

# XMATRIX

Calculation of design matrices for ANOVA in linear models

A computer program  
by Marti J. Anderson



Department of Statistics  
University of Auckland  
(2003)

## DISCLAIMER

This FORTRAN program is provided without any explicit or implicit warranty of correct functioning. It has been developed as part of a university-based research program. If, however, you should encounter problems with this program, the author will be happy to help solve them. Researchers can use this program for scientific and research purposes, but intellectual property and copyright for the source code and program remains the property of Marti Jane Anderson. This program relies on several canned FORTRAN routines from Numerical Recipes (Press et al. 1992). Namely, for finding the eigenvalues of a real symmetric matrix, using householder reduction, XMATRIX uses routines “tred2” and “tqli.” Also, to find the solution to linear equations (i.e. the inverse of a matrix) by Gauss-Jordan elimination, XMATRIX uses the “gaussj” subroutine.

Users of the program may refer to the present user’s guide as follows:

Anderson, M.J. (2003). XMATRIX: a FORTRAN computer program for calculating design matrices for terms in ANOVA designs in a linear model. Department of Statistics, University of Auckland, New Zealand.

*Author’s contact details:*

Dr Marti J. Anderson  
Department of Statistics  
University of Auckland  
Private Bag 92019  
Auckland, New Zealand  
Tel: 64-9-373-7599 ext 85052  
Fax: 64-9-373-7018  
Email: [mja@stat.auckland.ac.nz](mailto:mja@stat.auckland.ac.nz)  
Website: <http://www.stat.auckland.ac.nz/~mja>

## I. Description

XMATRIX is a computer program to help researchers produce design matrices corresponding to factors or interaction terms in ANOVA designs. Basically, it is well-known that ANOVA is a linear model and, as such, any term in an ANOVA design can be tested using a linear regression-type approach. For examples, see Draper and Smith (1981), Neter et al. (1996) or Appendix C in Legendre and Anderson (1999). Thus, it is possible to test multivariate response variables in complex ANOVA designs, one term at a time, on the basis of any dissimilarity measure of choice and with permutation tests, by using the DISTLM procedure (McArdle and Anderson 2001). The way to code the factor levels as categories in an ANOVA model requires a bit of time and energy, however, to do by hand. Thus, the current program was designed to help with creating the necessary input **X** matrices for testing terms in ANOVA models using DISTLM.

XMATRIX will generate simple contrasts (using the values of  $-1$ ,  $+1$  and  $0$ ) corresponding to the levels of factors (i.e. level 1 vs. level 2, level 2 vs. level 3, etc.). There will be as many contrasts as there are degrees of freedom for the factor of interest. These columns can then be output as raw codes or as orthogonal codes (which will produce different results from the raw codes in the case of an unbalanced design). Although there are many other ways of coding **X** matrices for ANOVA designs, the ones produced by the XMATRIX program are particularly efficient. Obviously, if some other particular contrasts are desired, these must be obtained using some other method. Note that you can stop the program at any time by typing Ctrl-C (as for any DOS program) or, if you are a Mac user and this does not work, then try typing a period (a full stop) while holding down the apple key.

To avoid dealing with long file names and paths to locate files, it is best to keep all files to be used by the program in the same place as the XMATRIX.exe (or XMATRIX.mac) file, for use with the program. If you choose to create a file in Excel to be used by the program, then save it as an ASCII \*.txt file. If you are using a Macintosh, then save the file (for example, from Excel) as "Text (Windows)". If you save it as a simple text file from Excel, it won't run unless you open up the file again (for example, using a text editor such as BBEdit) and manually add a carriage return ("Enter") to the end of the last line. This is a weird quirk of the way the Mac version of Excel saves text files that, for some reason unknown to me, causes my program to crash. You should not experience any such problem, however, if the file is saved as "Text (Windows)" or if the file was created using a different text editor. Depending on the nature of your particular problem, you may not need to use Excel at all to manipulate files and may be able to just use the XMATRIX program from scratch.

The program uses dynamic memory allocation, and so (theoretically) does not have any limits on the sizes of matrices (numbers of rows or columns) that may be used for the input files. If you are using a Macintosh and you get an error that reads:

```
BUFFER allocation failed
REWIND(UNIT=*,...
```

then you need to increase the memory allocated to the program. To do this, click on the program's icon and type "i" while holding the apple key, then choose "Memory" (or choose "Get Info > Memory" from the File menu). Depending on the size of your matrices, you might even have to increase this value to something obnoxious, like 50000 or so. However, if you have a large input file and cannot seem to get the program to work even after doing this, then please contact the author.

## II. Generating X matrices

The use of the program is best demonstrated by an example. Consider the experiment concerning the effect of shade and the effect of proximity to the seafloor on assemblages of subtidal invertebrates and algae on hard surfaces near marinas (courtesy of Dr Tim Glasby). For further details, see Glasby (1999).

The experiment was a two-way crossed (orthogonal) design having the following structure:  
Factor 1 = Position (fixed, 2 levels: far or near to the seafloor)

Factor 2 = Shade (fixed, 3 levels: shade, a procedural control in the form of a clear plexiglass shade, and no shade)  
and  $n = 4$  replicate observations per combination of levels of the above two factors.

In this case we wish to generate the following three  $X$  matrices for the terms in the ANOVA model for this experimental design:

$X_P$  corresponding to the effect of Position,

$X_S$  corresponding to the effect of Shade, and

$X_{P \times S}$  corresponding to the interaction term.

## A. The first factor

Let's start by generating the  $X$  matrix for Position. To create it, it is necessary to run the program twice:

1) Double-click on XMATRIX.exe (or XMATRIX.mac) and answer the questions to generate a single column of numbers, as follows:

Do you wish to output:

- 1) X matrix codes for a single term
- 2) X matrix codes for an interaction term
- 3) X matrix codes for a nested term
- 4) A single column of numbers for levels of a factor  
i.e. category labels

**4**

What is the number of levels of the factor?

**2**

How many replicates are there per factor level?

**12**

Type a name for the output file of results (\*.txt)

**Pos.txt**

Do you wish to repeat the output matrix several times?

- 1) Yes
- 2) No

**2**

Matrix has been written to output file

End of the program

Type q to quit

**q**

Opening up the output file “Pos.txt”, we see the following:

```

1.0
1.0
1.0
1.0
1.0
1.0
1.0
1.0
1.0
1.0
1.0
1.0
1.0
1.0
1.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0
2.0

```

This is a single column of numbers that shows which level of “Position” each of the 24 observations belongs to. It is the information we need in order to generate an appropriate X matrix for the factor. We can either use option 4 in XMATRIX to generate this, or we can create it first ourselves (e.g. in Excel), and then run XMATRIX on the resulting column. The latter would be necessary if we wished to generate codes for an unbalanced design.

Once we have obtained this column of numbers, we then have to run the program again by double-clicking on XMATRIX.exe (or XMATRIX.mac) to get the X matrix itself, as follows:

Do you wish to output:

- 1) X matrix codes for a single term
- 2) X matrix codes for an interaction term
- 3) X matrix codes for a nested term
- 4) A single column of numbers for levels of a factor  
i.e. category labels

**1**

Type the name of the file containing the variable with the category labels  
(as integers from 1 to the no. of categories)

**Pos.txt**

What is the total no. of rows (observations)?

**24**

What is the number of levels?

**2**

Type a name for the output file of results (\*.txt)

**PosX.txt**

Note: If the program crashes it may be because you already have a file in the directory with the same name as what you are typing here for the output file. If so, then you need to delete the previous file or type a new name. Unfortunately, the program is not smart enough to write over the top of an existing file.



Do you wish to output:

- 1) X matrix codes for a single term
- 2) X matrix codes for an interaction term
- 3) X matrix codes for a nested term
- 4) A single column of numbers for levels of a factor  
i.e. category labels

**4**

What is the number of levels of the factor?

**3**

How many replicates are there per factor level?

**4**

Type a name for the output file of results (\*.txt)

**Shade.txt**

Do you wish to repeat the output matrix several times?

- 1) Yes
- 2) No

**1**

How many times do you want to repeat it?

**2**

Note: we need to repeat the codes for Shade twice because there are two levels of factor 1 in this design.

Matrix has been written to output file

End of the program

Type q to quit

**q**

Opening up the file Shade.txt, we see:

```

1.0
1.0
1.0
1.0
2.0
2.0
2.0
2.0
3.0
3.0
3.0
3.0
1.0
1.0
1.0
1.0
2.0
2.0
2.0
2.0
3.0
3.0
3.0
3.0

```

Note how the coding for the second factor differs from the coding for the first factor. When running DISTLM, it will be very important that you know the order of the factors in the response data file, so that you know how to code the **X** matrices for each of those factors in an appropriate way. Next, we run the program again to get the actual **X** codes for the Shade factor, as follows:

Do you wish to output:

- 1) X matrix codes for a single term
- 2) X matrix codes for an interaction term
- 3) X matrix codes for a nested term
- 4) A single column of numbers for levels of a factor  
i.e. category labels

**1**

Type the name of the file containing the variable with the category labels (as integers from 1 to the no. of categories)

**Shade.txt**

What is the total no. of rows (observations)?

**24**

What is the number of levels?

**3**

Type a name for the output file of results (\*.txt)

**ShadeX.txt**

Do you want raw codes or orthogonal codes?

- 1) raw
- 2) orthogonal

**1**

Do you wish to repeat the output matrix several times?

- 1) Yes
- 2) No

**2**

Note: we do not need to repeat the matrix several times in this case because the single column of numbers that we have input already includes all 24 rows and so doesn't need to be expanded again. Keep in mind that ALL of your **X** matrices for a single design are going to have the SAME total number of rows. This is important in order for them to be able to be related to the **Y** matrix when the time comes, using DISTLM.

Matrix has been written to output file

End of the program

Type q to quit

**q**

Opening up the file “ShadeX.txt”, we see:

```

1.0    0.0
1.0    0.0
1.0    0.0
1.0    0.0
-1.0   1.0
-1.0   1.0
-1.0   1.0
-1.0   1.0
0.0   -1.0
0.0   -1.0
0.0   -1.0
0.0   -1.0
1.0    0.0
1.0    0.0
1.0    0.0
1.0    0.0
-1.0   1.0
-1.0   1.0
-1.0   1.0
-1.0   1.0
0.0   -1.0
0.0   -1.0
0.0   -1.0
0.0   -1.0

```

### C. The interaction term

Once we have the **X** matrices corresponding to the individual factors, we are ready to generate **X** matrices for interaction terms. This is pretty straightforward and is demonstrated below for the generation of an **X** matrix for the interaction of Position x Shade:

Do you wish to output:

- 1) X matrix codes for a single term
- 2) X matrix codes for an interaction term
- 3) X matrix codes for a nested term
- 4) A single column of numbers for levels of a factor  
i.e. category labels

**2**

Type the name of the file containing the 1st X matrix

***PosX.txt***

What is the number of columns in this matrix?

**1**

Type the name of the file containing the 2nd X matrix

***ShadeX.txt***

What is the number of columns in this matrix?

**2**

What is the total number of rows in the matrices?

**24**

Type a name for the output file of results (\*.txt)

***PxS.txt***

Do you want raw codes or orthogonal codes?

- 1) raw
- 2) orthogonal

**1**

Do you wish to repeat the output matrix several times?

- 1) Yes
- 2) No

**2**

End of the program

Matrix has been written to output file

Type q to quit

**q**

The **X** matrix codes for an interaction term are simply the direct products of the columns of the **X** matrices for their contingent factors. In the present case, the file "PxS.txt" has the following codes:

1.0	0.0
1.0	0.0
1.0	0.0
1.0	0.0
-1.0	1.0
-1.0	1.0
-1.0	1.0
-1.0	1.0
0.0	-1.0
0.0	-1.0
0.0	-1.0
0.0	-1.0
-1.0	0.0
-1.0	0.0
-1.0	0.0
-1.0	0.0
1.0	-1.0
1.0	-1.0
1.0	-1.0
1.0	-1.0
0.0	1.0
0.0	1.0
0.0	1.0
0.0	1.0

Note also that if you want to generate codes for three-way interactions, such as codes for  $A \times B \times C$ , then you would need to generate codes for  $A \times B$  first, then you would need to calculate the codes for the interaction between  $A \times B$  and  $C$  as a second step.

## D. A nested term

Generating codes for a nested term using XMATRIX is actually somewhat easier than even generating them for a single factor, at least in the case of a balanced design. Imagine that you would like to generate a matrix for factor B which has 4 levels and is nested within factor A, which has 3 levels, and there are  $n = 2$  observations within each level of B within A. You would only need to run XMATRIX *once*, as follows:

Do you wish to output:

- 1) X matrix codes for a single term
- 2) X matrix codes for an interaction term
- 3) X matrix codes for a nested term
- 4) A single column of numbers for levels of a factor  
i.e. category labels

**3**

What is the no. of levels of the upper level factor?

**3**

What is the no. of levels of the nested factor?

**4**

What is the number of observations per cell?

**2**

Type a name for the output file of results (\*.txt)

***NestedBX.txt***

Do you want raw codes or orthogonal codes?

- 1) raw
- 2) orthogonal

**1**

Do you wish to repeat the output matrix several times?

- 1) Yes
- 2) No

**2**

End of the program

Matrix has been written to output file

Type q to quit

**q**

Now, opening up the file "NestedBX.txt", we see:

1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	-1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	-1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	-1.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	-1.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.0

Note that the first 3 columns code for the four levels of factor B that are nested within level 1 of factor A (rows 1-8), with zeros everywhere else. Then columns 4, 5 and 6 code for the four levels of B that are nested within the second level of factor A (rows 9-16), with zeros everywhere else. Finally, the last three columns code for the four levels of factor B that are nested within the third level of factor A (rows 17-24), with zeros everywhere else. Thus, for a nested term, it is clear that there will be  $a*(b - 1)$  columns required for the coding, just like there are  $a*(b - 1)$  degrees of freedom for a nested term (with  $a$  = the number of levels of factor A and  $b$  = the number of levels of factor B).

### III. Running DISTLM with appropriate XMATRIX files

We next turn to an example of running DISTLM using the **X** matrices produced for individual factors from Tim Glasby's (1999) study. Before we run the program, it is important to know for any particular factor what the denominator is for the test. For example, to test the term "Position", we need to realize that the residual mean square is the denominator of this *F*-ratio (in this case, because the factors are fixed). In order for DISTLM to know what the "residuals" are in this model, we will therefore need to give it the **X** matrix corresponding to the full model. In the present case, this consists of the **X** matrices for Position, Shade and their interaction placed side by side. This can be done by opening up the individual **X** matrix files in Excel, pasting them together into a single sheet and saving this full model **X** matrix as an ASCII \*.txt file (with an appropriate descriptive name such as "FullX.txt"). The full model **X** matrix has five columns and looks like this for Tim's study:

```

1      1      0      1      0
1      1      0      1      0
1      1      0      1      0
1      1      0      1      0
1     -1      1     -1      1
1     -1      1     -1      1
1     -1      1     -1      1
1     -1      1     -1      1
1      0     -1      0     -1
1      0     -1      0     -1
1      0     -1      0     -1
1      0     -1      0     -1
-1     1      0     -1      0
-1     1      0     -1      0
-1     1      0     -1      0
-1     1      0     -1      0
-1    -1      1      1     -1
-1    -1      1      1     -1
-1    -1      1      1     -1
-1    -1      1      1     -1
-1     0     -1      0      1
-1     0     -1      0      1
-1     0     -1      0      1
-1     0     -1      0      1

```

Now, double clicking on "DISTLM.exe" (or "DISTLM.mac"), we can run the analysis as follows, keeping in mind that the data are stored in Tim.txt (with 46 variables and 24 observations), as described in the notes for PERMANOVA.

```
DISTLM v.4
-----
```

```
A program for analysing multivariate data
on the basis of any distance measure,
according to any linear ANOVA or regression model,
using permutations.
```

```
by M.J. Anderson
Department of Statistics
University of Auckland (2004)
```

```
Type the name of the input file containing your data
(i.e. response variables).
```

```
Tim.txt
```

```
Type a name for the output file of results (*.txt)
```

```
Results_Pos.txt
```

Nature of the data in the input file:

- 1) raw data (n x p)
- 2) distance matrix (n x n)

**1**

Structure of the input file:

- 1) rows are samples and columns are variables
- 2) columns are samples and rows are variables

**1**

How many variables (columns) are there?

**46**

How many observations (rows) are there?

**24**

Choice of transformation:

- 1) none
- 2) square-root
- 3) fourth-root
- 4)  $\ln(x)$
- 5)  $\ln(x+1)$
- 6)  $\log_{10}(x)$
- 7)  $\log_{10}(x+1)$
- 8) presence/absence

**3**

Choice of standardisation:

- 1) none
- 2) standardise by row (sample) sums
- 3) standardise by column (variable) sums
- 4) double standardise by row and column sums
- 5) standardise each variable to z-scores (normalise)
- 6) standardise each variable by dividing by its s.d.
- 7) standardise each variable by dividing by its range

**1**

Choice of distance measure:

- 1) Bray-Curtis dissimilarity
- 2) square root of Bray-Curtis
- 3) Euclidean distance
- 4) Orloci's Chord distance
- 5) Chi-square metric
- 6) Chi-square distance
- 7) Hellinger distance
- 8) Gower dissimilarity
- 9) Gower, excluding double zeros
- 10) Canberra distance
- 11) square root of Canberra distance
- 12) CY dissimilarity
- 13) Deviance based on the binomial
- 14) Deviance per observation (scale invariant)
- 15) Kulczynski dissimilarity
- 16) McArdle-Gower dissimilarity
- 17) Manhattan (city block) distance
- 18) Jaccard dissimilarity

**1**

Do you want to output the distance matrix?

- 1) Yes
- 2) No

**2**

Do you wish to output the mean squares for the numerator and denominator under permutation?

- 1) No
- 2) Yes

**1**

Is this test for an ANOVA design or a regression model?

- 1) Regression model
- 2) ANOVA design

**2**

Do you wish to obtain a P-value using a Monte Carlo sample from the theoretical asymptotic distribution under permutation?

(Note: useful if there are very few possible permutations)

- 1) Yes
- 2) No

**1**

Note: The asymptotic distribution of the test statistic under permutation is given by Anderson and Robinson (2003). As the program suggests, this can be extremely useful if there are too few possible permutations to get a reasonable test. As a matter of course, I virtually always say "Yes" to this question. The permutation and the Monte Carlo *P*-value will tend in any case to give extremely similar results (although the permutation test is exact). They will differ in the case of there being too few possible permutations to get a decent permutation test, in which case, trust and use the Monte Carlo *P*-value instead.

Type the name of the file containing the X design matrix which has the codes for the TERM OF INTEREST for the TEST (and for which rows correspond to the n observations)

**PosX.txt**

What is the number of columns for this X matrix?

**1**

Is the denominator mean square (MS) for the test provided by:

- 1) the residual MS, with no other terms in the model
- 2) the residual MS, but there are other terms in the model, or
- 3) the MS of another term in the model.

**2**

Type the name of the file containing the X design matrix for the FULL model, with all terms included (with rows corresponding to the n observations)

**FullX.txt**

What is the number of columns for this X matrix?

**5**

How many permutations do you want for the test?

(i.e. 99, 499, 999, 4999, etc.)

**9999**

Which general method of permutation do you want?

- 1) Unrestricted permutation of raw data or units
- 2) Permutation of residuals (reduced model)
- 3) Permutation of residuals (full model)

**1**

Type an integer to be used as the seed for the random permutations

**49**

Do you wish to permute units other than the individual observation units (i.e. groups of units)?

- 1) No
- 2) Yes

**I**

Note: this is an important question. If the denominator were *not* the residual, then a different set of units might need to be permuted under the null hypothesis. See Anderson and ter Braak (2003) for details. If you need to answer “Yes” to this question, then you need to provide the program with a single column of numbers that indicate which observations should be kept together as units under permutation. This can usually be generated by using the XMATRIX program and choosing option 4.

The permutations are then done (with the screen giving updates) and the results are output to the screen and to the output file, as follows:

```

DISTLM v.4
-----
A program for analysing multivariate data
on the basis of any distance measure,
according to any linear ANOVA or regression model,
using permutations.

by M.J. Anderson
Department of Statistics
University of Auckland (2004)

Input file of data: Tim.txt
Input file of X design matrix for term of interest: Pos.txt
Input file of X design matrix for the full model: Full.txt

The no. of observations = 24
The no. of variables = 46

--- Results ---
-----

```

	df	Sum of squares	Mean square
Numerator	1	5636.32713	5636.32713
Denominator	18	7428.27464	412.68192
Total	23	17907.84249	

```

-----
                                pseudo-F = 13.65780
                                permutation P = 0.00010
                                Monte Carlo P = 0.00010
-----
Data were transformed to fourth root
No standardisation
Analysis based on Bray-Curtis dissimilarities
Permutation of raw data
No. of permutations used = 9999
The integer chosen as the seed for the permutations = 32
Proportion of variation explained = 0.3147

```

Note the  $F$ -statistic here is identical to the value of  $F$  obtained for the factor Position using PERMANOVA (see the PERMANOVA user notes). Any difference in  $P$ -values would be due to the choice of seed for the random number generator.

## IV. References

Anderson, M.J. and Robinson, J. (2003). Generalized discriminant analysis based on distances. *Australian and New Zealand Journal of Statistics* 45(3): 301-318.

- Anderson, M.J. and ter Braak, C.J.F. (2003). Permutation tests for multi-factorial analysis of variance. *Journal of Statistical Computation and Simulation* 73: 85-113.
- Draper, N.R. and Smith, H. (1981). Applied regression analysis, 2<sup>nd</sup> edition. John Wiley and Sons, New York.
- Glasby, T.M. (1999). Interactive effects of shading and proximity to the seafloor on the development of subtidal epibiotic assemblages. *Marine Ecology Progress Series* 190: 113-124.
- Legendre, P. and Anderson, M.J. (1999). Distance-based redundancy analysis: testing multispecies responses in multifactorial ecological experiments. *Ecological Monographs* 69: 1-24.
- McArdle, B.H. and Anderson, M.J. (2001). Fitting multivariate models to community data: a comment on distance-based redundancy analysis. *Ecology* 82(1): 290-297.
- Neter, J., Kutner, M.H., Nachtsheim, C.J. & Wasserman, W. (1996). *Applied linear statistical models*, 4<sup>th</sup> edition. Irwin, Chicago, Illinois.
- Press, W.H., Teuklosky, S.A., Vetterling, W.T. and Flannery, B.P. (1992). *Numerical Recipes in FORTRAN*, 2<sup>nd</sup> edition. Cambridge University Press, Cambridge.