MDSX PROJECT: ATTRIBUTION

NOTE ON OWNERSHIP, ATTRIBUTION OF MDSX PROGRAMS AND DOCUMENTATION

The attribution note in the original version of MDS(X) (up to SV3.2) refers to the University of Edinburgh and University College, Cardiff, which were the academic affiliations of the then Principal Investigators of the MDS(X) Project.

MDS(X) is now operated by a Project Team, consisting of:

MDS(X) Project Team:

Professor A.P.M. Coxon	(University of Essex)
Dr A. Brier	(University of Southampton)
Professor C.L. Jones	(University of Toronto)
Mr D.T. Muxworthy	(University of Edinburgh)
Dr W. Otto	(University of Zurich)
Dr P.M. Davies	
Dr S.K. Tagg	(Strathclyde University)
Dr W van Schuur	(Groningen University)

MDS(X) operates as a without-profit academic freeware group, and is self-supporting.

Use of the programs should be attributed as:

MDSX Project, University of Essex

Other references: Citation of material from:

1. The User's Guide to Multidimensional Scaling **(TUG):**
   Coxon, A.P.M. (1982) The User's Guide to Multidimensional Scaling **, London: Heinemann**
2. MDS(X) User Manual **(TUM)**
   The MDS(X) Series of Multidimensional Scaling **Programs (1981). Edinburgh: Inter-University/Research Councils Series, Report no 51**
3. Key Texts in Multidimensional Scaling **(KTMDS)**
   Davies, P.M. & A.P.M. Coxon (1982), eds Key Texts in Multidimensional Scaling **London: Heinemann**

Conditions:

No author, MDSX Project Team member, or any other person acting on behalf of them assumes any liability for consequences resulting from the use of these programs, or any information, apparatus or method of processing disclosed in this series of publications. This document may not be printed in whole or in part without permission of the Team.

MDSX Co-ordinator:

Comments and enquiries on the MDSX Library and Technical Documentation should be addressed in the first instance to the Project Team Co-ordinator:

Professor A.P.M. Coxon
MDSX Project
Department of Sociology / Health & Social Services Institute
University of Essex
Wivenhoe Park
Colchester CO4 3SQ
Tel: +1206 872622 (direct); Fax: +1206 873410
Personal e-mail: apmc@essex.ac.uk

Project e-mail: mdsx@essex.ac.uk
Project URL: http://newmdsx.com

RUNNING MDS(X) PROGRAMS in MS-DOS

1. It is often easier to run MDS(X) programs entirely within the MS-DOS **C: prompt** window, and use utilities like WORDPAD and EDIT to create the input file. This is because editing symbols can cause problems to FORTRAN READ instructions, and it is therefore best to ensure input files are in ASCII format (e.g. .txt files).
2. The column position is crucial for reading data in MDS(X) programs (especially the CONTROL field beginning in col. 1 and the OPERAND field beginning in col. 16). It is therefore best to use an equal-spaced font, such as COURIER, in order to line up file contents correctly.
3. When running a program, it is sensible to ensure that the program .exe file and the prepared input data-file are in the same directory (if not, use the full path-name of the relevant file).
4.

DATA INPUT

1. The general rules of data input are given in "How to Use MDS(X) programs", which is chapter 1 of The User Manual (see file *HowToUse*)
2. It is often easiest to begin by copying the Test Input file of the program you wish to use (after all, you KNOW that it is correct!) and then editing it to suit your data.
3. Check what the defaults are for the program (in the MDS(X) User Manual and MDS(X) Pocket Guide sections of the program Documentation); it saves having to make redundant instructions.
4. Every input file MUST have the following instructions:

```
INPUT FORMAT    .....
READ MATRIX
<data>
COMPUTE
FINISH
```

INPUT FORMAT specification is a recherché activity and dying art ... and sometimes occasions problems when a data-line exceeds the 80-column width. In brief, the format statement creates a "grid" for reading data; it allocates certain columns in a line/card to a sequence of variables (in the case of MDSX, often a line of a lower-triangular matrix). (See Note on Input Format....)

READ MATRIX must always follow INPUT FORMAT and the following data are read according to the INPUT FORMAT rule.

COMPUTE is the instruction actually to perform the analysis; without it, nothing will be done!

FINISH terminates a RUN; in its absence the program will look for another data-set to follow, but will not abort prior tasks.

RUNNING A PROGRAM

Programs can be run either interactively or as batch-jobs. At this stage, you are not advised to use the Interactive option ...

INTERACTIVELY

1. ... But if you do perform a run interactively, then type the name of the program you wish to use in capitals, and on a new line. (e.g. MINISSA)
2. Each Instruction is checked on entry, and is either accepted, or else it generates an error message.
3. In the case of error, the corrected instruction can then be re-typed.
4. After a specified number of errors, the program terminates, making it more sensible to work in ...

BATCH-MODE

1. The instruction to run a program, has the form:

   programname <inputfilename> outputfilename

2. It is ESSENTIAL to have the < > brackets round the input file-name, but not round the output file-name. In the normal way, the input-file will have been been pre-prepared, and the program will produce a file for the output (or it will over-write an existing file).
3. An example would be:

   minissa <mydata.inp> mydata.out

4. the outputfile will contain the output results (and/.or error-messages), in .txt format.
5. It is sensible to EDIT the output file, which sometimes has two unreadable characters at the start; delete these before going any further.
6. Errors due to unacceptable program instructions will appear in the body of the output file, followed by the information that the run has terminated. Run-time errors, by contrast, will appear in the command line. The Run-time errors (whose vagueness and opacity is legendary) are specified in the appendix to this note.

DATA OUTPUT

The output file will contain "line-printer output" (whose line-length was 126 characters), and as a consequence, wrap-around will almost certainly occur in today's MS-DOS PC environment. Fortunately, such wrap-around is easy to remedy:

1. take the output file into MSDOS *EDIT*, or WORDPAD
2. remove the first two (unreadable) characters (... don't ask!)
3. Save the file, and then insert into a WORD or WORDPERFECT file

4. set the type to an equal-spaced font, such as COURIER or CourierNew, and to a 7-point size.
5. It may also be necessary to widen margins to prevent residual wrap-around
6. It helps to embolden the entire file for readability
7. To titivate the output configuration by labelling points:
   Turn off the Insert key on the keyboard (because in this way additional spaces are not inserted, which would otherwise shift point-locations), and type in the brief stimulus name/ instead of the stimulus number
   Use colour to map information about different subsets of points.

EXPORTING RESULTS: PUNCH or "SPSS_OUTPUT"

MDS(X) has the possibility of providing "PUNCH " output. In the old day, this is exactly what happened – on Fortran Stream 07 (the card-punch) desired output data were produced, for input into other programs. Usually, users wanted Configuration co-ordinates (for subsequent graph-plotting) or a rectangular matrix of residuals etc.

This still has utility as a primitive EXPORT procedure. The specified results are in a file called FORT7 in ASCII format, and come ready with their INPUT FORMAT.

APPENDIX

RUN-TIME ERROR MESSAGES (using MDSX)

Most MDSX program error messages are caught by the routines and reported in MDSX terms. However several are generated by the FORTRAN system used to create the programs. These generally appear as numbers tied to a printed set of error messages in the IBM Fortran/2 manual. As few people will have access to this manual, the list below identifies the most likely errors. Note that almost all are provoked by faulty commands ---very few are intrinsic to the programs themselves.

Number	EXPLANATION
034	Invalid input/output unit (will usually refer to a data file)
077	Divide by zero (usually a data problem)
2000-2499	Messages in this range refer to Input/Output problems (check command files)
2500-2548	Messages in this range refer to Format Control of Input/Output (check command files)
2512	Incorrect character after format specifier width (the fieldwidth fraction part of an 'F' field descriptor is missing a period.)
2514	Incorrect integer in input (an 'I' input field is missing a legal integer format)
2516	Incorrect format specifier start (a format specifier doesn't begin with a left parenthesis --check that the INPUT FORMAT begins in column 16)
2536	Specifier missing in format specification (usually missing comma between items in a format statement)
2537	Too many brackets in a format specifier (only 9 pairs of brackets allowed)
3000-3098	Messages in this range refer to problems with the Operating System Interface
3012	DOS couldn't find a file
3023	End of file before newline on reading a formatted sequential record (A 'newline' is a carriage return/line feed pair; an end of file is either an end of file condition or an end of file marker)
3076	Problem encountered writing to an output file.
3081/3082/3083/3084	Errors reading an unformatted data file (could be a corrupt file)
3091	Memory reallocation failure (insufficient memory to satisfy a request for more working space)

et/aug93