample Bombsights. In C. Eisenhart et al (eds.). *Selected Techniques of Statistical Analysis*. McGraw-Hill, New York, 1947, 111-184.

Hotelling, H. A Generalized T Test and Measure of Multivariate Dispersion. *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 1951, 23-41.

Lawley, D. N. A Generalization of Fisher's Z-test. *Biometrika*, 1939, 30, 180-187.

McKeon, J. J. F Approximations to the Distribution of Hotelling's T_o^2 . *Biometrika*, 1974, 61, 381-383.

Muller, K. E. A New F Approximation for the Pillai-Bartlett Trace Under H_o . *Journal of Computational and Graphical Statistics*, 1998, ???

Pillai, K. C. S. Some New Test Criteria in Multivariate Analysis. *Annals of Mathematical Statistics*, 1955, 26, 117-121.

Pillai, K. C. S. *Statistical Tables for Tests of Multivariate Hypotheses*. Manila, University of the Philippines Statistical Center, 1960.

Rao, C. R. An Asymptotic Expansion of the Distribution of Wilks' Criterion. *Bulletin of the Institute of International Statistics*, 1951, XXXIII(c), 177-180.

Roy, S. N. p -Statistics or Some Generalizations in Analysis of Variance Appropriate to Multivariate Problems. *Sankhya*, 1939, 4, 381-396.

Roy, S. N. The Individual Sampling Distribution of the Maximum, the Minimum, and Any Intermediate One of the p-Statistics on the Null Hypothesis. *Sankhya*, 1945, 7, 133-158.

Rao, C. R. *Linear Statistical Inference and its Applications* (2nd ed.). Wiley, New York, 1973, 555.

Timm, N. H. *Multivariate Analysis with Applications in Education and Psychology*, Wadsworth, Belmont, California, 1975.

Wilks, S. S. Certain Generalizations in the Analysis of Variance. *Biometrika*, 1932, 24, 471-494.

6. UTILITY IML MODULES

This section describes a set of utility modules designed to aid the use of MISSMOD, although not a part of MISSMOD. Only limited error checking is done by these modules. Note that the U matrices produced by the modules can be used as C matrices (for cell mean coding schemes) simply by transposing the results.

6.1 NAMELIST (A Function)

Usage: yourlist = NAMELIST(STEM,LOW,HIGH,BY);

This function generates a row of names, STEMlow to STEMhigh, by..., a character matrix. STEM is a character string (1x1). Require $0 \leq \text{LOW} \leq \text{HIGH}$, and $1 \leq \text{BY}$, integers (1x1). See the MAKESS example in 4.9 for an example use. Example 1 in Section 1.5, includes the following code:

```
DEPVAR=NAMELIST("SCORE",2,10,2);
```

This statement creates the following matrix:

```
*DEPVAR = { "SCORE2" "SCORE4" "SCORE6" "SCORE8" "SCORE10" };
```

6.2 UMEAN (A Function)

Usage: UAVE = UMEAN(P); This function returns J(P,1,1/P), which provides a column (vector) to use as an averaging U matrix. UAVE=UMEAN(4); yields the column [.25 .25 .25 .25]';

6.3 UPOLY1 (A Module)

This module produces a U matrix and associated names for a test of trends. The names can be used as THETACNM. Centering and scaling are used to increase numerical accuracy.

Usage: RUN UPOLY1(VALUES, NAME, U, NMOUT);

Inputs. VALUES, a set of numeric treatment levels (values), forming a matrix with one row or column.

NAME, a character string providing stem of names for trends.

Outputs. U, a matrix with columns orthonormal polynomial coefficients (excludes zero order).

NMOUT, a character matrix with one row of names.

Example 1 (Section 1.5) uses this module.

6.4 UPOLY2 (A Module)

This module produces a U matrices and associated names for a tests of trends and interaction for a two within-subject factors. The names can be used as THETACNM. Assume Factor 1, with levels VALUES1, varies slowly,

and that Factor 2, with levels VALUES2, varies rapidly.

Centering and scaling are used to increase numerical accuracy.

Usage: RUN UPOLY2(VALUES1,NAME1, VALUES2,NAME2,
U1,NMOUT1, U2,NMOUT2, U12,NMOUT12);

Inputs VALUES1=1st set of numeric treatment levels/values
NAME1 = 1st character string providing stem of names;
VALUES2=2nd set of numeric treatment levels/values

```

NAME2 = 2nd character string providing stem of names;
Outputs U1      =matrix of orthonormal polynomial coefficient
                columns for 1st factor (excludes zero order)
NMOUT1 =list (1 row matrix) of 1st factor names (charactr);
U2      =matrix of orthonormal polynomial coefficient
                columns for 2nd factor (excludes zero order)
NMOUT2 =list (1 row matrix) of 2nd factor names (charactr);
U12     =matrix of orthonormal polynomial coefficient
                columns for interaction (excludes zero order)
NMOUT12=list (1 row matrix) of interaction names(charactr);

```

Example 1 (Section 1.5) uses UPOLY1, which treats the special case of one factor.

6.5 UTRENDS (A Function)

Create polynomial trends matrix (columnwise), excluding the zero order. Centering and scaling is used to help numerical accuracy.

Usage: UPOLY = UTREND(VALUE);

For example, TIME= {2 4 6}; and UTIME=UTREND(TIME); yields the 3 x 2 matrix

```

[ -1/sqrt(2)    1/sqrt(6) ]
[   0          -2/sqrt(6) ]
[  1/sqrt(2)    1/sqrt(6) ]

```

APPENDIX A: MISSMOD Notation

Tables A.1 - A.8 summarize the notation used throughout the MISSMOD documentation and, to the extent possible, notation used in the MISSMOD code as well. The matrix `_SS_` is a dynamic matrix which changes throughout computations. The structure of this matrix at various places in the computations is discussed elsewhere in the MISSMOD documentation.

Table A.1 Notation for Data Matrices

Description	Mathematical Notation	MISSMOD Notation
Data matrix	$Z \text{ (N x NV)}$	<code>Z</code>
Number of observations (rows of <code>Z</code>)	N	<code>N</code>
Number of variables (columns) in <code>Z</code>	NV	<code>NV</code>
Matrix of dependent variable values [subset of the columns (variables) of <code>Z</code>]	$Y \text{ (N x p)}$	<code>Y</code>
Number of dependent variables (columns of <code>Y</code>)	p	<code>p</code>
Matrix of independent variables values [subset of columns (variables) of <code>Z</code>]	$X \text{ (N x q)}$	<code>X</code>
Number of independent variables (columns of <code>X</code>)	q	<code>q</code>
Rank(<code>X</code>)	$r = \text{Rank}(X)$	<code>r</code>
Matrix of values of "other" variables which are in <code>Z</code> but are not in <code>X</code> or <code>Y</code>	W	<code>W</code>
Number of variables (columns) of <code>W</code>	$NV - (p+q)$	$NV - (p+q)$
Matrix of (raw) sums of squares and cross products	$Z'Z = \begin{bmatrix} X'X & X'Y & X'W \\ Y'X & Y'Y & Y'W \\ W'X & W'Y & W'W \end{bmatrix}$	<code>SSCP, _SS_</code>

Table A.2 Linear Model Parameters and Assumptions

Description	Mathematical Notation	MISSMOD Notation
Model Equation	$E(Y) = X*BETA$	$E(Y)=X*BETA$
Matrix of primary parameters ("regression coefficients")	$BETA \ (q \times p)$	$BETA \ (q \times p)$
Covariance assumptions		
(a) Univariate model ($p = 1$)	$V(Y)=\sigma^{**}(2) I (N)$	$V(Y)=\sigma^{**}(2) I (N)$
(b) Multivariate model ($p > 1$)	$V[Y(,j)]=SIGMA(j,j) I (N)$ $Cov[Y(,j),Y(,k)] =$ $SIGMA(j,k) I (N)$	$SIGMA(J,J)*I(N)$ $SIGMA(J,K)*I(J)$
	$V[Y(i,)] = SIGMA \ (p \times p)$	$SIGMA$
	$Cov[Y(i,),Y(k,)] = 0, i=k$	
Secondary parameters (See Note 1)		
(a) Univariate model	$C \ B - THETA0$ $ax1$	$THETA=C*BETA-THETA0$
(b) Multivariate model	$C \ B \ U - THETA0$ axb	$THETA=C*BETA*U-THETA0$
where the matrices are		
C, independent variable "contrast" matrix	$C \ (axq), Rank(C) = a \leq q$	C
U, dependent variable "contrast" matrix	$U \ (p \times b), Rank(U) = b \leq p$	U

Note 1. Secondary parameters from Less Than Full Rank models are assumed to be estimable and/or "testable"; MISSMOD checks these assumptions.

Table A.3 Linear Model Parameter Estimates and Related Matrices

Parameter or Matrix	Mathematical Formula for Estimator, or Description	MISSMOD Notation
BETA	$B = (X'X)^{-1}X'Y$ [FR]	B
	$(X'X)^{-1}X'Y$ [LTER]	
SIGMA (multivariate model)	$S = \frac{(Y - XB)'(Y - XB)}{N - \text{Rank}(X)}$	S
sigma** (2)	$s^{**}(2) = \frac{(Y - XB)'(Y - XB)}{N - \text{Rank}(X)}$	s** (2)
THETA	CB - THETA0 (univariate) CBU - THETA0 (multivariate)	T

Table A.4 Some Variances and Covariances of Estimators in General Linear Models

a. Univariate Model (B is $q \times 1$, T is $a \times 1$)

$$V(B) = \sigma^{**}(2) (X'X)^{**}(-1)$$

$$V(B(i,)) = \sigma^{**}(2) [(i,i) - \text{elt. of } (X'X)^{**}(-1)]$$

$$V(T) = \sigma^{**}(2) C(X'X)^{**}(-1)C'$$

$$V(T(i,)) = \sigma^{**}(2) [(i,i) - \text{elt. of } C(X'X)^{**}(-1)C']$$

b. Multivariate Model (B is $q \times p$, T is $a \times b$)

$$V(B(i,j)) = \text{SIGMA}(j,j) * [(i,i) - \text{elt. of } (X'X)^{**}(-1)]$$

$$V[B(i,)] = \text{SIGMA} * [(i,i) - \text{elt. of } (X'X)^{**}(-1)]$$

$$V[B(,j)] = \text{SIGMA}(j,j) * (X'X)^{**}(-1)$$

$$\text{Cov}[B(i,j), B(i',j')] = \text{SIGMA}(j,j') [(i,i') - \text{elt. of } (X'X)^{**}(-1)]$$

$$V(T(i,j)) = [(j,j) - \text{elt. of } U' \text{SIGMA } U] [(i,i) - \text{elt. of } C(X'X)^{**}(-1)C']$$

$$V[T(i,)] = U' \text{SIGMA } U [(i,i) - \text{elt. of } C(X'X)^{**}(-1)C']$$

$$V[T(,j)] = [(j,j) - \text{elt. of } U' \text{SIGMA } U] C(X'X)^{**}(-1)C'$$

$$\text{Cov}[T(i,j), T(i',j')] = [(j,j') - \text{elt. of } U' \text{SIGMA } U] [(i,i') - \text{elt. of } C(X'X)^{**}(-1)C']$$

Notes

1. For a Less Than Full Rank (LTFR) model, if: (a) one agrees to use a specific, deterministic generalized inverse, $(X'X)^{**}(-)$ for $(X'X)$, and (b) one agrees to accept $B = (X'X)^{**}(-)X'Y$ as the unique least squares estimator under consideration, then $V(B)$ is a meaningful expression and one can obtain variances and covariances of B and its elements from the expressions in the table by replacing $(X'X)^{**}(-1)$ by $(X'X)^{**}(-)$.
2. Unbiased estimators of all quantities in the table may be obtained by replacing parameters by estimators.
3. The "usual" standard errors of estimates are obtained by taking square roots of the variance estimates described in Note 2. These standard errors are biased estimators of the corresponding parameters, the square roots of the variances shown in the table.

Table A.5 Tests of Hypotheses in the General Linear Univariate Model

Hypotheses	Test Statistics	Degrees of Freedom
1. $H_0: \text{BETA} = 0$ $H_a: \text{BETA} \neq 0$	$F = B'X'XB / (rs^{**}(2))$	$r, N-r$ Note: $r = \text{Rk}(X)$
2. $H_0: \text{BETA}(i,)=0$ $H_a: \text{BETA}(i,) \neq 0, \text{ or}$ $H_b: \text{BETA}(i,)>0, \text{ or}$ $H_c: \text{BETA}(i,)<0$	$t = B(i,)\{ s^{**}(2)[(i,i)^{**}(-)\text{elt.of}$ $(X'X)^{**}(-)]^{**}(-1/2)\}$	$N-r$
3. $\text{THETA} = CB - \text{THETA0}$ ax1 $H_0: \text{THETA} = 0$ $H_a: \text{THETA} \neq 0$	$F=T'\{ [C(X'X)^{**}(-)C']^{**}(-1)\}T/(as^{**}(2))$	$N-r$
4. Special Case: THETA a scalar ($a=1$) $H_0: \text{THETA} = 0$ $H_a: \text{THETA} \neq 0, \text{ or}$ $H_b: \text{THETA} > 0, \text{ or}$ $H_c: \text{THETA} < 0$	$t=T[s^{**}(2)C(X'X)^{**}(-)C']^{**}(-1/2)$	$N-r$

Note: In 3 and 4 it is assumed $a = \text{Rank}(C)$ and THETA is estimable.

Table A.6 Tests of Hypotheses in the General Linear Multivariate Model
I. Fundamental Notation for Tests of $H_0: \text{THETA} = 0$

Mathematical Notation	MISSMOD Notation	TESTGLH Code Notation	Description
p	<code>_PARM_(1,2)</code>	# of dependent	var's (cols.) in Y
q	q		# of independent var's (cols of X)
N	N	<code>_PARM_(1,1)</code>	# of observations (# of rows of X, Y)
$r = \text{Rank}(X)$	r	<code>_PARM_(1,3)</code>	$r = \text{Rank}(X)$, $r \leq q$, $r < N$
$df = N - r$	df	<code>_DF_</code>	Error Degrees of freedm
THETA (axb)	THETA		CBU-THETA0 is a secondary parameter
a	a	<code>_RQ1_</code>	# of rows of THETA
b	b	<code>_RP1_</code>	# of columns of THETA
$H_0: \text{THETA}=0$	$H_0: \text{THETA}=0$		General Linear Multi- variate Hypothesis
H (bxb)	H	<code>_MSH_*(N-r)</code>	$\text{THETA}'[C(X'X)^{-1}C']\text{THETA}$ is the SSCP matrix for hypothesis
E (bxb)	E	<code>_MSE_*(N-r)</code>	$E = U'SU \cdot df$, the SSCP matrix for error
$\text{EVAL}(\text{HE}^{**}(-1))$	$\text{EVAL}(\text{HE}^{**}(-1))$	<code>_HEIVAL_</code>	Eigenvalues of $\text{HE}^{**}(-1)$
$\text{EVEC}(\text{HE}^{**}(-1))$	$\text{EVEC}(\text{HE}^{**}(-1))$	<code>_CANVEC_</code>	Eigenvectors of $\text{HE}^{**}(-1)$
$\text{EVAL}(H(H+E)^{**}(-1))$	$\text{EVAL}(H(H+E)^{**}(-1))$	<code>_CANRSQ_</code>	Eigenvalues of $H(H+E)^{**}(-1)$
$s = \text{Min}(a,b)$	s	<code>_RS1_</code>	$s = \text{Rank}(H) = \text{Rank}(\text{HE}^{**}(-1)) =$ max # non-zero eigenvalues
$m = (a-b-1)/2$	m	<code>_RM2_</code>	standard parameter of multivariate test stat's
$n = (df-b-1)/2$	n	<code>_RN1_</code>	standard parameter of multivariate test stat's

Table A.7 Tests of Hypotheses in the General Linear Multivariate Model
II. Special Cases which Reduce to Univariate Tests

Hypotheses	Test Statistic	Degrees of Freedom
1. $H_0: \text{BETA}(,j)=0$ $H_a: \text{BETA}(,j) \neq 0$	$F = \frac{[B(,j)]'X'X[B(,j)]}{r \cdot s(j,j)}$	$r, N-r$
2. $H_0: \text{BETA}(i,j)=0$, vs. $H_a: \text{BETA}(i,j) \neq 0$, or $H_b: \text{BETA}(i,j) > 0$, or $H_c: \text{BETA}(i,j) < 0$	$t = B(i,j) \{ s(j,j) \}^{-1/2} \cdot [(i,i) - \text{elt. of } (X'X)^{-1}]^{-1/2}$	$N-r$
3. $\text{THETA} = \text{CBU} - \text{THETA}_0$ (axb)		
3.1 $b=1$ (0 has one column), $a>1$		
$H_0: \text{THETA} = 0$	$F = T' \{ [C(X'X)^{-1}C]'T / (aU'SU) \}$	$a, N-r$
$H_a: \text{THETA} \neq 0$	where $T = \text{CBU} - \text{THETA}_0$	
3.2 $b=1, a=1$ (THETA is a scalar)		
$H_0: \text{THETA} = 0$, vs.		
$H_a: \text{THETA} \neq 0$, or $H_b: \text{THETA} > 0$, or $H_c: \text{THETA} < 0$	$t = T[U'SU C(X'X)^{-1}C']^{-1/2}$ (see 3.1 for definition of T)	$N-r$
4. $a=1, b > 1$	$F = \frac{df+1-b}{b} \cdot \frac{1 - \text{Tr}[(H+E)^{-1}]}{\text{Tr}[H(H+E)^{-1}]}$	$b, 2(df+1-b)$
$H_0: \text{THETA} = 0$		
$H_a: \text{THETA} \neq 0$		

Note: Case 4 is actually not a "special case which reduces to a univariate test"; however, in Case 4 all the "usual" multivariate test statistics are equivalent and reduce essentially to $\text{Tr}(HE^{-1})$. Moreover, the indicated F statistic has, under H_0 , a central F distribution with the indicated degrees of freedom.

Table A.8 Tests of Hypotheses in the General Linear Multivariate Model
III. $H_0: \text{THETA} = 0$ vs. $H_a: \text{THETA} \neq 0$

A. Fundamental Statistics:

A.1. Roy's Largest Root

$$\begin{aligned} \text{RLR}(s) &= \text{Max}[\text{EVAL}(H(H+E)^{**}(-1))] \\ &= \text{Max}[\text{EVAL}(HE^{**}(-1))] / [1 + \text{Max}(\text{EVAL}(HE^{**}(-1)))] \end{aligned}$$

See Heck's charts for many critical values.

A.2. Wilks' Likelihood Ratio

$$W = |H|/|H+E| \text{ with parameters } s, m, n$$

See Note 1.

A.3. Hotelling-Lawley Trace

$$\begin{aligned} U(s) &= \text{Tr}[HE^{**}(-1)] \text{ with parameters } s, m, n \\ &= \text{sum of the } s \text{ non-zero eigenvalues of } HE^{**}(-1) \end{aligned}$$

A.4. Pillai's Trace

$$V(s) = \text{Tr}[H(H+E)^{**}(-1)]$$

B. Exact Distributions of W for Special Cases

B.1. $a = 2, b > 1$ (THETA has 2 rows)

$$F = \frac{1 - W^{**}(1/2)}{W^{**}(1/2)} * \frac{df + 1 - b}{b} \quad \text{with } 2b, 2(df+1-b) \text{ degrees of freedom}$$

B.2. $b = 2, a \geq 1$ (THETA has 2 columns)

$$F = \frac{1 - W^{**}(1/2)}{W^{**}(1/2)} * \frac{df - 1}{a} \quad \text{with } 2a \text{ and } 2(df-1) \text{ degrees of freedom}$$

B.3. The cases $a = 1, b \geq 1$, and $b = 1, a \geq 1$ reduce to univariate tests; see Table A.7.

C. Approximation for the Distribution of W for the Cases

$$a > 2, b > 2$$

Rao's F Approximation

$$\text{Let: } R_m = df - (b - a + 1)/2$$

$$R_s = [(a^{**}(2)b^{**}(2) - 4) / (a^{**}(2) + b^{**}(2) - 5)]^{**}(1/2)$$

$$R_l = (ab - 2) / 4$$

$$F = \frac{1 - W^{**}(1/R_s)}{W^{**}(1/R_s)} * \frac{R_m R_s - 2 R_l}{ab}$$

Under H_0 , F is approximately distributed as a central F variable with ab and $(R_m R_s - 2 R_l)$ degrees of freedom.

Table A.8 (continued)

D. Approximations to the Distributions of Trace Statistics

D.1. Hotelling-Lawley Trace - McKeon's Method

$$F = (df2) * U(s) / (df1),$$

$$\text{where } \gamma = \frac{ab(df2-2)}{df2(df-b-1)}$$

$$df1 = ab$$

$$df2 = 4 + (ab + 2) \left[\frac{(df)^2 - (df)(2b+3) + b(b+3)}{(df)(a+b+1) - (a+2b+b^2-1)} \right]$$

Under H_0 , F is approximately distributed as a central F with $df1$ and $df2$ degrees of freedom.

D.2. Hotelling-Lawley Trace - Pillai's Method

$$F = (df2) * U(s) / (df1) * s,$$

$$\text{where } df1 = s(2m + s + 1) = ab$$

$$df2 = 2(sn + 1)$$

Under H_0 , F is approximately distributed as a central F with $df1$ and $df2$ degrees of freedom.

D.3. Pillai's Trace - Muller's Method

$$F = (df2) * V(s) / [(df1) * (s - V(s))]$$

$$\text{where } \gamma = s$$

$$df1 = \frac{ab}{s(df+a)} \left[\frac{s(df+s-b)(df+a+2)(df+a-1)}{df(df+a-b)} - 2 \right]$$

$$df2 = \frac{s(df+s-b)}{(df+a)} \left[\frac{s(df+s-b)(df+a+2)(df+a-1)}{df(df+a-b)} - 2 \right]$$

Under H_0 , F is approximately distributed as a central F with $df1$ and $df2$ degrees of freedom.

APPENDIX B: INSTALLATION GUIDE

1. Copy the source code to a directory accessible to SAS jobs. The directory structure of the distribution zip file is recommended. The statement `%LET LMDIRECT = ---` ; must reflect the location where MISSMOD source code has been stored.
2. Optionally modify DEFOPT.IML to change the settings of default options. The file INITL.IML contains the matrix of option names. This implies which location in `_OPT_` should be reset from 0 to 1 or 1 to 0.
3. Optionally the line of code `_EPSL_=1E-12;` should be modified to reflect the number of digits of accuracy of the computer and operating system in use. This value corresponds to the smallest value distinguishable by the machine from zero. Note that this reflects the number of digits of accuracy, not the most extreme exponent. The current value assumes a maximum of 14.2 digits of accuracy (without rounding).