

MICROSIRIS



Statistical and Data Management
Software System

Version 24

March 8, 2015

Developed by Van Eck Computer Consulting

Contributing editor and user interface design consultant:
Richard Neal Van Eck, PhD

COPYRIGHT 2014 NEAL & SUSAN VAN ECK. ALL RIGHTS RESERVED.
PORTIONS COPYRIGHT 1979-2003 THE REGENTS OF THE UNIVERSITY OF MICHIGAN. ALL RIGHTS RESERVED.
IVEware COPYRIGHT © 2002 THE REGENTS OF THE UNIVERSITY OF MICHIGAN. ALL RIGHTS RESERVED.

The development, production and support for MicroSiris are solely the responsibility of
Van Eck Computer Consulting.

MicroSiris

Copyright 2014 Neal & Susan Van Eck. All rights reserved.

Portions copyright 1979-2003 The University of Michigan. All rights reserved.

Permission is granted to use, copy and redistribute this software for any purpose, so long as no fee is charged and so long as the copyright notice above, this grant of permission, and the disclaimer below appear in all copies made. Permission to modify or otherwise create derivative works of this software is not granted.

This software is provided as is, without representation as to its fitness for any purpose, and without warranty of any kind, either express or implied, including without limitation the implied warranties of merchantability and fitness for a particular purpose. Van Eck Computer Consulting, the authors, and the Regents of the University of Michigan shall not be liable for any damages, including special, indirect, incidental, or consequential damages, with respect to any claim arising out of or in connection with the use of the software, even if it has been or is hereafter advised of the possibility of such damages.

IVEware

Copyright 2002 The Regents of The University Of Michigan. All rights reserved.

Permission is granted to use, copy and redistribute this software for any purpose, so long as no fee is charged and so long as the copyright notice above, this grant of permission, and the disclaimer below appear in all copies made; and so long as the name of The University of Michigan is not used in any advertising or publicity pertaining to the use or distribution of this software without specific, written prior authorization. Permission to modify or otherwise create derivative works of this software is not granted.

This software is provided as is, without representation as to its fitness for any purpose, and without warranty of any kind, either express or implied, including without limitation the implied warranties of merchantability and fitness for a particular purpose. The Regents of The University of Michigan shall not be liable for any damages, including special, indirect, incidental, or consequential damages, with respect to any claim arising out of or in connection with the use of the software, even if it has been or is hereafter advised of the possibility of such damages.

MONANOVA

This command is adapted from MONANOVA, written by J.B. Kruskal, copyright (c) 1993 by AT&T.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. This software is being provided "as is", without any express or implied warranty. In particular, neither the authors nor AT&T make any representation or warranty of any kind.

Trademarks

OSIRIS is a trademark of The University of Michigan.

MicroSiris is a trademark of Neal & Susan Van Eck.

SAS is a trademark of the SAS Institute, Inc.

SPSS is a trademark of IBM, Inc.

SYSTAT is a trademark of SYSTAT, Inc.

Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in Sub paragraph c(1)(ii) of the Right in Technical Data and Computer Software Clause at 252.227-7013

WORKFLOW IN MICROSIRIS

If you are familiar with others systems like SPSS, SAS, and STATA, you may have some initial difficulty understanding how what you do in MicroSiris relates to what you are used to doing. The following section shows the common workflow in MicroSiris.

1. Enter Data

- a. Can be done in MicroSiris directly:

Type into [Excel spreadsheet](#) using the "Data Entry with Excel" button on initial command screen and save the file as a .CSV file.

After saving the .CSV file, "Data Entry with Excel" creates a MicroSiris dataset from the .CSV file

- b. Use [IMPORT](#) to:

Import a [CSV](#) data file from SAS or STATA (See [Using SAS Data](#) or [Using STATA Data](#))

Import an SPSS .por file (See [Using SPSS Data](#)).

Once in MicroSiris, can be displayed with [LIST DATASET](#)

2. Clean up data

Use [WILDCODE CHECK](#) for outliers

Check univariates ([USTATS](#) and [TABLES](#))

Use [CONSISTENCY CHECK](#) for inconsistencies

Use [TRANSFORM](#) with a filter to permanently eliminate cases

Correct the data

Use [EDIT DATASET](#) to correct data

Use [RECODE](#) setup from [CONSISTENCY CHECK](#) to eliminate systematically.

3. Build indices, brackets, transform with [RECODE](#) and [TRANSFORM](#)

4. Aggregate data if needed ([AGGREGATION](#), [RECODE](#))

May need to sort first ([SORT DATASET](#))

5. Add variables or cases to data if needed ([MERGE DATASETS](#))

6. Choose analyses

- a. [Select command](#)

Choose appropriate variables

Browse list by typing ? in selection field, selecting variable(s), and clicking the "add" button to add them to the list of variables, the weight variable, repetition variable, etc.

Choose appropriate variables by knowing your data and what you're trying to learn. You can list variables in the dictionary any time and view a saved dictionary listing using the VIEW button on the command screen.

Choose Options for analyses, including recoding as desired.

For commands that allow multiple analyses, i.e., "next" panels like the ANOVA dialog below, everything outside the boxed analysis parameters are global, e.g. recoding, and apply to all the analyses; all things inside the box are not, and the global parameters are greyed out or rendered unchangeable after entering the first analysis parameters.

One-way Analysis of Variance (ANOVA)

Filter/Job Title

Repetition statement ? Use previous

Use

Recode number ☐

Weight variable ☐

Largest dependent variable code allowed 10

Display

☒ No dictionary

☐ Dictionary

☐ Dictionary with value labels

Analysis Statement 1

Dependent (outcome) variables, e.g., V1-V3,V5,R6 or ? to browse

Treatment variables, e.g., V1-V3,R5,R6 or ? to browse

☐ Delete cases where equal to MD1 ☐ Delete cases where equal to MD2

HELP Next Done Cancel

- b. Perform analyses, producing output that is viewable and printable (use printed output check box on command screen)
- c. Use the VIEW button on the command screen to view, edit, output.

7. Analyze residuals if appropriate

8. Use related follow-on commands; e.g., [SEARCH](#) after using [MCA](#) to narrow the focus

GUIDE TO COMMANDS

Data Management Commands

Preparing Data for Input

IMPORT Imports SAS, SPSS, Excel, and CSV files and fixed-format files.
MERGE DATASETS..... Merges (combines) two or more datasets.

Sorting data sets

SORT DATASET Sorts datasets.

Checking and Correcting Datasets

CONSISTENCY CHECK Performs consistency checks among variables.
EDIT DATASET Corrects individual variables case by case.
TRANSFORM/RECODE..... Corrects and modifies datasets.
WILDCODE CHECK Checks and documents invalid variable values.

Displaying Datasets

LIST DATASET Lists dictionaries and datasets.

Transforming Datasets

AGGREGATION Aggregates individual records across user defined subsets and computes summary descriptive statistics.
EXPORT Exports datasets for other systems.
IMPUTE Perform single or multiple imputations of missing values using Sequential Regression Imputation.
RECODE Recodes variables.
TRANSFORM..... Reformats and transforms datasets; produces permanently recoded datasets when used with RECODE.

Matrix Manipulation Command

MATRANS..... Converts, prints and transforms matrices.

Converting output to a CSV file

TEXT-TO-CSV..... Converts output to a csv file for Excel report.

Analysis Commands

Frequency Distributions and Associated Statistical Measures

CHART Interface to Excel for plots.
DESCRIBE Estimates population means, proportions, subgroup differences, contrasts and linear combinations of means and proportions.
PROBABILITY Calculate probabilities for F, t, chi-square, and normal statistics.
TABLES Produces frequency tables, nonparametric statistics and measures of association for nominal and ordinal data.
USTATS Produces descriptive statistics (means, standard deviations).

Correlation and Regression Analysis

<u>IVEWARE</u>	Missing data imputation and regression analysis.
<u>LOGIT LINEAR</u>	Performs regression on a dichotomous dependent variable.
<u>CANCORR</u>	Canonical correlation.
<u>CORRELATIONS</u>	Computes Pearsonian correlation coefficients, allowing for unequal n's for each pair of variables.
<u>REGRESSION</u>	Performs multiple, step-wise, and dummy variable regression.
<u>REGRESS</u>	Fits linear, logistic, polytomous, Poisson, Tobit and proportional hazard regression models for data resulting from a complex sample design. The Jackknife repeated replication approach is used to estimate the sampling variances.

Analysis of Variance and Hypothesis Testing

<u>ANOVA</u>	One-way analysis of variance.
<u>ANOVAR</u>	One-way analysis of variance with repeated measures.
<u>DISCRIM</u>	Multivariate discriminant analysis.
<u>MANOVA</u>	Performs multivariate analysis of variance using a hierarchical regression model.
<u>CONJOINT</u>	Monotone(conjoint) analysis of factorial designs.
<u>T-TEST</u>	Performs hypothesis testing on means, differences of means, and differences of means between two groups.

Multivariate Analysis using Ordinal and Nominal Predictors

<u>MCA</u>	Performs multiple regression using one or more categorical independent variables (Multiple Classification Analysis).
<u>MNA</u>	Performs a multivariate analysis of nominal-scale dependent variables using a series of parallel dummy variable regressions.
<u>SEARCH</u>	Performs a binary segmentation procedure to find a set of predictors with the greatest predictive power.

Factor Analysis and Multidimensional Scaling

<u>CAP</u>	Analyzes a single spatial configuration.
<u>COMPARE</u>	Evaluates the similarity of two configurations of points.
<u>FACTOR ANALYSIS</u>	Performs factor analysis on a correlation matrix.
<u>MINISSA</u>	Guttman-Lingoes nonmetric multidimensional scaling.

Cluster Analysis

<u>CLUSTER ANALYSIS</u>	Performs cluster analysis, using partitioning, agglomerative, divisive, fuzzy, hierarchical, and monothetic methods.
--------------------------------------	----------------------------------------------------------------------------------------------------------------------

Index Reliability and Change Response Analysis

<u>CHANGE RESPONSE</u>	Change Response analysis.
<u>ITEM ANALYSIS</u>	Item analysis.
<u>INDEX RELIABILITY</u>	Internal consistency reliability of composite measures.

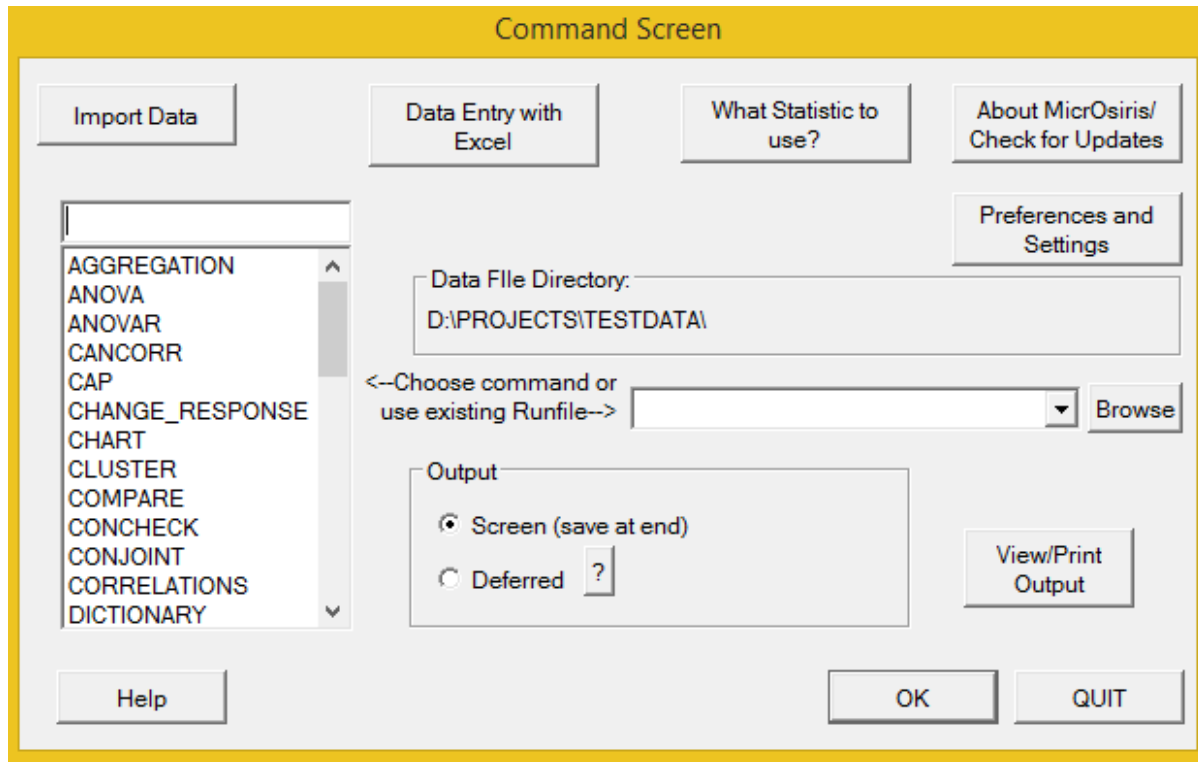
Survival Analysis

<u>LIFE TABLE</u>	Life table analysis.
--------------------------------	----------------------

FUNDAMENTALS

Getting Started with MicroSiris

When you first start MicroSiris, you are presented with the following screen:



This screen is used to specify what data files you will be working with, and what commands (statistics or data transformations) you want to use with your data file(s).

1. Use the [Data Entry with Excel](#) or [IMPORT Data](#) buttons to enter data.
2. Choose a command.
3. Use the [Decision Tree for Statistics](#) to determine appropriate statistics if desired.

After choosing a command you are prompted to specify the data files you wish to use. If no other choices are to be made, e.g., to send the output to a file instead of the screen, click OK to start the command. You can then:

1. Define a subset of the data to use ([Filter statement](#))
2. Provide a [job title](#) (default is a blank line)
3. Choose options, such as which variables to use, whether to use a [weight variable](#), what to print, how to handle "missing data, etc.

The MicroSiris Dataset

MicroSiris datasets are comprised of a data dictionary and a related data file. The dictionary file contains variable descriptions and code category labels.

The data file is where your actual data is stored. Together with the dictionary file it forms the MicroSiris dataset. The data file will always match the dictionary file but have the suffix .DAT instead of .DIC.

The data dictionary is a file of records, one per variable. When you ask for variable 10, for example, MicroSiris uses the dictionary to find its location within the data file and other information as the variable name. [Variable category labels](#) are also stored in the dictionary and retrieved for use in describing analysis results.

Data dictionaries are created when [importing](#) data into MicroSiris, or with the [Data Entry with Excel](#) button. They are revised with the [EDIT DATASET](#) command.

When you already have fixed length records with fixed field widths for variables, you must use [IMPORT](#) to create the dictionary to match the data file.

Variables

Each variable in a MicroSiris dataset has a number and a set of attributes associated with it including the location of the variable within each record of the data file, the variable width, type, number of decimal places, and the values to be treated as missing data. This information is stored in a dictionary.

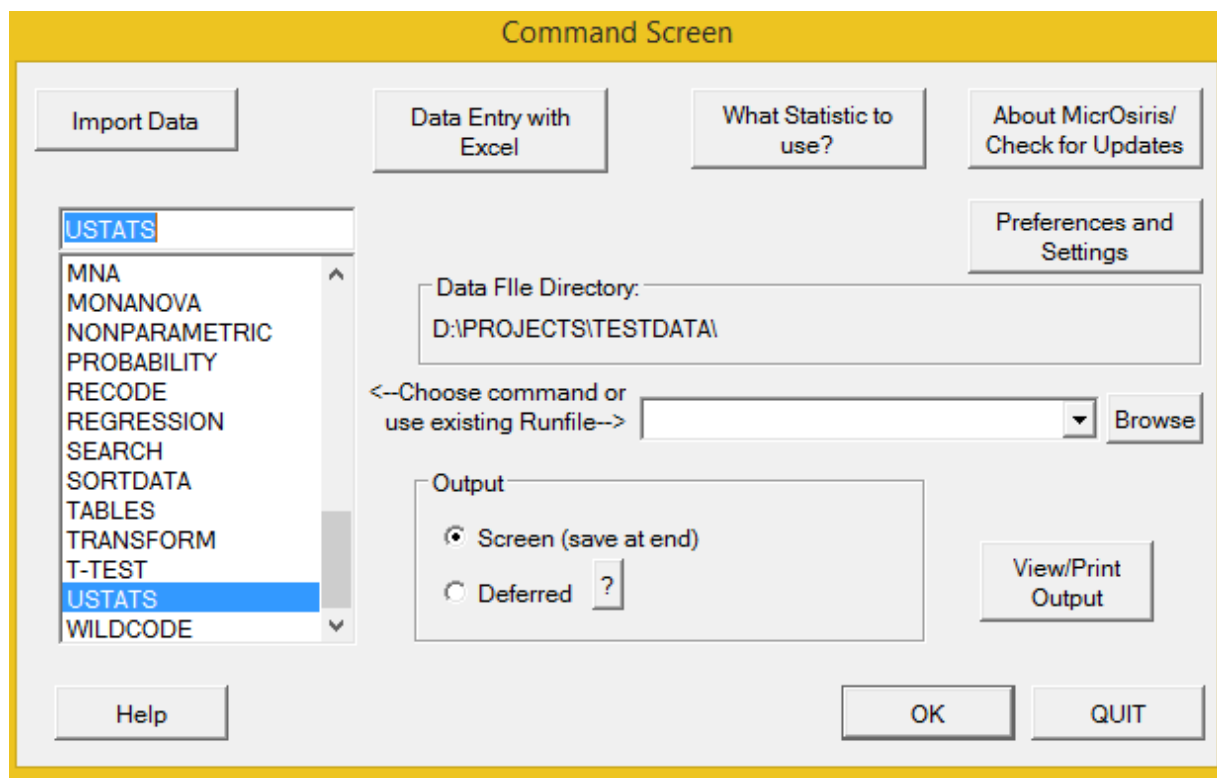
Variables are referenced in MicroSiris by their associated variable numbers, e.g., V12, but they can be selected interactively by name by putting a ? in the dialog box where variables are selected, instead of directly entering the variable numbers. The letter V preceding the variable number is optional in most cases except when using them in equations in [RECODE](#) and from RECODE result variables which are preceded with an R, e.g., R100=V1/2.

Category Labels

Category labels are another type of record optionally included in a MicroSiris dictionary. Category labels supply names for the codes used for each variable category, e.g., 1=none, 2=several times a week, 3=every day, used to label the printed output when appropriate. Category labels records are created and added to a dictionary with the [Data Entry with Excel](#) button when you create the dataset, or added and revised with the [EDIT DATASET](#) command.

Example: Univariate statistics with the USTATS command.

The example uses MicroSiris to obtain univariate statistics on three variables in the file SAMPLE provided with installation of MicroSiris. It gives a brief overview of the MicroSiris command process and illustrates some of its standard features.



The command screen snapshot shows a list box from which to choose a command. It also shows a data directory path telling MicroSiris where to find and store the data files you specify.

If you are unsure of which command to use, you can click on the HELP button at the bottom left of the command screen and select "Guide to Commands" to learn about the commands, or you can click on the Decision Tree for Statistics button to be guided through the process of selecting the appropriate statistics for you data file and research questions. Try it to get a feel for what's there.

You have the option of viewing the results on screen as they appear or you may defer the output until the command finishes and see it displayed with WORDPAD.

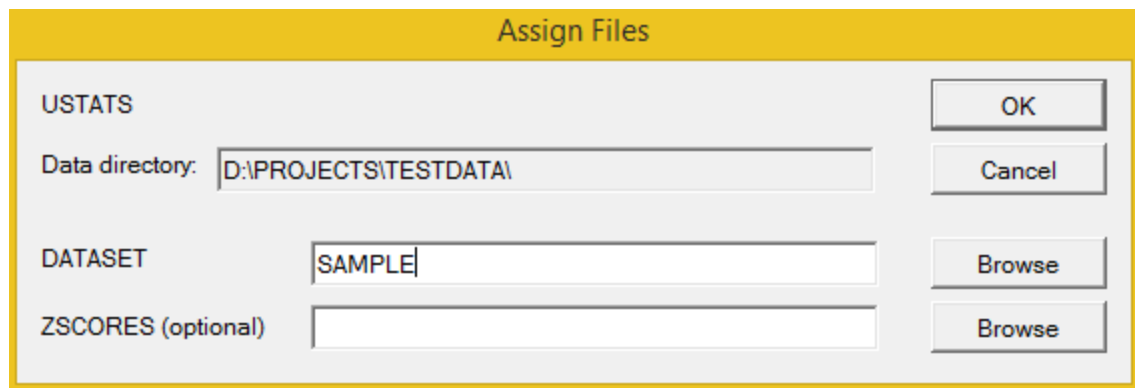
The sample screen above shows that output from the command will go to the screen by default because the button labeled Screen is shown in the "clicked" position.

After you specify the files and command you want to use, click the OK button.

The space labeled Runfile is for when you have created a "Runfile" with NOTEPAD, WORDPAD or a word processor. [Runfiles](#) are a useful way to store commands for complex jobs that may need rerunning or checking before final processing.

In our example, we want univariate statistics, so we scrolled down and selected USTATS. (If you know the command name you want, you can type it the blank space at the top of the list)

Next we need to tell MicroSiris where to find the data to analyze. As soon as we specify USTATS in the command box, a screen pops up indicating the file assignments needed:



Assign Files

USTATS

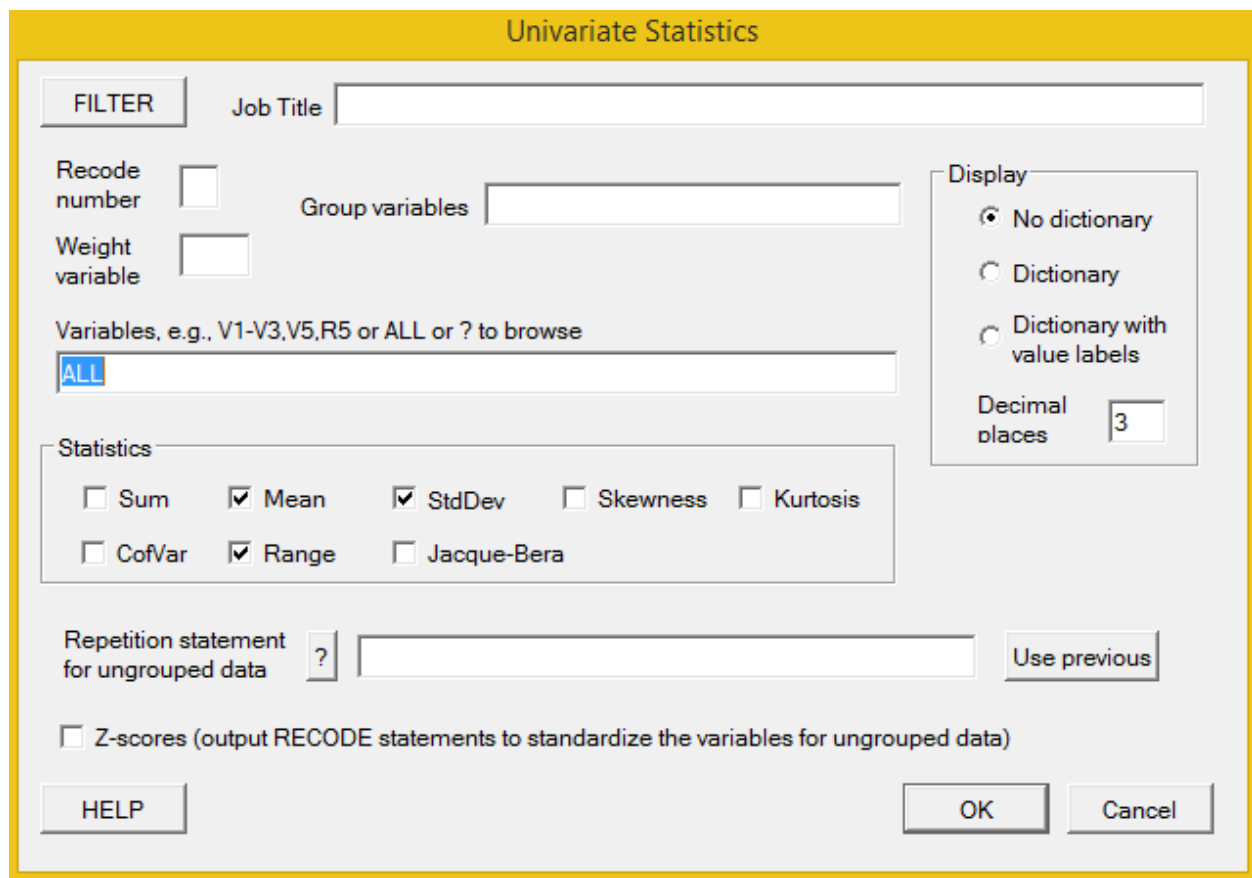
Data directory:

DATASET

ZSCORES (optional)

Type SAMPLE in the DATASET box and press RETURN or click the OK button to return to the command screen. Then click the OK button to start the USTATS command.

At this point, a dialog box opens that allows you to specify options:



Univariate Statistics

Job Title

Recode number ☐ Group variables

Weight variable

Variables, e.g., V1-V3,V5,R5 or ALL or ? to browse

Statistics

☐ Sum ☒ Mean ☒ StdDev ☐ Skewness ☐ Kurtosis

☐ CofVar ☒ Range ☐ Jacque-Bera

Display

☒ No dictionary

☐ Dictionary

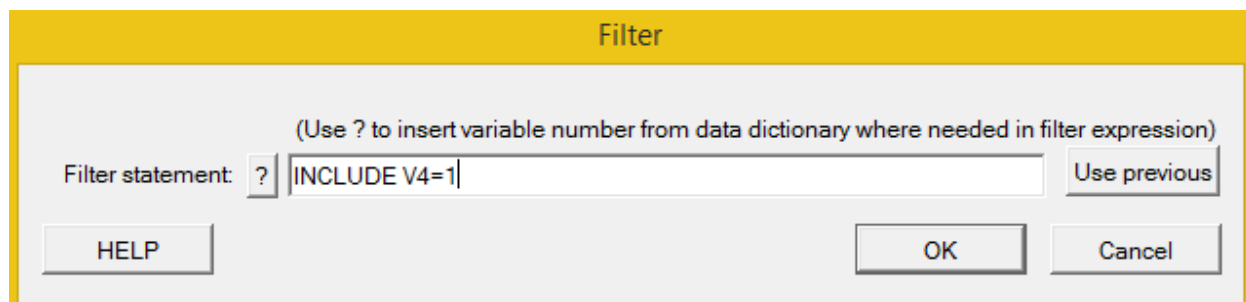
☐ Dictionary with value labels

Decimal places

Repetition statement for ungrouped data

☐ Z-scores (output RECODE statements to standardize the variables for ungrouped data)

Mean, StdDev, and Range are already checked because they are the defaults. Press the Filter button to add a filter:



Filter

(Use ? to insert variable number from data dictionary where needed in filter expression)

Filter statement: ? INCLUDE V4=1

Buttons: HELP, OK, Cancel, Use previous

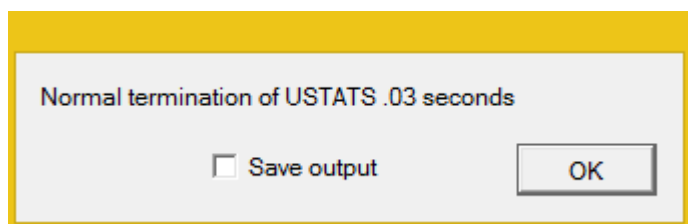
Type in the filter shown. The filter line tells USTATS to include only those cases where variable 4 has the value 1. When done, press return or OK to return to the USTATS window. Then click OK to take all the defaults and produce the univariate statistics:

		N	Mean	Standard Deviation	Minimum	Maximum
Better or Worse	V1	4	1.000	3.559	-4.000	4.000
Income (000)	V2	4	10.725	17.538	0.900	37.000
Children	V3	5	2.200	2.168	0.000	5.000
Weight 1	V4	5	1.000	0.000	1.000	1.000
Assets	V5	5	2.558	1.266	1.110	4.430

We chose to use all the variables in SAMPLE. If you want to use fewer variables and forget what variables are in the MicrOsiris dataset, just type a "?" in the variables entry field to get a quick listing of the variables in a window at the top-right of the screen (see right panel)

If you've already typed in some variable numbers, you can select more by typing a "?" at the end of the list already typed in. You can type a "?" in any box requiring a variable number, e.g., a weight variable or repetition statement, to get a list of available variables. If no dictionary (or matrix file) has been assigned, you are prompted for one.

Finally, you get the normal termination screen:



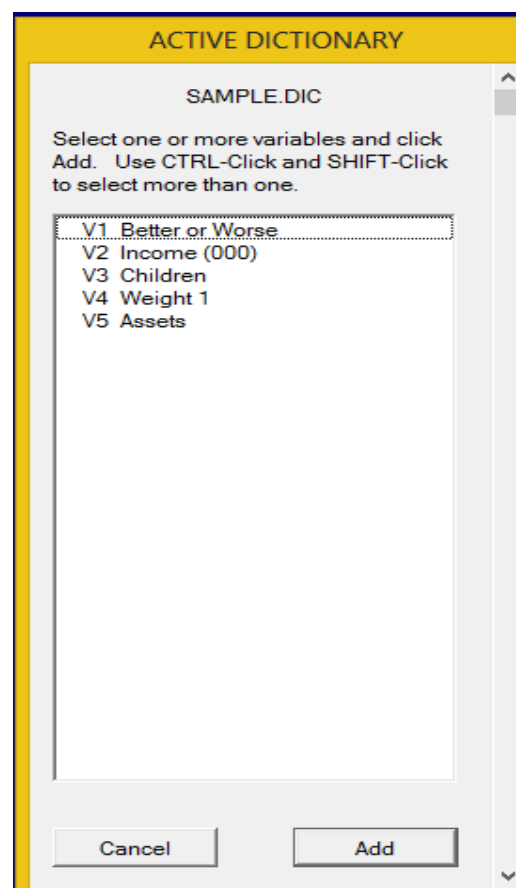
Normal termination of USTATS .03 seconds

☐ Save output

OK

You have now completed the process of generating your statistics and can begin interpreting the output.

If you click on Save output, you are also prompted for a file in which to save the output, and you can choose to save it as a normal text file or as a CSV file for import into Excel.:



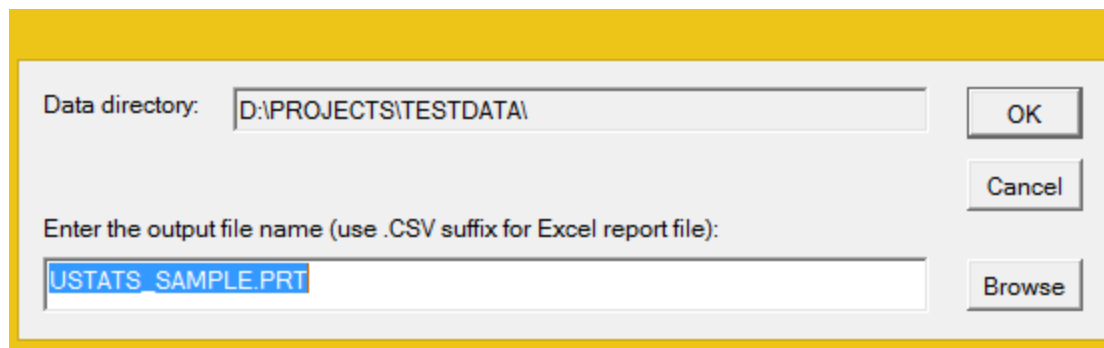
ACTIVE DICTIONARY

SAMPLE.DIC

Select one or more variables and click Add. Use CTRL-Click and SHIFT-Click to select more than one.

V1 Better or Worse
V2 Income (000)
V3 Children
V4 Weight 1
V5 Assets

Buttons: Cancel, Add



A screenshot of a software dialog box with a yellow border. The dialog box contains two input fields and three buttons. The first input field is labeled "Data directory:" and contains the text "D:\PROJECTS\TESTDATA\". To its right is an "OK" button. The second input field is labeled "Enter the output file name (use .CSV suffix for Excel report file):" and contains the text "USTATS_SAMPLE.PRT". To its right are "Cancel" and "Browse" buttons.

Data directory:	D:\PROJECTS\TESTDATA\	OK
Enter the output file name (use .CSV suffix for Excel report file):	USTATS_SAMPLE.PRT	Cancel Browse

The following sections provide additional information you may find helpful. There are other help sections that provide detailed information about working in MicroSiris, including how to get data into and out of [SPSS](#), [SAS](#), and [STATA](#) and how the workflow process works in MicroSiris.

Getting Data into MicroSiris

Preparing data for analysis with MicroSiris always involves the creation of two separate files: the dictionary file and the associated data file, collectively called a MicroSiris dataset.

You get new data into MicroSiris by:

1. Using the [Data Entry with Excel](#) button (easiest way).
2. [Importing](#) data from [SPSS](#), [SAS](#), or [STATA](#), or a >CSV file.

The Data File

The data file is where your actual data is stored. Together with the dictionary file it forms the MicroSiris dataset. The data file will always match the dictionary file but have the suffix .DAT instead of .DIC.

Weight Variables

A weight variable, which may be used in an analysis to increase or decrease the significance of a case, may be stored in the data as an additional variable. If different weights are appropriate to different uses of the data, each set of weights can be stored as a separate variable. You then select the appropriate one for a given analysis.

Formulas for many statistics available in MicroSiris assume simple random samples. Using a weight variable violates this assumption--you should be aware of this when interpreting results.

Missing Data

Missing data are data values specified by the user as inappropriate, unknown, or ambiguous. MicroSiris allows two missing-data values for each variable; these are normally used to exclude those data values from analytic techniques. If not specified in the dictionary, MD1 will be 1.5 billion by default and MD2 will be 1.6 billion. These values are unlikely to match any real values in a dataset and will not show in the dictionary. A [character numeric](#) field of all blanks is always considered missing data.

The value of a variable, X, is considered missing data if:

- X=MD1
- or MD2 is positive and $x \geq MD2$
- or MD2 is negative and $x \leq MD2$
- or $X=1.5$ billion.

With most analysis commands, missing-data values are automatically excluded from the analysis. However, some commands have specific options for the treatment of missing data.

See the [IVEWARE](#) command for a good way to impute values for missing data.

Matrix Files

Matrix files are for derived measures such as correlation coefficients, configurations, or factors. Usually these matrices are created by another command for further analysis. For example, the [correlation](#) command produces a matrix that can be used by the [regression](#) command.

You can also create matrix files within Excel (see [Matrix File Formats](#) for details).

A matrix created by a previous command in the same run is automatically available to subsequent commands if you assign the matrix a number in one command and reference that number in a subsequent command.

MicrOsiris Commands

Each MicrOsiris command consists of a set of file assignments and options to carry out a particular analysis or data manipulation.

Choose a command from the list shown on the command prompt screen and enter the file assignments when prompted.

Filter statement

A Filter statement selects a subset of data cases. It is expressed in terms of variables and the values they must have to include or exclude a data record ([case](#)) from processing by the command. Using filters is optional, but if supplied, each data record is tested against the filter criteria before any recoding or command-specific case selection is done. Examples are:

```
INCLUDE V2=1-5 AND V7=23, 27, 35 AND V8=1, 3, 4, 5  
EXCLUDE V10=2-3,6,8-9 AND V30=1-4 OR V91=25  
INCLUDE V50='CALIFORNIA','MICHIGAN','NEW YORK'
```

A REPEAT filter statement is a special form of the Filter statement which begins with REPEAT followed by a single variable number, and = sign, and a list of values for which to repeat an analysis. Example:

```
REPEAT V1=1,2,>9
```

The REPEAT filter causes the analysis to be performed for each value of the variable, i.e., the example above requests an analysis for all records with V1=1, and repeated for all records with V1=2, and again for all values greater than 9.

In a REPEAT statement ranges are allowed, but the REPEAT values must all be integers, and the range indicates a repetition for each value in the range. Thus REPEAT V1=1-3 is the same as REPEAT V1=1,2,3.

If you need to do additional filtering when using a REPEAT filter, you can add an AND expression to the REPEAT statement:

```
REPEAT V1=1,2,>9 AND V24>99
```

You can also do additional filtering using RECODE with REJECT:

```
RECODE
```

```
IF V10 EQ 5 AND V2=1 THEN REJECT  
END
```

You can use a REPEAT filter with commands that also have a REPETITION statement (ANOVA , ANOVAR, TABLES, T-TEST, and USTATS), but the REPETITION statement is more efficient since it requires only a single pass of the dataset.

Format for Filters

1. A filter begins with INCLUDE, EXCLUDE or REPEAT.
2. A filter may contain up to 100 expressions consisting of a variable number, an equals sign (=), and a list of possible values, e.g., V2=1, 5-9.
3. Connect expressions with the conjunctions AND and OR.
4. Expressions are evaluated from left to right, with expressions connected by AND always evaluated before expressions connected by OR.

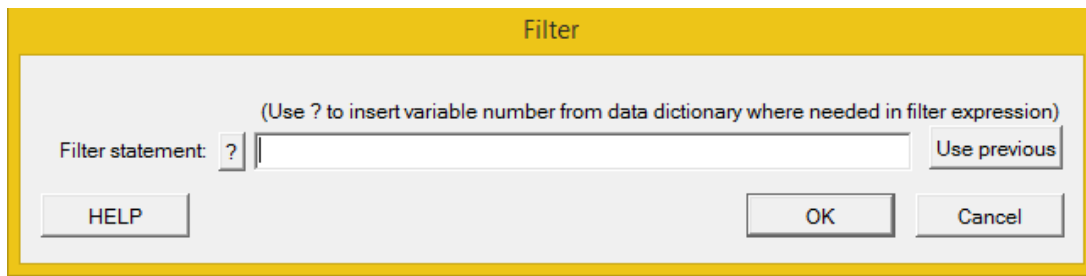
For example, "V1=2 OR V3=5 AND V10=9" requires that either a value of 2 from variable 1 occurs, or values of 5 from variable 2 and 9 from variable 10 occur, or the specified values from each of the three variables occurs.

5. Parentheses are NOT allowed.
6. Variables may appear in more than one expression.

The Expressions in a Filter

1. Both numeric and alphabetic variables may be used, but only the first 8 characters of an alphabetic variable are significant. Alphabetic variable values must be enclosed in primes and are extended with trailing blanks for comparison with data values if needed.
2. Variables created or modified by RECODE cannot be used in Filters (see the REJECT statement in RECODE for an alternative).
3. Values are specified singly or as a range of values with decimal places as needed, and are separated by commas, e.g., "V1=2-3,5,7-9", or "V5=1.5-3.5"
4. Open-ended ranges are indicated by > or <, e.g., INCLUDE V1=0, 3-5,>10. 5.

When MicroSiris expects a filter, it prompts with:

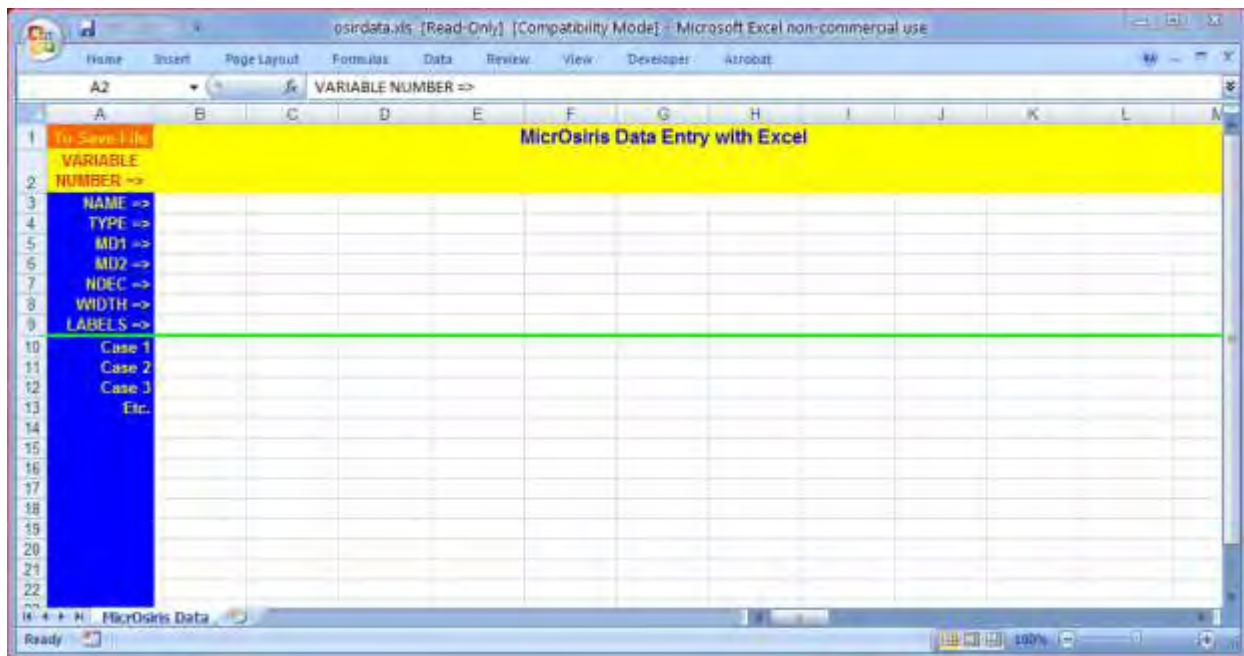


BUILDING OR IMPORTING A MICROSIRIS DATASET

If you enter your data using the "Data Entry with Excel" command button, you create a complete dataset (dictionary and datafile). When you **IMPORT DATA** data from another program or system such as SAS or SPSS, you also create a complete dataset.

Importing data using Excel

The easiest way to quickly get new data into MicroSiris is by clicking on the "Data Entry With Excel" button on the command screen. This opens the read-only template file DATA_ENTRY.XLSX.:



Pop-up directions appear as you select each cell.

1. Names can be up to 24 characters.
2. TYPE defaults to N (numeric). Numeric values are any acceptable number formats allowed in Excel. Numeric variables are always stored in MicroSiris as binary values of width 8, which retains 16 significant digits.
3. MD1 and MD2 are optional and specify codes which represent missing or inapplicable responses. If specified, they must be integers in the range -99999999 to 99999999 with no decimal places.
4. NDEC is the number of decimal places to use when displaying the variable value.
5. The WIDTH row is only used for alphabetic variables and is ignored for numeric variables. If left blank, a default value of 24 is used for alphabetic variables.

LABELS are code category labels. Enter the category labels, for each variable that has them, in the form:

1=label 1,2=label 3,....

The labels must be entered as one string and repeated for each variable that requires them. They may not contain a double quote (").

Values entered are checked for validity as they are entered.

After setting values for the initial rows, you can enter values in the data rows directly or paste data from other spreadsheets into them. An all-blank row below the green line is treated as the end of all data. Do not use line breaks (ALT-ENTER) in any field.

When all the data is entered, save the file as a [CSV](#) file.

Creating a MicroSiris Dataset with IMPORT

With [IMPORT](#), a MicroSiris dataset is created from an [SPSS](#) PC port file, a [SAS](#) or [STATA](#) file (exported as a [CSV](#) file), an Excel spreadsheet created via the DATA_ENTRY.XLSX file, or text data records from a [CSV](#) file created with a spreadsheet or database program. IMPORT reads records from the data file and assumes values appear in the order they are found.

If you already have a data file of fixed length records

When you have a data file with records all the same length and with each variable in a fixed-width field, you must use [IMPORT](#) to create a matching dictionary, specifying the location, width and form of each variable.

CLEANING THE DATASET

Checking for Wild Codes

Often the user of new or unfamiliar data wants to check for wild or invalid codes (variable values). The presence of non-numeric codes in numeric variables is classified in MicroSiris as "bad data" and is treated as missing data. The number of cases detected with bad data is always reported for those variables.

The **WILDCODE CHECK** command detects both bad data values and wild codes (erroneous values of numerical variables). It checks a set of variables for legitimate data values and lists all invalid codes by case ID and variable number. A simple example is shown below:

Options:

```
ID=V1,V2 MAXERR=100
```

Code specifications:

```
VAR3=V3-V5,V7 MIN=0 MAX=9  
VAR8=V8 INVALID CODE=7  
VAR10=V10 CODES=(1-5,8,9) MIN=1 MAX=9  
VAR11=V11
```

The example specifies that the only valid codes (values) for variables 3, 4, 5, and 7 are in the integer range 0 to 9. All codes for variable 8 are valid except 7, and the only valid codes for V10 are in the integer ranges 1 to 5, 8, and 9. Variable 11 will be checked only for non-numeric codes. Variables 1 and 2 are used as ID variables and the command will stop after 100 errors are reported (MAXERR= 100).

You should use WILDCODE CHJECKER routinely to validate all new datasets. Once the bad values have been identified, you can correct them using FIX_DATASET or RECODE with TRANSFORM

Another way to do a general check on the coding is to obtain simple frequencies with the TABLES command.

Consistency Checking

A different quality control measure is checking for logical consistency between and among variables. For example, checking that only records for women indicate more than zero pregnancies, or that a respondent's present age is greater than or equal to his or her age at marriage.

Consistency checking is accomplished with the **CONSISTENCY CHECK** and **RECODE** commands. RECODE is used to create for each test a result variable whose value is one if a case does not meet that particular consistency test and is zero otherwise. CONSISTENCY CHECK then tests the value of each result variable and prints an appropriate (user defined) message to document each inconsistency found, along with the values of the variables involved.

First test (only females have pregnancies)

```
IF V10 NE 1 AND V11 GT 0 THEN
```

d test (present age greater than or equal to age at m

```
IF V14 GT V15 THEN R2=1 ELSE R2=0
```

CONSISTENCY CHECK and specify

RECODE 1 ID=V2

TEST=R1 NAME='SEX DISCREPANCY' VARS=V10,V11

the first test (TEST=R1) each time variable 11 is greater

After the problem cases have been identified, the next step is to determine the root cause of the problem. This can be done by asking a series of questions, such as:

The easiest way is to use EDIT DATASET, assuming you have Excel or the OpenOffice



The screenshot shows a Microsoft Excel window titled "NEW.LSV - Microsoft Excel non-commercial use". The formula bar displays "A1 = Make corrections, save, and exit Excel to return to MicroSisn FIXDATA." The spreadsheet contains the following data:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Make corrections, save, and exit Excel to return to MicroSisn FIXDATA.											
2	VARIABLE V1	V2	V3	V4								
3	NAME	Better or income (0 Children)	Weight	I								
4	TYPE	N	N	N	N							
5	MD1		99									
6	MD2	9	99	9	9							
7	NDEC	0	1	0	0							
8	WIDTH	16	16	16	16							
9	LABELS 0=Worse, 1=Better 0=None, 1=One child, 2=Two children, 3=Three children, 4=Four children, 5=Five children											
10	DATA	1	2	0	1							
11	DATA	0	3	4	2							
12	DATA	1	37	5	2							
13	DATA	0	0.9	1	3							
14	DATA	1	99	1	1							
15	DATA	1	23.4	5	2							
16	DATA	1	34.5	6	3							
17	DATA	0	65.4	3	4							
18	DATA	0	22	4	1							
19	DATA	1	12.2	2	2							
20	DATA	1	11.1	1	3							
21	DATA	1	0	0	1							

Note: Instructions appear when you move onto a cell.

By default, Excel applies no formatting to a [CSV](#) file; so you might find it easier to make the corrections if you first right justify and autosize column one, and center row 2. Make corrections to the data rows (columns 2-n) and close the file. FIX_DATASET then creates the corrected MicroSiris dictionary and data files.

Use RECODE with TRANSFORM to correct systematic errors or for datasets too large for Excel. For instance, when all occurrences of V1=1 must be changed to 2, use the RECODE statement
"IF V1 EQ 1 THEN V1=2"

DATA MODIFICATION AND INDEX GENERATION

MicrOsiris Recoding

The MicrOsiris recode facility provides recoding of data. RECODE operates on variables within one case, permits you to recode variable categories of one or more variables, to generate new variables from combinations of variables, and provides control of these operations through tests of logical expressions and several specialized statements.

Recoded variables are created with the [RECODE](#) command and subsequent commands refer to them as if they existed in the original dataset. RECODE provides only temporary recoding except when used with the [TRANSFORM](#) to create a permanently recoded dataset.

See [RECODE](#) for a full description of the recode language.

Adding Variables to a File (merging two or more files)

You usually need to do this when processing data from a survey in which the same respondents were interviewed more than once, leading to the creation of separate files for each interview or set of variables. The primary way to do this is to use the [MERGE](#) program, which can easily add variables to datasets as well as update files more generally. The following example illustrates the simple use of MERGE to merge two files:

```
MERGE DATASET1=DATA1 DATASET2=DATA2 DATAOUT=MDATA  
MERGING TWO FILES  
PRINT=(OUTD) PAD=MD1 ID=V1 MATCH=UNION VAR=1-20 ADDVARS=23-29 RENUM=1  
END
```

In the example, the output dataset will contain one case for each unique ID in either dataset (union) and variables will be renumbered beginning with 1. Cases in either file not matching any case in the other will be padded with missing-data values for the missing variables. Both datasets must be in sort order, and must have the same ID variable numbers.

Other kinds of merges are also easy to do, with the option of printing ID values for unmatched records; for merges requiring only complete cases (intersection), specify MATCH=INTER. A full description of these and other options may be found in the write-up for [MERGE](#).

Adding Records to a File (combining two files)

This is required when data collected in two or more batches must be combined to form a complete dataset. In this case, each file must contain records of the same variables and structure. Use the [MERGE](#) command to concatenate two files. You can also use FIX_DATASET with the Excel option to add records as well as correcting existing ones.

REPEAT and Repetition

For some commands you can do any number of analyses on different subsets with a single command using a filter repeat statement or a repetition statement, or both.

FILTER REPEAT statement

A FILTER REPEAT statement is an option to automatically repeat a run for up to 100 subsets defined by a single variable. See [Filter Statement](#) for details.

REPETITION statement

The commands [ANOVA](#), [ANOVAR](#), [DISCRIM](#), [TABLES](#), [T-TEST](#), and [USTATS](#) have a repetition option.

The REPETITION option defines multiple analyses for up to 25 variable categories. Repetition statements provide for labeling the variable categories constructed, and have the format:

Vn=list1[\$name1]/list2[\$name2]/...

A repetition statement defines one or more subsets of cases, where each subset is defined by Vn=listn and a name. One analysis is performed for each subset. Names may be up to 24 characters. The list of acceptable values includes single values and ranges of values between -32767 and +32767. A case not belonging to any repetition is eliminated. Cases falling in more than one repetition are included only in the first such repetition.

Example: (V1=1\$Primary/2-5\$Other)

defines and labels two repetitions, one when variable V1=1, labeled "Primary" and the other when V1 is in the range 2-5, labeled "Other."

USING DATA FROM OTHER SYSTEMS

Using SPSS data in MicroSiris

Create an SPSS portable file with the command: `export out = filename.por`

Or use the "Save As" command from the data editor window and choose SPSS portable as the file type.

SPSS allows 3 discrete missing-data codes or a range of missing-data codes. MicroSiris allows only two missing-data codes--one discrete and one the lower (if positive) or upper (if negative) bound for valid data. The first SPSS missing-data code becomes the MicroSiris first missing data code and the second becomes the MicroSiris second missing-data code (range indicator). If a range is found, MicroSiris sets the beginning of the range as the second missing-data code and ignores the other codes. To avoid problems you should recode all missing-data codes to one discrete code and/or a range indicator before importing to MicroSiris.

SPSS also uses an SPSS system missing-data value for banks in the data set. These values are converted to the first MicroSiris default missing-data code 1.5 billion.

Click on the **IMPORT DATA** button in MicroSiris and choose SPSS. You can choose to bring over the SPSS short names (8 characters, no blanks) or the first 24 characters of the long names. Variable code category values are brought over, if present.

In SPSS, use "Read Text File" option of the file menu to open the text file.

1. Select "no predefined format" (the default) and click next
2. Select delimited (default) and yes for variable names at the top and click next twice
3. Indicate select double quote for the text qualifier and click next
4. Follow the rest of the prompts to save the file.

SPSS does not allow blanks and special characters (for example, blanks, periods, /, \, and parentheses), MicroSiris will change these to underscores when exporting names as part of the file. Other special characters not allowed are automatically changed by SPSS.

To preserve value labels that were imported to MicroSiris with an SPSS file, select Copy Data Properties from the data menu in SPSS and choose the original file as the source file.

Using SAS data in MicroSiris

Use the SAS EXPORT proc to create a CSV file from your SAS dataset, e.g.,

```
PROC EXPORT DATA=DBS, filename  
OUTFILE="c:\microsirir\data\filename.csv"  
  DBMS=DLM REPLACE;  
  DELIMITER=';';  
  PUTNAMES=YES;  
RUN;
```

Click on the **IMPORT DATA** button in MicroSiris and choose **CSV**. Use the ALPHA option to indicate which variables are alphabetic and specify their widths.

Missing data from SAS are converted to the default first missing-data code 1,500,000,000.

Use MicroSiris commands as desired, using filename for the input dataset. (See [WORKFLOW in MicroSiris](#))

Export the data back into SAS if desired:

Use **EXPORT** with the SAS option to create a SAS DATA step Runfile. The file will contain an SAS data step with an INFILE statement pointing to the output file assigned to DATAOUT, an INPUT statement for all selected variables, a LABEL statement for each variable, and a SAS FORMAT statement indicating how many decimal places to display. The SAS filename assigned on the DATA statement will be the first 8 characters of the MicroSiris filename (up to the "." if present).

Using STATA data in MicrOsiris

STATA has a facility to export and import datasets as delimited text, which provides easy data interchange with MicrOsiris.

1. Use the `outsheet` command in STATA to create a comma-delimited text file.

```
outsheet [varlist] using filename, nonames nolabel comma
```

`nolabel` specifies that the numeric values of labeled variables are to be written in the file rather than the label associated with each value., and `comma` specifies comma-separated format rather than the default tab-separated format.

2. Click on the [IMPORT DATA](#) button in MicrOsiris and choose [CSV](#).
3. Use MicrOsiris commands as desired, using filename for the input dataset. (See [WORKFLOW in MicrOsiris](#))
4. Export the data back into STATA if it was changed within MicrOsiris:

Use [EXPORT](#) with the [CSV](#) and NAMES options to create a text file with the suffix `.txt`

5. In STATA, use `insheet` to read the text file.

```
insheet [varlist] using filename.txt
```

`insheet` reads text files where there is one observation per line and the values are separated by commas. The first line of the file can contain the variable names or not.

Using OSIRIS data in MicroSiris

OSIRIS is a general purpose statistical package written for use on IBM mainframes. It is no longer actively supported. However, an enormous store of survey data is available in OSIRIS format from the Inter-University Consortium for Political and Social Research (ICPSR) at the University of Michigan.

OSIRIS datasets are similar to MicroSiris datasets, comprised of a dictionary file and a data file. There are two significant differences:

1. The dictionary file, known as a type 1 dictionary, is a **binary** (internal number storage format) file limited to smaller variable numbers and missing-data codes than the MicroSiris dictionary.
2. Since OSIRIS was developed on IBM mainframes, the dictionary and data files were coded in EBCDIC (Extended Binary Coded Decimal Interchange Code) instead of ASCII, which is used on PCs.

To use OSIRIS datasets in MicroSiris, click on the IMPORT DATA button in MicroSiris and choose OSIRIS. The data file, originally stored in EBCDIC format, usually comes from ICPSR already in ASCII format, but If the data file is in EBCDIC, select the 'Convert data file to ASCII' option.

Using IVEware in MicroSiris

There are three IVEware modules invoked by MicroSiris¹:

IMPUTE uses a multivariate sequential regression approach to imputing item missing values. The **IMPUTE** command returns an imputed data set for further analysis in MicroSiris with other commands, e.g., **SEARCH**, and the IVEware modules **DESCRIBE** and **REGRESS**.

DESCRIBE estimates the population means, proportions, subgroup differences, contrasts and linear combinations of means and proportions and variance estimates appropriate for a user specified complex sample design.

REGRESS fits linear, logistic, polytomous, Poisson, Tobit and proportional hazard regression models for data resulting from a complex sample design, using a Jackknife repeated replication to estimate the sampling variances. See also **REGRESSION** and **LOGIT LINEAR**.

See the **IVEware User Guide** for more information. MicroSiris variable numbers are used as the short 8-character variable names referenced in the Guide. MicroSiris inserts the MicroSiris variable names in the output next to the variable numbers, so you will see both in the MicroSiris output form each module.

Existing IVEware datasets can be exported to MicroSiris with the IVEware command %putdata:

```
%putdata; datain mydata; table mydata.txt; run;
```

The file mydata.txt is then imported with the MicroSiris **IMPORT** command for a .txt file with a tab delimiter.

IVEware was developed by the Survey Methodology Program at The University of Michigan's Survey Research Center, Institute for Social Research.

EXPORTING OUTPUT TO EXCEL AND OTHER SPREADSHEETS

You have the option of saving all output from a command to a CSV file at the close of each command. Numbers, tables, and text are always separated properly into cells.

A CSV file contains data items separated by commas, where each new line represents a new row, and each row has one or more fields separated by a comma. CSV files are commonly used for transferring data between spreadsheets and databases in a simple text-based format. The CSV file format is supported by almost all spreadsheet software such as Excel and OpenOffice.org Calc, although Excel uses the list separator of the current local settings, e.g., a semi-colon instead of a comma for European locales. Many database management systems support the reading and writing of CSV files.

CITING MICROSIRIS SOFTWARE

Referencing Data Analysis Performed using MicroSiris Software:

Your citation for the data analysis you performed with MicroSiris software should contain the name MicroSiris and the version number. An example citation would be as follows:

The data analysis was generated using MicroSiris software, Version 24, Copyright 2014 from www.microsirris.com.

AGGREGATION -- DATA AGGREGATION

File Assignments:	DATASET	Input dataset
	DATAOUT	Output dataset

GENERAL DESCRIPTION

Creates a MicroSiris dataset containing aggregated statistics for subsets of the input dataset.

COMMAND FEATURES

AGGREGATION creates a dataset with a unit of analysis higher than that of the original input dataset. For example, from a dataset of achievement test scores collected for all students in a school district, AGGREGATION could create another dataset with schools as the unit of analysis. Each school record would contain summary information about the test scores of all the students in the school: for example, the mean score on test X, Y and Z, or the highest and lowest scores on test A, B and C.

The output dataset can be used directly for further analysis, or you can use the MERGE command to merge the aggregate statistics with the dataset from which they are derived.

Use AGGREGATION with RECODE to do complex aggregations not possible with AGGREGATION alone.

Missing data are deleted for aggregated variables. Transfer variables are not checked for missing data. Cases for which the weight variable is zero or negative or contains missing data are deleted.

INPUT DATA

All consecutive records with identical values for the ID variables are treated as a subset. Use [**SORT DATASET**](#) first to sort the data by ID variables if necessary. Aggregated variables must be numeric; ID and transfer variables may be alphabetic or numeric.

OUTPUT DATA

The output dataset is a MicroSiris dataset consisting of one record for each subset of the input dataset. The contents of each record are: the ID variables and any requested transfer variables, the N for the subset, the sum of weights (if weighted), and the requested aggregated variables.

The diagram below illustrates a simple aggregation, in which V10 is the ID variable and the only aggregated variable is the sum of V12. Input records 1-4 are summarized in output record 1; input records 5-6 are summarized in output record 2; and input records 7-9 are summarized in output record 3.

Input Dataset		Output Dataset		
V10	V12	V10	V11(N)	V12(V12 Sum)
1	10	1	4	72
1	30	3	2	12
1	12	5	3	54
1	20			
3	2			
3	10			
5	16			
5	18			
5	20			

Output variable numbers are determined by the VSTART option.

Variable numbers: ID and transfer variables retain their original numbers; aggregate variables are numbered according to the VSTART option. The order of the variables is: 1) the ID and transfer variables; 2) the N and sum of weights (if weighted) for the subset; and 3) the aggregated variables in order of their appearance in the setup.

Aggregated variables:

Variable names are the same as that of the corresponding input variable, except that a code for the type of statistic is appended. The codes are:

Code	Definition
MAX	Maximum
MEAN	Mean
MIN	Minimum
N	Number of cases in the subset
STD	Standard Deviation
SUM	Sum
VAR	Variance
WTN	Sum of Weights

Number of decimal places:

Statistic	Decimal Places
N	0
WTN	Same as input weight variable
MIN, MAX, SUM	Same as corresponding input variable
STD,VARIANCE,MEAN	2 + NDEC of the corresponding input variable

The dictionary number of decimal places only indicates decimal places for output display.

Missing-data codes: The first missing-data code is 1.5 billion. The second missing-data code is 1.6 billion.

OPTIONS

Choose AGGREGATION from the command screen and make selections.

ID=(variable list)

List of up to 10 variables which define the subset. All consecutive records having the same values for these variables constitute a subset.

Default: None. An ID variable list must be supplied.

PRINT=(DICT,OUTD)

DICT Print the input dictionary.

OUTD Print the output dictionary.

RECODE=n Use RECODE n, previously entered via the RECODE command.

TRANSFER=(variable list)

Output the value on the 1st record of a subset for each variable in the list.

Default: Don't transfer any values; only the statistics requested with the STATS option will appear in the output dataset with the ID variables.

VSTART=n Number the aggregated variables consecutively starting with the larger of Vn, one more than the highest transfer variable, the ID variable number. *Default:* One more than the highest transfer variable number or ID variable number.

WT=n Use variable n as a weight variable.

VARS=(list) Use the variables specified in the list.

STATS=(MAX,MEAN,MIN,N,STD,SUM,VAR,WTN)

For each subset, selected statistics are computed for each variable specified by the VARS option.

N	Number of non-missing data values.
WTN	Sum of weights on non-missing-data.
MIN	Minimum value.
MAX	Maximum value.
SUM	Sum.
MEAN	Mean
VAR	Sample variance.
STD	Sample standard deviation.

Default: None; must be specified.

EXAMPLE

Computing summary statistics for variable 12 for each subgroup. LIST DATASET is used at the end to list the AGGREG output dataset.

*** DATA AGGREGATION ***

Dataset AGGREG

Creating dataset AGGREG2

Computing subgroup statistics N, Maximum, Sum, Mean, Variance, Standard Deviation
For variables V12

First aggregate variable number is V11

ID variable is V10

INPUT DICTIONARY

	TYPE	LOC	WID	DEC	MD1	MD2
V10 VARIABLE 10	C	1	1	0		
V12 VARIABLE 12	C	2	7	0		

OUTPUT DICTIONARY

	TYPE	LOC	WID	DEC	MD1	MD2
V10 Variable 10	C	1	1	0		
V12 Variable 12	C	2	7	0		
V13 N for subset	F	17	8	0		
V14 N Variable 12	F	25	8	0		
V15 MAX Variable 12	F	33	8	0		
V16 SUM Variable 12	F	41	8	0		
V17 MEAN Variable 12	F	49	8	2		
V18 VAR Variable 12	F	57	8	2		
V19 STD Variable 12	F	65	8	2		

For the whole dataset:

9 cases accepted

3 output records

*** DATASET LISTING ***

Listing the aggregated data

Dataset C:\DEVELOPMENT\PROJECTS\TESTDATA\AGGREG\AGGREG2

	V10	V12	V11	V12	V13	V14	V15	V16	V17
	Variable	Variable	N for	N	MAX	SUM	MEAN	VAR	STD
RECORD	10	10	subset	Variable	Variable	Variable	Variable	Variable	Variable
				12	12	12	12	12	12
1	1	1	4	4	30	72	18.00	82.67	9.09
2	3	3	2	2	10	12	6.00	32.00	5.66
3	5	5	3	3	20	54	18.00	4.00	2.00

3 CASES READ

3 CASES LISTED

ANOVA -- ONE WAY ANALYSIS OF VARIANCE

File Assignments:	DATASET	Input dataset
-------------------	---------	---------------

GENERAL DESCRIPTION

One-way analysis-of-variance for single factor designs. See [ANOVAR](#) for repeated measures ANOVA. See [MANOVA](#) for multifactor univariate analysis of variance.

COMMAND FEATURES

Multiple Analyses: You can do any number of analyses with one command. One table is produced for each possible combination of the dependent and control variables. Additional control is provided by the REPETITION option, used to define multiple analyses for up to 25 categories.

Missing Data: For each analysis, cases with missing data on the dependent (outcome) variable are excluded; cases with missing data on the treatment variable are optionally excluded. You can use the [IVEWARE](#) command to first impute missing data if desired.

PRINTED OUTPUT

Analysis of Variance Statistics

For each analysis, the command prints

- The total sum of squares of the dependent variable
- Eta
- Eta squared
- Eta (adjusted)
- Eta squared (adjusted)
- Omega squared
- Omega squared effective size
- The sum of squares between groups
- The sum of squares within groups
- The F-ratio and its associated probability (for unweighted data only)

Descriptive Statistics within Categories of the Treatment Variable

For each analysis, a table is printed giving the following information for each code value of the treatment variable:

- Number of valid cases
- Sum of weights
- Sum of weights as a percentage of table sum
- Mean of dependent variable scores
- Standard deviation of dependent variable scores

Sum of dependent variable scores
Sum of dependent variable scores as a percentage of table sum
Sum of dependent variable scores squared

OPTIONS

Choose ANOVA from the command screen and make selections.

For a Runfile use:	ANOVA Filter statement (optional) Job Title Keyword choices from below
--------------------	---------------------------------------------------------------------------------

MAXC=n Largest code of the independent variable allowed. Codes less than 0 or greater than MAXC are eliminated. (Large values use a lot of memory).
Default: MAXC=10. MAXC must be less than 1000.

PRINT=DICT DICT Print the input dictionary.

RECODE=n Use RECODE n, previously entered via the RECODE command.

REPETITION=(Vn=list1[\$name1]/list2[\$name2])
Defines up to 25 subsets of cases where each subset is defined by Vn=listn.
One analysis is done for each subset. Example: REPE= (V1=1\$Primary/2-5\$Other) defines and labels two repetitions, one for V1=1 and the other for V1 in the range 2-5. See [REPEAT and REPETITIONS](#) for more information.

WT=n Use variable n as a weight variable.

Analysis Statements

More than one statement may be supplied, followed by an END statement.

DELETE=(MD1,MD2)
MD1 Delete all cases where any independent variable equals its first missing-data code.
MD2 Delete all cases where any independent variable equals its second missing-data code

DEPV=(variable numbers) List of dependent (outcome) variables.

VARS=(variable numbers)
List of treatment variables. These are the variables that define unique treatment groups (e.g. placebo=0, therapy=1).

EXAMPLE

The dependent variable is Family Income (V268) and the treatment variable is Size of Car (V193). Families with no children (V26) and Race code 1 (V37) are selected with the Filter.

*** ONE WAY ANALYSIS OF VARIANCE ***

Survey of Consumer Finances

Dataset SCF

Including V37=1 AND V26=0 AND V193=<9

165 cases accepted

TABLE 1

CONTROL VARIABLE V193 SIZE OF CAR
DEPEND. VARIABLE V268 TOTAL FAMILY INC

	CODE	N	%	MEAN	STANDARD DEVIATION	SUM OF X	%	SUM OF SQUARES
No Car	0	32	19.4	2988.969	2960.976	9.564700E+04	6.4	5.576746E+08
Small	1	9	5.5	12111.333	8559.955	1.090020E+05	7.3	1.906342E+09
Compact	2	12	7.3	11707.500	6491.802	1.404900E+05	9.4	2.108365E+09
Mid-Size	3	19	11.5	10486.842	7004.660	1.992500E+05	13.3	2.972678E+09
Large	5	93	56.4	10272.559	8545.153	9.553480E+05	63.7	1.653168E+10
TOTAL		165	100	9089.315	7980.599	1.499737E+06	100	2.407674E+10

TOTAL SUM OF SQUARES	1.0445154E+10	
ETA	0.3818024	
ETA Squared	0.1457731	
ETA (adjusted)	0.3527285	
ETA Squared (adjusted)	0.1244174	(Kelley's epsilon)
OMEGA Squared	0.1237567	
OMEGA Squared effect size	0.3758131	
Between means sum of squares	1.5226220E+09	
Within groups sum of squares	8.9225319E+09	
F(4,160)	6.826	
Probability (F)	0.000	

ANOVAR -- ONE WAY ANALYSIS OF VARIANCE WITH REPEATED MEASURES

File Assignments:	DATASET	Input dataset
-------------------	---------	---------------

GENERAL DESCRIPTION

One-way analysis-of-variance with repeated measures, using a GLM approach. See [MANOVA](#) for multifactor univariate analysis of variance.

COMMAND FEATURES

Missing Data: Cases with missing data on any variable are excluded. You can use the [IVEWARE](#) command to first impute missing data if desired.

PRINTED OUTPUT

Analysis of Variance Statistics

Means and standard deviations
Standard ANOVA summary table
The F-ratio and its associated probability

OPTIONS

Choose ANOVAR from the command screen and make selections.

For a Runfile use:	ANOVAR Filter statement (optional) Job Title Keyword choices from below
--------------------	----------------------------------------------------------------------------------

RECODE=n Use RECODE n, previously entered via the RECODE command.

REPETITION=(Vn=list1[\$name1]/list2[\$name2

Defines up to 25 subsets of cases where each subset is defined by Vn=listn.
One analysis is done for each subset. Example: REPE= (V1=1\$Primary/2-5\$Other) defines and labels two repetitions, one for V1=1 and the other for V1 in the range 2-5. See [REPEAT and REPETITIONS](#) for more information.

WT=n Use variable n as a weight variable.

VAR=(variable numbers) List of variables to compare.

REFERENCES

Rutherford, Andrew. *Introducing ANOVA and ANCOVA, a GLM Approach*. Sage Publications Ltd., 2001. Reprinted 2007.

EXAMPLE

Comparing v1, v2, and v3 (conditions A, B, and C).

*** ANOVAR -- ONE WAY ANALYSIS OF VARIANCE WITH REPEATED MEASURES ***

Dataset GLM

Condition variables

V1 Condition A
V2 Condition B
V3 Condition C

8 cases accepted

Means and Standard Deviations

		V1	V2			
			Standard			
		Mean	Deviation			
Condition A	V1	6.000	1.512			
Condition B	V2	10.000	1.414			
Condition C	V3	11.000	1.773			

Source of variation	SS	DF	MS	F	P(F)
Subjects	14.000	7	2.0000		
Experimental conditions	112.00	2	56.000	20.632	0.000
Error	38.000	14	2.7143		
Total	164.00	23			
Eta Squared	0.74667				

Post Hoc tests	Comparison	Mean Difference	T	P(T)	P(Bonferroni)
Condition A	Condition A and Condition B	4.0000	4.8558	0.043	0.128
	Condition A and Condition C	5.0000	6.0698	0.026	0.078
Condition B	Condition B and Condition C	1.0000	1.2140	0.289	0.868

CANCORR -- CANONICAL CORRELATION

File Assignments:	DATASET	Input dataset (conditional)
	MATIN	Input matrix (conditional)

GENERAL DESCRIPTION

Computes canonical correlation for partitions of a matrix of Pearson correlation coefficients.

PRINTED OUTPUT

Matrix of correlations.

Canonical weights. The left and right partition canonical weights are printed.

Factor Structures. The left and right partition factor structures are printed with factors as columns and rows as tests.

Variances. The total and redundant proportional variances are printed for each factor of each partition.

Multivariate Statistics:

Wilks' Lambda

Chi- Square and estimated probability of Chi-Square occurring by chance.

Pillai's Trace

Hotelling-Lawley Trace

Roy's Greatest Root

Chi Square tests with successive roots removed

INPUT DATA

Matrix of Pearson correlation coefficients or raw data to compute one. A Pearson correlation coefficient assumes that the variables were measured on an interval scale.

OPTIONS

Choose CANCORR from the command screen and make selections.

For a Runfile use:	CANCORR
	Filter statement (optional)
	Job Title
	Keyword choices from below

MATRIX=n The number of a correlation matrix produced by CORRELATIONS or assigned to MATIN.

Default: Raw data input.

VARs=(list 1/list 2)

Two lists of variable numbers divided by a slash(/). List one defines partition 1, list two defines partition 2. *There must be the same number of variables in each list.*

Options for Raw Data Input (MATNO=0)

DELETE=PAIRS|CASES

PAIRS Pair-wise deletion.

CASES Case-wise deletion.

Default: DELETE=PAIRS.

RECODE=n Use RECODE n, previously entered via the RECODE command.

WT=n Use variable n as a weight variable

REFERENCES

Cooley, W.H. and Paul R. Lohnes. *Multivariate Analysis*. Wiley, New York, 1971.

Overall, J. E. and C. J Klett. *Applied Multivariate Analysis*. New York: McGraw-Hill, 1972.

Reprinted: Krieger, 1983. pp. 321, 430, 441-68.

EXAMPLE

Conventional symmetrical matrix with partitions v1-v2/v3-v4.

```
*** CANCORR -- CANONICAL CORRELATION ANALYSIS ***

Using matrix 1, CORRELATIONS, based on 100 cases from CANCORR.MTX

Number of variables in left partition: 2

Number of variables in right partition: 2

Number of cases: 100

CORRELATIONS
          V1      V2      V3      V4
      Better  Income  Children Weight 1
      or Worse (000)
Better or Worse V1  1.000  0.400  0.500  0.600
Income (000)    V2  0.400  1.000  0.300  0.400
Children        V3  0.500  0.300  1.000  0.200
Weight 1        V4  0.600  0.400  0.200  1.000

CANONICAL WEIGHTS

Left Partition, column-wise

      1      2
1  0.856  0.677
2  0.278 -1.055
```

Right Partition, column-wise

	1	2
1	0.545	0.863
2	0.737	-0.706

FACTOR STRUCTURE FOR LEFT PARTITION

	1	2
1	0.967	0.255
2	0.620	-0.785

Factor	Variance Proportion	Redundancy
1	0.660	0.360
2	0.340	0.000

Total redundancy for left set, given right set = 0.360

FACTOR STRUCTURE FOR RIGHT PARTITION

	1	2
1	0.692	0.722
2	0.846	-0.534

Factor	Variance Proportion	Redundancy
1	0.597	0.326
2	0.403	0.000

Total redundancy for right set, given left set = 0.326

MULTIVARIATE STATISTICS

Wilks' lambda	.45386889
Pillai's Trace	.54662714
Hotelling-Lawley Trace	1.2021866
Roy's Greatest Root	1.2012768

Chi-Square 76.229877 D.F. = 4 PROB = 0.000

Chi-Square tests with successive roots removed

Roots Removed	Canonical R	R-squared	Chi-Square	D.F.	Wilks' Lambda	Prob
0	0.7387	0.546	76.23	4	0.454	0.000
1	0.0301	0.001	0.09	1	0.999	0.767

CAP -- CONFIGURATION ANALYSIS

File Assignments:	MATIN	Input matrix
	MATOUT	Output Configuration (optional)

GENERAL DESCRIPTION

Performs spatial configuration analysis. It optionally centers, normalizes, rotates, and translates dimensions, and can provide varimax rotation and principal axis solutions. A matrix of inter-point scalar products and inter-point distances may be computed.

SPECIAL USES

You can use CAP to re-orient a configuration in order to better compare it with other configurations. Rotation, for example, may make a configuration easier to interpret. You can also use CAP to further analyze a configuration matrix output by FACTOR ANALYSIS and multidimensional scaling programs. Output from CAP can serve as input to CLUSTER.

SPECIAL TERMINOLOGY

Configuration matrix: Coordinates of each point in a configuration are given in a row-by-column format. Each row of a configuration matrix provides the coordinates of one point of the configuration. Thus the number of rows equals the number of points (variables), while the number of columns equals the number of dimensions.

PRINTED OUTPUT

Centered configuration: (Optional: see option CENTER.) If you specify CENTER and the input configuration is already centered, CAP prints only the message "Input configuration already centered."

Normalized configuration: (Optional: see option NORMALIZE.) If you specify NORMALIZE and the input configuration is already normalized, CAP prints only the message "Input configuration already normalized."

Principal axis solution: (Optional: see option PRAXIS.) The rows of the matrix are the points and the columns are the axes. The elements in the matrix are the projections of the points on the axes.

Scalar products matrix: (Optional: see option SCALAR.) The lower-left half of the symmetric matrix is printed. Each element of the matrix is the scalar product (or dot or inner product) for a pair of points (variables).

Inter-point distances: (Optional: see option DISTANCES.) The lower half of the symmetric matrix is printed. Each element in the matrix is the distance between a pair of points (variables).

Rotation specifications: (Optional: see option ROTATE.)

Rotated or translated configuration: (Optional: see option PRINT)

Varimax rotation: (Optional: see option VARIMAX.) This is the configuration matrix after rotation to maximize the normal varimax criterion. It will have the same number of rows and columns as the input configuration matrix.

Sorted configuration: (Optional: see option ORDER.) Each column of the configuration matrix, after being ordered, is printed horizontally across the page.

Plot identification table: (Optional: see option PLOT.) For each plot of a transformed configuration, a table is printed showing the correspondence between the label in the plot and the true variable number.

Plot of the rotated or translated configuration: (Optional: see option PLOT.) At each requested step, the transformed configuration is plotted two axes at a time.

INPUT DATA

The input to CAP is one configuration matrix consisting of one row for each variable in the configuration with the columns representing the dimensions 1 through n. An input matrix may be taken from the MATIN file assignment or passed from FACTOR_ANALYSIS.

OUTPUT DATA

Output Configuration Matrix: When WRITE is specified, the output configuration matrix is written to the file assigned to MATOUT. It is assigned the input matrix name followed by "(adjusted by CAP)".

RESTRICTIONS

CAP treats all values of the input matrix as valid, i.e., missing-data codes are not recognized.

If the transformations ROTATE or TRANSLATE are specified, they are performed only in the last step, i.e., a configuration may be centered and then transformed, but not transformed and then centered.

OPTIONS

Choose CAP from the command screen and make selections.

CENTER	Shift origin to centroid of space.
DISTANCES	Compute matrix of inter-point distances.
MATRIX=n	The number of the configuration matrix produced by FACTOR ANALYSIS or from the MATIN file assignment. <i>Default:</i> n=1

NORMALIZE Alter size of space so sum of squared loadings equals the number of variables.

PRAXIS Perform principal axis rotation.

SCALARS Compute matrix of scalar products.

VARIMAX Perform orthogonal (varimax) rotation

ORDER Sort and print the configuration.

PRINT=EACH The configuration is printed as it was input and after each separate analysis step. (e.g., if CENTER, NORMALIZE, PRAXIS, and VARIMAX are specified, five configurations are printed: input, centered, normalized, principal axes and varimax.) The input and the final configurations are always printed.

PLOT=EACH The configuration is plotted as it was input and after each separate analysis step requested. The final configuration is always plotted.

WRITE Write the output configuration matrix to the file assigned to MATOUT.

OUTM=n Assign n to the output configuration matrix when WRITE is specified.
Default: OUTM=999.

ROTATE=(DEGREES=d,DIM=n,m)
 d: The angle of rotation in degrees.
 n,m: Two dimensions to be rotated (only pairwise rotation).

TRANSLATE=(ADD=n,DIM=m)
 n: The value to add to each coordinate on dimension m (may be negative and have decimal places).
 m: The dimension to be translated.

EXAMPLE

45 degree rotation of the 85th Congress Roll Call Analysis configuration matrix.

```

*** CONFIGURATION ANALYSIS ***

85th Congress Roll Call Analysis

Using input matrix 1

Options selected:

    Centering
    Normalizing
    Varimax rotation

Input Configuration Matrix:

```

					1	2
APR04	1957	HR	6387	V27	-0.18500	-0.95300
APR04	1957	HR	6287	V29	-0.35200	-0.86400
APR05	1957	HR	6387	V30	-0.39300	-1.21600

APR17	1957	HR	6871	V40	0.86700	-1.14200
JUN05	1957	HRE	259	V50	-0.71100	0.80300
JUN18	1957	HR	6127	V51	-0.75000	0.56300
JUN05	1957	HR	6127	V52	-0.66300	0.61800
JUL25	1957	HR	1	V66	-0.11900	0.60200
AUG08	1957	HR	8992	V80	-0.39600	-1.38300
AUG14	1957	S	2130	V89	1.05800	0.03000

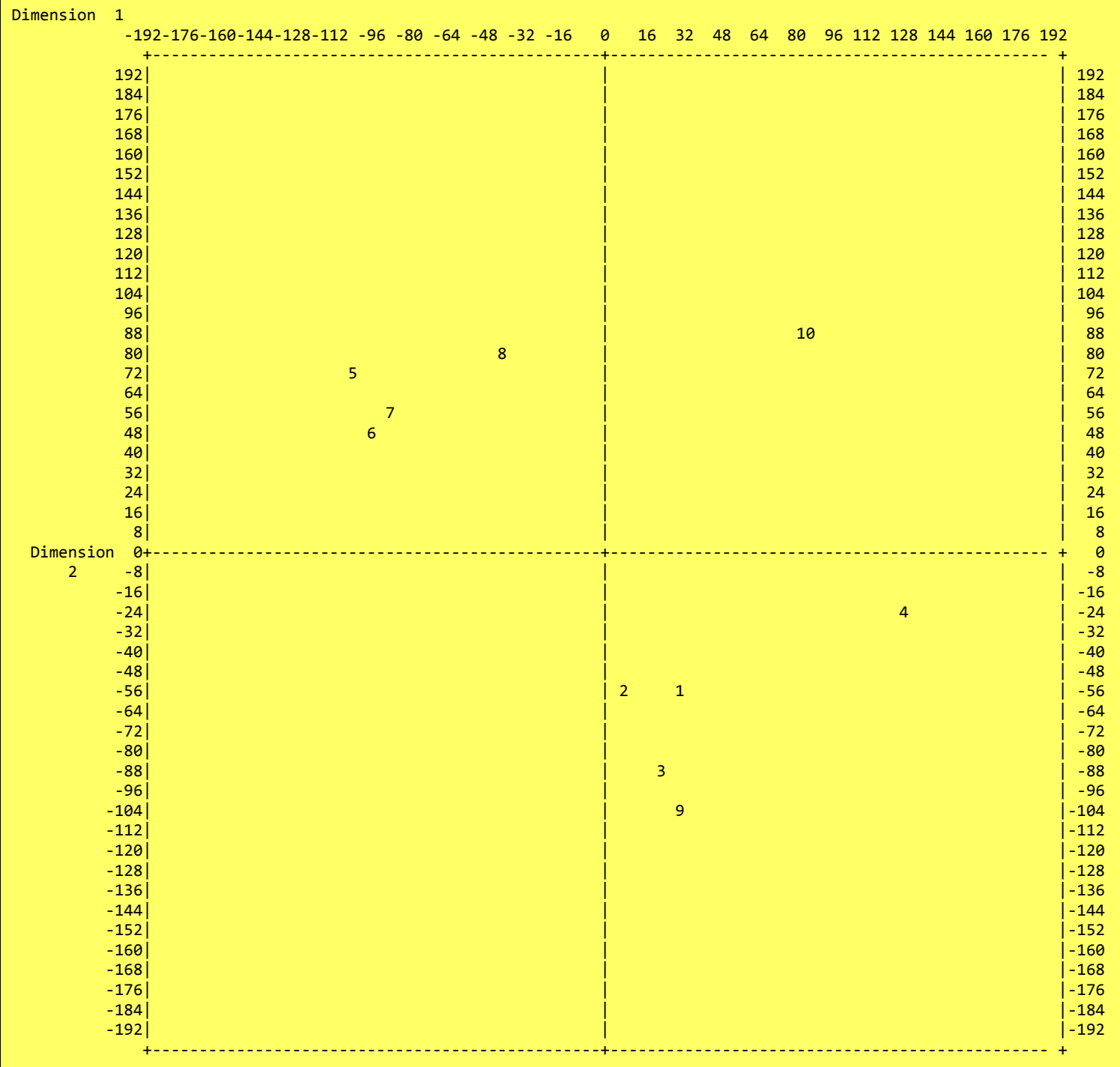
Varimax Rotation History

0.16443
0.21722
0.21722
0.21722
0.21722
0.21722

Varimax Solution Configuration

					1	2
APR04	1957	HR	6387	V27	0.30754	-0.58298
APR04	1957	HR	6287	V29	0.11838	-0.58809
APR05	1957	HR	6387	V30	0.25662	-0.91440
APR17	1957	HR	6871	V40	1.31558	-0.22761
JUN05	1957	HRE	259	V50	-1.01727	0.68394
JUN18	1957	HR	6127	V51	-0.93262	0.45601
JUN05	1957	HR	6127	V52	-0.88414	0.54681
JUL25	1957	HR	1	V66	-0.40326	0.80164
AUG08	1957	HR	8992	V80	0.33651	-1.06108
AUG14	1957	S	2130	V89	0.90267	0.88575
Sum of Squares					5.68218	5.10142

Plot	Label	Variable	Name
1		V27	APR04 1957 HR 6387
2		V29	APR04 1957 HR 6287
3		V30	APR05 1957 HR 6387
4		V40	APR17 1957 HR 6871
5		V50	JUN05 1957 HRE 259
6		V51	JUN18 1957 HR 6127
7		V52	JUN05 1957 HR 6127
8		V66	JUL25 1957 HR 1
9		V80	AUG08 1957 HR 8992
10		V89	AUG14 1957 S 2130



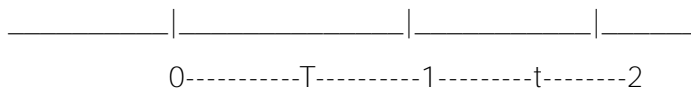
CHANGE RESPONSE UNCERTAINTY ANALYSIS

File Assignments:	TABLE1	Input matrix
	TABLE2	Input matrix
	PATTERN	Pattern matrix

GENERAL DESCRIPTION

J. S. Coleman developed models of change and response uncertainty to study the distributions of opinions in populations. For example, if 25% of a population buys brand X in a visit to the grocery store or responds "yes" to a question, how should this be interpreted in terms of the distribution of persons? Does it mean that there is a fixed 25 percent who like brand X or respond "yes" with a fixed 75 opposed? Or does it meant the each person will respond this way about 25 percent of the time and give the opposite response about 75 percent of the time? This cannot be determined from the single response alone--it is confounded with the individual and one cannot separate the distribution of individuals from the distribution of responses. Two or more responses from the same individual are required. CHANGE_RESPONSE seeks to help answer this question.

Input to the command is two tables that cross-tabulate responses between time 0-1 (T) and 0-2 (t). The time period between 0 and 1 need not be the same as that separating 1 and 2.



See Coleman, 1964, page 58.

The TABLES command can create and save these tables for input to CHANGE_RESPONSE.

SPECIAL TERMINOLOGY

- P_{it} the proportion of individuals in state i at time t .
- $P_{i_0j_1}$ the proportion of individuals giving response i at time 0 and response j at time 1.
- R_{ijt} the probability of change from state i to state j in the time period t .
- Q_{ij} The transition rate for each element from state i to state j .
- V_{it} the probability of an element being in state i at time t .

PRINTED OUTPUT

Input tables of responses.

Transition rates and probabilities: Q-matrix and R-matrix

Response uncertainties at time 0

Standard reliability estimates: $\frac{P_{i0j_1}}{Sqrt(P_{i0}P_{j_0})}$ See Coleman (1964), page 86)

$$Sqrt(P_{i0}P_{j_0})$$

Covariances : $C(V_{i0}V_{j_0}) = P_{i0j_0} - P_{i0}P_{j_0}$ These measure uncertainty of response.

M^* The number of independent elements governing a response. See Coleman (1964) page 63.

Contagion parameters for elements: $C_{ij} = 1 / (M_{ij}^* - 1)$. The response uncertainty of pairs of responses, disregarding the other categories.

Projections. These are associated with each dichotomy requested.

Equilibrium, movement, and polarization.

Actual and calculated proportions. The actual proportions are those obtained after collapsing the original tables into dichotomies. The calculated proportions are those obtained using transition rates.

Average individual change. This is the average individual change over time period t , the time between the second and third observations.

INPUT DATA

Two tables cross-tabulating responses between times 0-1 and 0-2, and a pattern matrix of 0s and 1s matching the dimensions of the tables and indicating where positive transitions are desired.

OPTIONS

Choose CHANGE_RESPONSE from the command screen and make selections.

PRINT=ALL Print intermediate items.
Default: Print only the input matrices, initial Q matrices, transition matrices and projections.

TAB1=n The number of the Table 1 matrix.
Default: TAB1=1.

TAB2=n The number of the Table 2 matrix.
Default: TAB2=2.

PAT=n The number of the positive transitions desired pattern matrix or 0s and 1s.
Default: PAT=3.

T=(list of numbers)
Transition time periods, e.g., .3333, 1.3333, .5, 1.5, 1 Last must be 1.

TS=(list of numbers)
Transition time periods for projection, e.g., 0 .333, 1, 1.5, 5

D1=(list) Dichotomy 1. List of row numbers to lump together to create dichotomy 1.

D2-D5 Up to 4 more dichotomies may be specified.

REFERENCES

Coleman, J.S. Models of Change and Response Uncertainty. Englewood Cliffs, New Jersey: Prentice Hall, 1964.

EXAMPLE

TAB1=1 TAB2=2 PAT=3 T=(.333,1.333,.5,1.5,1) TS=(0,.333,1,1.5,5) D1=(1,2) D2=(1,3)

*** CHANGE AND RESPONSE UNCERTAINTY ANALYSIS ***					
Using matrix 1, TABLE 1, based on 0 cases from TABLE1.MTX					
Using matrix 2, Table 2, based on 0 cases from TABLE2.MTX					
Using matrix 3, Pattern, based on 0 cases from PATTERN.MTX					
TABLE 1					
	1	2	3	4	Total
1	87.000	21.000	14.000	3.000	125.000
2	6.000	60.000	1.000	29.000	96.000
3	24.000	8.000	93.000	24.000	149.000
4	11.000	38.000	14.000	128.000	191.000
Total	128.000	127.000	122.000	184.000	561.000
Table 2					
	1	2	3	4	Total
1	75.000	24.000	12.000	14.000	125.000
2	17.000	46.000	2.000	31.000	96.000
3	44.000	16.000	58.000	31.000	149.000
4	17.000	49.000	25.000	100.000	191.000
Total	153.000	135.000	97.000	176.000	561.000
POSITIVE TRANSITION PATTERN MATRIX					
	1	2	3	4	
1	0	1	1	0	
2	1	0	0	1	
3	1	0	0	1	
4	0	1	1	0	
Transition time periods:					
	0.3333	1.3333	0.5000	1.5000	1.0000
Transition time periods for projections:					
	0.0000	0.3330	1.0000	1.5000	5.0000
DICHOTOMIES					
1: 1 2					
2: 1 3					

FOR TABLE 1:

Q-MATRIX (TRANSITION RATES), NUMBER OF ITERATIONS: 6

	1	2	3	4
1	-0.433	0.262	0.171	0.000
2	0.096	-0.570	0.000	0.473
3	0.248	0.000	-0.486	0.238
4	0.000	0.311	0.110	-0.421

FOR TABLE 2:

Q-MATRIX (TRANSITION RATES), NUMBER OF ITERATIONS: 16

	1	2	3	4
1	-0.572	0.384	0.189	0.000
2	0.341	-1.105	0.000	0.763
3	0.605	0.000	-1.066	0.460
4	0.000	0.599	0.301	-0.900

TRANSITION PROBABILITIES

	1	2	3	4
1	0.780	0.199	-0.045	0.005
2	0.181	0.631	-0.158	0.009
3	0.259	0.040	0.622	0.218
4	-0.016	0.174	0.178	0.754

FOR 3-WAVE PANEL:

Q-MATRIX (TRANSITION RATES), NUMBER OF ITERATIONS: 7

	1	2	3	4
1	-0.210	0.282	-0.072	0.000
2	0.361	-0.384	0.000	0.023
3	0.334	0.000	-0.673	0.339
4	0.000	0.238	0.291	-0.529

P(I0,J0) USING P(I0,J1) AND TRANSITION PROBABILITIES FOR TIME 0-1

	1	2	3	4
1	0.162	0.022	0.011	0.002
2	-0.002	0.117	-0.004	0.052
3	0.024	0.009	0.201	0.031
4	0.012	0.054	0.003	0.266

P(I0,J0)/(P(I0)*P(J0))

	1	2	3	4
1	0.726	0.113	0.046	0.007
2	-0.012	0.684	-0.018	0.216
3	0.097	0.042	0.755	0.103
4	0.043	0.226	0.009	0.781

P(I0,J0) USING P(I0,J2) AND TRANSITION PROBABILITIES FOR TIME 0-2

	1	2	3	4
1	0.210	-0.045	-0.095	0.056
2	-0.001	0.108	-0.045	0.080
3	0.025	-0.006	0.210	0.031
4	-0.009	0.047	-0.035	0.328

P(I0,J0)/(P(I0)*P(J0))

	1	2	3	4
1	0.941	-0.229	-0.392	0.205
2	-0.007	0.629	-0.209	0.330
3	0.104	-0.028	0.792	0.103
4	-0.033	0.195	-0.115	0.963

PROJECTION FOR DICHOTOMY 1

DAYS IN FUTURE	0.00	0.33	1.00	1.50	5.00
	-0.1943	-0.0881	0.0851	0.1868	0.5246
	0.0740	0.0964	0.0647	0.0460	-0.0158

EQUILIBRIUM P = .6276 MOVEMENT (K) = .4154 POLARIZATION (C) = 1.7073

ACTUAL AND CALCULATED PROPORTIONS

	P10	P11	P12	P1010	P1011	P1012
ACTUAL	0.39	0.45	0.51		0.31	0.29
CALCULATED	-0.19	-0.09	0.57	0.31	0.31	0.29

AVERAGE INDIVIDUAL CHANGE

POSITIVE = 3.3056 NEGATIVE = -7.7554 TOTAL = 11.0611

PROJECTION FOR DICHOTOMY 2

DAYS IN FUTURE	0.00	0.33	1.00	1.50	5.00
	-1.0038	-0.8400	-0.5655	-0.3989	0.2062
	-0.2975	-0.2393	-0.2758	-0.2980	-0.3784

EQUILIBRIUM P = .4456 MOVEMENT (K) = .3601 POLARIZATION (C) = 3.3738

ACTUAL AND CALCULATED PROPORTIONS

	P10	P11	P12	P1010	P1011	P1012
ACTUAL	0.49	0.45	0.45		0.39	0.34
CALCULATED	-1.00	-0.84	-0.14	0.43	0.39	0.34

AVERAGE INDIVIDUAL CHANGE

POSITIVE = 4.3835 NEGATIVE = -4.8049 TOTAL = 9.1884

CHART -- EXCEL CHART INTERFACE

File Assignments:	DATASET	Input dataset
-------------------	---------	---------------

Displays up to 5 selected variables as an Excel spreadsheet for creating charts.

The data are converted into an intermediate [CSV](#) file and passed to Microsoft Excel (assuming you have Excel or the Microsoft Works equivalent.) You can then select the columns (variables) desired and use the Insert Chart function to create a chart and format the chart as desired with Chart Tools.

Options are selected from an interactive window. The following choices are:

RECODE=n Use RECODE n, previously entered via the RECODE command.

VARS=(variable numbers) |ALL

CLUSTER

File Assignments:	DATASET	Input dataset (conditional)
	MATIN	Input matrix (conditional)

GENERAL DESCRIPTION

Clustering is used to find simple structure in a mass of data. CLUSTER attempts this by classifying objects into optimally homogeneous groups. It is primarily a descriptive or exploratory tool in contrast with statistical tests used for inferential purposes.

CLUSTER is primarily used with dissimilarity matrices of dimension 100 or less. For data mining purposes with large datasets consider using SEARCH, which uses a binary segmentation procedure to develop a predictive model for a dependent variable and is a much better option than CLUSTER.

COMMAND FEATURES

Clustering algorithms operate on either of two types of input structures:

- 1) n objects, such as people by means of m attributes (variables), such as marital status, sex, and number of children; represented by an n by m dataset.
- 2) A collection of proximities that must be available for all pairs of objects, represented by an n x n matrix.

The clustering scheme used can be by 1) partitioning, which constructs k clusters, or 2) hierarchical, where initially each variable is a separate cluster; successive clusters are made by combining a single pair of clusters from the previous level to form a new cluster.

The choice of a clustering algorithm depends both on the type of data available and on the particular purpose. Two of the methods, K-means and Ranks, accept datasets of any size. The Monothetic option requires a dataset of zeros and ones. The rest accept small or sampled measurement datasets and dissimilarity matrices.

When the input is a dataset, means and standard deviations are displayed and the cluster numbers are optionally merged with the input dataset to create an output dataset.

Partitioning Methods

K-means

The aim of K-means is to divide M points in N dimensions into K clusters so that the within-cluster sum of squares is minimized. The algorithm seeks "local" optimal solutions such that no movement of a point from one cluster to another will reduce the within-cluster sum of squares. It is intended for situations in which you have already determined the expected number of clusters (n), perhaps via SEARCH.

Ranks method

It sums the ranks of each predictor, sorts the data by the sum of ranks, and divides the dataset by the n largest differences in the sum of ranks distribution. It is intended for situations in which you have already determined the expected number of clusters (n), perhaps via SEARCH.

K-medoids

Clusters objects that are measured on p interval-scaled variables.

The k-medoid method selects k representative objects and assigns each remaining object to the nearest representative object. The representative objects are chosen so that the average distance of the object to all the other objects of the same cluster is minimized. The optimal representative object is called the *medoid* of its cluster, and the method of partitioning around medoids is called the *k-medoid technique*.

The *k-medoid* method tries to find "spherical" clusters, that is, clusters that are roughly ball-shaped. It is therefore not suited to discover drawn-out clusters. The method is especially recommended if one is also interested in the representative objects themselves, which may be very useful for data reduction or characterization purposes. This option allows a more detailed analysis of the partition by providing clustering characteristics and a graphical display (*silhouette plot*); an appropriate choice of k can be made on the basis of a validity index it computes.

The K-medoid method is more robust than k-means with respect to outliers and can also be applied when the input data is a dissimilarity matrix, but datasets of more cases than the maximum matrix dimension specified in Preferences (default 100) must be [sampled](#). You can also specify a minimum and maximum number of clusters.

The option FUZZY avoids "hard" decisions. Instead of saying "object a belongs to cluster 1," FUZZY can say that "object a belongs 90% to cluster 1, 5% to cluster 2, and 5% to cluster 3," meaning that 'object a' probably should be assigned to cluster 1 but that there is still a glimpse of doubt in favor of clusters 2 and 3. In fuzzy analysis, these different situations are described by means of membership coefficients such as

Object 1	Cluster 1	Cluster 2	Cluster 3
a	0.90	0.05	0.05
b	0.05	0.90	0.05
d	0.10	0.10	0.80
d	0.10	0.45	0.45
e	0.33	0.33	0.34

They reflect that a belongs mostly to cluster 1, that b belongs mostly to cluster 2, and that c belongs mostly to cluster 3. Object d is interesting because it is about halfway in between clusters 2 and 3, yielding a 45% membership in both. At the same time its membership coefficient in cluster 1 is only 10%, because it lies much further away from that cluster. The ability to describe such ambiguous situations is an important advantage of the fuzzy approach; a "hard" clustering algorithm would assign object e to one of the clusters, leading to a distorted result.

For more information, see chapters 2 and 4 in Kaufman, Leonard and Peter J. Rousseeuw.

Hierarchical Methods

These methods accept dissimilarity matrix input but are not suitable for datasets of more cases than the maximum matrix dimension specified in Preferences and must be **sampled**, but can accept dissimilarity matrices. Not suitable for more than 100 objects, as the printout rapidly grows and becomes unmanageable.

Agglomerative nesting

Starts with all objects apart and chooses the smallest dissimilarity to form a new cluster. Thereafter, uses the group average method average of all dissimilarities in each group to form a new dissimilarity matrix and continues.

Divisive

Starts with all objects in one group and chooses the largest average dissimilarity of one object from the rest to form the first splinter group. Thereafter chooses the average dissimilarity of the largest group and compares it to the average dissimilarity with the objects of the splinter group.

Single Linkage

Agglomerative, using the minimum dissimilarity of all pairwise distances. Subject to chaining effect where two distinct clusters are chained together by a single link leading to drawn out linear shaped clusters.

Complete Linkage

Agglomerative, using the maximum dissimilarity of all pairwise distances. May lead to many clusters with small within-cluster dissimilarities.

Centroid distance

Intended for squared Euclidean distances only. Dissimilarity between two clusters is defined as the Euclidean distance between their centroids.

Ward's method

Similar to Centroid distance, but merges clusters with minimal dissimilarities.

Monothetic (binary values of 0/1)

For dataset input only.

Selects one variable and divides the set of objects into two groups with and without that attribute. Continues the process with each subgroup. The separation variable is chosen for which the sum of similarities to all other variables is as large as possible.

INPUT DATA

For RANKS, KMEANS and MONOTHETIC, a MicrOsiris dataset of measurement data.

For all other options, input should be a dataset of measurement data or a symmetrical matrix whose elements are measures of dissimilarities. Use the **MATRANS** command to calculate dissimilarities from correlations and other similarities matrices.

For monothetic hierarchical splitting, the input is a MicrOsiris dataset whose values must be binary, i.e., 0's and 1's.

Sampling

For all but RANKS, KMEANS, and MONOTHETIC, if input is a dataset of measurement data CLUSTER creates a dissimilarity matrix before analysis. Sampling is used to restrict the size of the dissimilarity matrix to a manageable size. If the whole dataset has fewer than the maximum matrix dimension specified in Preferences, no sampling is done. The number of records sampled is determined by the value of the maximum matrix dimension set in Preferences from the MicrOsiris command screen.

Missing Data.

For dataset input, cases with missing-data in any variable are deleted automatically. Consider using **IVEWARE** or **USTATS** to impute for missing data before running SEARCH. Use USTATS to standardize the data first if desired.

Use **MATRANS** to prepare dissimilarity matrices.

Occasionally variables are measured by arbitrary conventions, and the magnitude of the measure of their similarity is more important than its sign. Negative values are sometimes a result of scoring conventions rather than an indication of an intrinsically negative relationship between variables. Use the MATRANS to remove the effect of these arbitrary measurement conventions by reducing the number of negative signs in the input matrix.

For measures of distances which are to be used as squared Euclidean distances, use the SQUARE option of MATRANS to square each element.

Use **CORRELATIONS** to produce correlations and covariances from interval-scaled data and MATRANS to convert them to dissimilarities.

For nominal and ordinal data, use TABLES to produce Spearman Rho, Gamma, Tau-c , Lambda, or Cramer's V correlation matrices and MATRANS to convert them to dissimilarities.

OUTPUT DATA

An output dataset with cluster numbers merged with the input dataset is optional.

SPECIAL TERMINOLOGY

Clustering Vector. A vector of numbers to which cluster each object belongs.

Cluster Alpha The coefficient alpha of the newly formed cluster.

Cluster Index. Criterion to determine the pair of clusters to combine into a new clustering.

Cluster R_{BAR}. The imputed average correlation between clusters. Less sensitive than the coefficient alpha to the number of clusters.

Cluster F. Ratio of the between-cluster variances to the within-cluster variances adjusted by their degrees of freedom.

Dunn's partition coefficient. A measure that takes the value 1 for a "hard" partition.

Isolated clusters. Denoted by L and L^* . (See Gordon, 1981).

C is an L cluster if for every object i in C $\max d(i,j) < \min d(i,h)$ for all j in C and all h not in C .

C is an L^* cluster if for every i,j in C $\max d(i,j) < \min d(l,h)$ for all l in C and h not in C

Libert's coefficient. A measure of non-fuzziness.

Medoid. A representative object for a cluster of objects (Kaufman and Rousseeuw, 1987).

Silhouette Coefficient. A measure of the strength of the structure found.

PRINTED OUTPUT

Means, Standard Deviations, Minimum/Maximum for each cluster for dataset input.

KMEANS, RANKS

Number of cases in each cluster and percent of cases in each cluster.

PAM, CENTROID, WARD, SINGLE/COMPLETE Linkage analysis results.

Number of representative objects and the final average distance.

For each cluster: representative object ID, number of objects and the list of objects belonging to this cluster.

Coordinates of medoids: values of analysis variables for each representative object (input dataset only).

Clustering vector: vector of numbers indicating to which cluster each object belongs.

Silhouette plot

FUZZY analysis results. For each number of clusters in turn (going from CMIN to CMAX):

Number of clusters, objective function value at each iteration.

For each object, its ID and the membership coefficient for each cluster, partition coefficient of Dunn and its normalized version,

Closest hard clustering: number of objects and the list of objects belonging to each cluster.

Clustering vector.

Silhouette plot.

AGGLOMERATIVE nesting analysis results contain the following:

Final ordering of objects (identified by their ID) and dissimilarities between them.

Banner plot.

Agglomerative coefficient AC (average strength of the clustering structure i.e., fraction of "blackness" in the banner plot).

DIVISIVE analysis results contain the following:

Final ordering of objects (identified by their ID) and diameters of the clusters.

Banner plot.

Divisive coefficient DC (average width of the divisive banner).

MONOTHETIC analysis results contain the following:

For each step, the cluster to be separated, the list of objects (identified by their ID variable values) in each of the two subsets, and the variable used for the separation.
The final ordering of objects.
The variable used for the separation.

OPTIONS

Choose CLUSTER from the command screen and make selections.

For a Runfile use:	CLUSTER Filter statement (optional) Job Title Keyword choices from below
--------------------	-----------------------------------------------------------------------------------

MATRIX=n n is the number of the matrix produced by a previous command or read in by the MATIN file assignment.
Default: Create dissimilarity matrix from MicroSiris dataset (measurement or binary data).

When input is a MicroSiris measurement dataset:

RECODE=n Use RECODE n, previously entered via the RECODE command.

DISTANCES=EUCLIDEAN|CITYBLOCK
Default: EUCLIDEAN. For KMEANS and RANKS, distances are always Euclidean.

WRITE Write output dataset with cluster numbers added.

ID=variable number
ID variable for output dataset. The ID variable cannot be alphabetic.
Default: ID is required and when WRITE is specified may not be negative or have fractional values and must be ascending.

VAR=(variable numbers) Use the variables specified in the list.

PRINT=MATRIX
MATRIX Print the input matrix.

TYPE= KMEANS|RANKS|PAM|FUZZY|AGGLOM|DIVISIVE |SINGLE|COMP|CENTROID|MONO
KMEANS K-means (dataset input only).
RANKS Rank order method (dataset input only).
PAM Partition around medoids.
FUZZY Partition around medoids with fuzzy clustering.
AGGLOM Agglomerative nesting hierarchical clustering.
DIVISIVE Divisive hierarchical clustering.
SING Single linkage method.
COMP Complete linkage method.
WARD Ward's method
CENTROID Centroid distance method (squared Euclidean distances).
MONO Monothetic clustering of binary objects (more than 3 variables, data input only).

MINC=n For PAM and FUZZY: the minimum number of clusters to try. For KMEANS and RANKS, the exact number of clusters to form.
Default: MINC=2. Maximum value is 10.

MAXC=n For PAM and FUZZY, the maximum number of clusters to try. $N \geq 1 \leq 10$.
Default: MAXC= MINC. Maximum value is 10.

MIN% Minimum percentage of cases per group when TYPE=RANKS.
Default: MIN%=10.

KP=(RANDOM|FIRST|LAST
Initial points for TYPE=KMEANS
RANDOM: K randomly selected points.
FIRST: First k points in the dataset.
LAST: Last k points in the dataset.
Default: KP=RANDOM.

REFERENCES

Kaufman, Leonard and Peter J. Rousseeuw. *Finding Groups in Data*. Wiley, New York, 1990.

EXAMPLES

Example 1: Group 1 cluster option: Partition around medoids, 10 objects and 2 variables (see p.73, Kaufman & Rousseeuw).

```

*** CLUSTER ANALYSIS ***

Partition around k-medoids

Dataset CLUSTER\MATRIX3

MIN clusters = 2   MAX clusters = 2

Distances are Euclidean

Dissimilarity matrix

      Object Object Object Object Object Object Object Object Object Object
      1      2      3      4      5      6      7      8      9      10
Object 1   0.00
Object 2   5.00   0.00
Object 3   4.47   1.00   0.00
Object 4   4.00   3.00   2.00   0.00
Object 5   9.00   5.83   5.39   5.00   0.00
Object 6  24.00  20.22  20.10  20.00  15.00   0.00
Object 7  24.08  20.62  20.40  20.10  15.13   2.00   0.00
Object 8  24.19  20.88  20.62  20.22  15.30   3.00   1.00   0.00
Object 9  24.33  21.19  20.88  20.40  15.52   4.00   2.00   1.00   0.00
Object 10 28.16  24.74  24.52  24.19  19.24   5.00   4.12   4.00   4.12   0.00

Number of representative objects: 2

```

Final average distance = 2.186

Cluster	Medoid	Size	Objects
---------	--------	------	---------

1	Object 3	5	Object 1 Object 2 Object 3 Object 4 Object 5
2	Object 8	5	Object 6 Object 7 Object 8 Object 9 Object 10

Coordinates of Medoids

Object 3	5.00	2.00
Object 8	25.00	7.00

Clustering vector

1	1	1	1	1	2	2	2	2	2
---	---	---	---	---	---	---	---	---	---

CLUSTERING CHARACTERISTICS

Cluster 1 an isolated L*-Cluster with diameter 9.00 and separation 15.00

Cluster 2 an isolated L*-Cluster with diameter 5.00 and separation 15.00

Number of isolated clusters: 2

Diameter of each cluster

9.00	5.00
------	------

Separation of each cluster

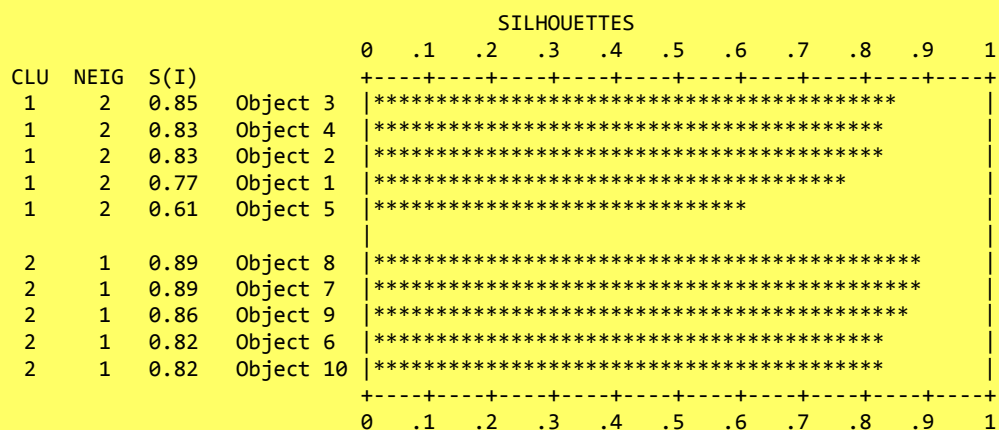
15.00	15.00
-------	-------

Average distance to each medoid

2.57	1.80
------	------

Maximum distance to each medoid

5.39	4.00
------	------



Cluster 1 has average silhouette width .78

Cluster 2 has average silhouette width .86

For the entire data set the average silhouette width is .82
indicating a strong structure was found.



COMPARE CONFIGURATIONS

File Assignments:	MATIN	Input matrix
	TARGET	Target matrix
	EMATRIX	Error residuals matrix
	RMATRIX	Rotated configuration matrix

GENERAL DESCRIPTION

COMPARE is based on Schonemann and Carroll's procedure for "fitting one matrix to another under choice of a central dilation and rigid motion." The technique rotates one configuration (the problem space) to the space of the other configuration (the target space) to achieve a least-squares fit. In seeking the best fit, the rotation is a "rigid motion," which maintains the orthogonality of the axes.

A typical application is to compare the configurations produced by non-metric scaling analysis and factor analysis from the same input data.

COMMAND FEATURES

Point Correspondence: A one-to-one correspondence is set up between rows in the problem configuration and rows in the target configuration. Unmatched rows are allowed, but are ignored in the process of determining centers, rotations, and contraction/expansions. The resulting transformations are performed on the entire problem configuration so that unmatched points are "carried along" into their correct relative positions in the target space.

Centering: A configuration may be centered by computing a displacement vector of column means and subtracting this from each row of the configuration. If matrix B has been centered, then the B(hat) that is printed will have been "de-centered" by re-adding the column means from B. Thus, $B(\text{hat}) = B - E$ is always true, but here B is the original B before centering.

SPECIAL TERMINOLOGY

Configuration: Conceptually, a collection of "points" in a multidimensional coordinate "space" described by a rectangular matrix, in which each row is a vector representing one point in the configuration. Thus, each column represents one dimension of the coordinate space, and the matrix element in row i, column j is the j-th coordinate for point i.

The Underlying Mathematical Model: The equation for the general linear model used is:

$$B = cA^T + JG' + E$$

where A = the problem configuration; its orientation will be altered to achieve a least-squares fit with the target configuration.

B =the target configuration. Related to this matrix is the matrix of best fit, B(hat), which is the rotated configuration derived from A:

$$B(\text{hat}) = cAT + JG' = B - E$$

T =the transformation matrix used to rotate matrix A to produce the B(hat) matrix.
COMPARE requires that $T'T = I$ so the orthogonality of the axes is maintained.

c = a scalar factor used to contract or expand the rotated configuration.

G' = the translation vector (gamma) used to align the centers of B and B(hat).

J = a column vector of 1's used to replicate G' for each row of A.

E =the error (residuals) matrix: difference between rotated and target configurations.

PRINTED OUTPUT

Input Matrices: The problem and target configurations are printed after reflecting any rows and before centering.

Result matrices: The transformation matrix T is printed, and T given in degrees is rotated (i.e., angles between axes in the problem space vs. axes in the target space).

The Contraction Factor c.

The Translation Vector G'.

The B(hat) Matrix.

The Error Residuals Matrix E.

Configuration Plots: These are projections of B and B(hat) onto successive pairings of the dimensions in the target space. Each plot occupies one page (see option NPLOT).

OPTIONS

Choose COMPARE from the command screen and make selections.

CONTRACT Allow the rotated problem configuration to contract or expand.

EMATRIX=n A matrix number to assign to the error residuals matrix, E.
Default: None; if the option is not given, the matrix is printed but is not output for further analysis.

NPLOT=n The number of dimensions (column variables) to be plotted in every combination of two at a time ($n(n-1)/2$ plots are produced). If n=0, no plots are produced; if n=1, all dimensions are plotted.
Default: NPLOT=1

PROBLEM=(MATRIX=n,NAME='name',ROWVARS=(variable list),COLVARS=(variable list),REFLECT=(variable list)|ALL, ORIGIN=CENTER| ZERO)

Describes the problem configuration, A:

MATRIX The input matrix number.
 NAME A 1- to 24-character name to label this matrix.
 Default: Use the TITLE stored with the matrix file.
 ROWVARS The list of row variable numbers to include for this matrix.
 COLVARS The list of column variable numbers to include for this matrix. If
 one input matrix uses fewer columns than the other, columns of
 zeros are added so the number of columns match.
 REFLECT List of row variables to reflect. A row is reflected by inverting the
 sign of each element in the row.
 ORIGIN=CENTER|ZERO
 CENTER Shift the origin for this configuration to the center of
 the points. Center after reflection.
 ZERO Assume the origin is the center of the configuration.
 Default: ORIGIN=CENTER. All rotations, contractions, and
 translations are relative to the chosen ORIGIN.

RMATRIX=n A matrix number to assign to the rotated configuration, B.
 If not given, the matrix is printed but not saved.

TARGET=(MATRIX=n,NAME='name',ROWVARS=(variable list), COLVARS= (variable
 list),REFLECT=(variable list)|ALL, ORIGIN= CENTER|ZERO)
 Describes the target configuration, B. The meanings of the associated options
 are the same as for PROBLEM above.

WRITE Write the output matrices to a file.

REFERENCES

Schonemann, Peter H. and Robert M. Carroll. "Fitting One Matrix to Another under Choice of a Central Dilation and a Rigid Motion." *Psychometrika*, Vol. 35, 1973, pp. 245-255.

EXAMPLE

```

*** COMPARE - CONFIGURATION COMPARISON ***

Problem matrix is matrix number 1 MATRIX A

Target matrix is matrix number 2 MATRIX B

4 ROWS CORRESPOND BETWEEN INPUT MATRICES

Problem configuration:

MATRIX A
          V101      V102
    DIMENSION X DIMENSION Y
POINT 1  V1      -4.000      .000
POINT 2  V2      -3.000     -4.414
POINT 3  V3       .000     -1.414
POINT 4  V4       .000      .000

```

Configuration will be centered

TARGET CONFIGURATION:

MATRIX B

		V201	V202
		DIMENSION V	DIMENSION W
POINT A	V1	-4.456	-1.688
POINT B	V2	-.165	-1.646
POINT C	V3	-3.230	-2.045
POINT D	V4	-1.114	-4.235

Configuration will be centered

TRANSFORMATION MATRIX T

		V201	V202
		DIMENSION V	DIMENSION W
DIMENSION X	V101	.449	-.894
DIMENSION Y	V102	-.894	-.449

T AS DEGREES ROTATED

		V201	V202
		DIMENSION V	DIMENSION W
DIMENSION X	V101	63.327	153.327
DIMENSION Y	V102	153.327	116.673

Contraction factor C = 1.00

TRANSLATION VECTOR GAMMA

	V101	V102
	DIMENSION V	DIMENSION W
	-2.758	-4.621

BEST FIT MATRIX B(HAT)

		V201	V202
		DIMENSION V	DIMENSION W
POINT 1	V1	-4.553	-1.047
POINT 2	V2	-.160	.041
POINT 3	V3	-1.494	-3.987
POINT 4	V4	-2.758	-4.621

RESIDUAL MATRIX E

		V201	V202
		DIMENSION V	DIMENSION W
POINT 1	V1	.097	-.641
POINT 2	V2	-.005	-1.687
POINT 3	V3	-1.736	1.942
POINT 4	V4	1.644	.386

GOODNESS OF FIT MEASURES

TR(E'E)/PQ = 1.6125
Normalized symmetric error = 2.0431

Root mean square distance = 1.7958

CONFIGURATION PLOTS

PLOT IDENTIFICATION TABLE:

```

-----
--- ROTATED CONFIGURATION ---

```

```

POINT ROWVAR
00      V1  POINT 1
01      V2  POINT 2
02      V3  POINT 3
03      V4  POINT 4

```

```

--- TARGET CONFIGURATION ---

```

```

POINT ROWVAR
A0      V1  POINT A
A1      V2  POINT B
A2      V3  POINT C
A3      V4  POINT D

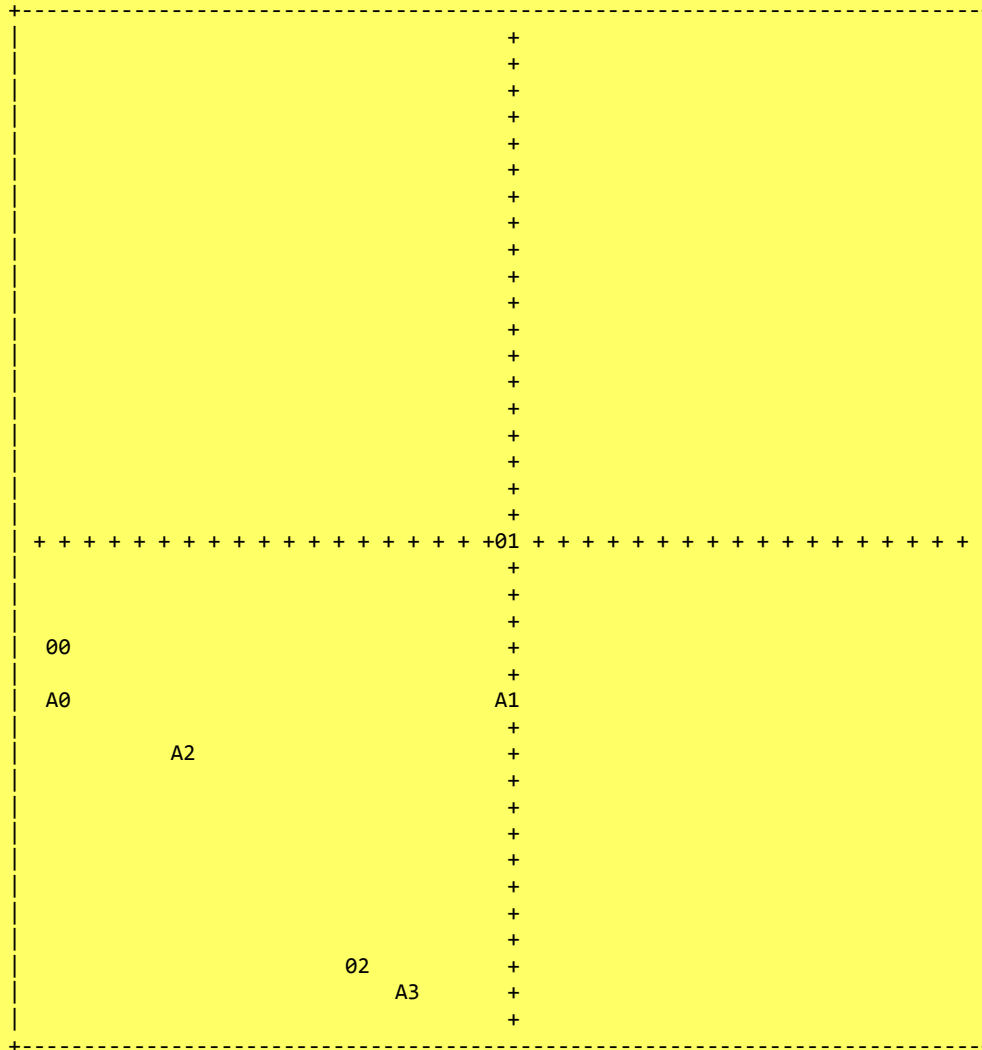
```

```

--- ALL PLOT SCALES RANGE FROM -4.62      TO      4.62

```

V2 DIMENSION W (down) vs. V1 DIMENSION V (across)



CONSISTENCY CHECK

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

CONSISTENCY CHECK, used in conjunction with RECODE, performs consistency checks by testing for illegal relationships between values for groups of variables. CONSISTENCY CHECK takes specifications you provide, indicating data inconsistencies from tests made in RECODE, and displays information allowing you to locate each inconsistency. You can then use RECODE and TRANSFORM or FIX_DATASET to correct the inconsistencies.

CONSISTENCY CHECK can print different sets of variables for each consistency test and labels each inconsistency with a name and number.

COMMAND FEATURES

A consistency test is defined through RECODE by testing for a logical relationship between the values of a set of variables. If the values for a case conform to the conditions of the test, the variables are consistent for that test. If the values do not conform, the variables are inconsistent. You designate an inconsistency by assigning the value 1 to a variable in RECODE and assigning the value 0 otherwise. (e.g., R1=1 if inconsistent, R1=0 otherwise). Generally one R-type variable, specified by a Condition Statement, is used for each consistency test.

Because different rules may test different logical relationships between common variables, it is possible for a single variable to be consistent for one test but inconsistent for another. Before correcting variables it may be wise to examine a large number of variables to ensure the correction will not cause other inconsistencies.

PRINTED OUTPUT

For each case containing an inconsistency, one identification line is printed. If the VARS option is used, the variables listed and their values for that case are also printed. For each inconsistency the following are printed:

- The number of the inconsistency
- The name of the inconsistency
- The list of variables appearing in the Condition Statement and their values

After processing the whole dataset, a summary table is printed which contains:

- The number of cases processed
- The number of cases containing at least one inconsistency
- The number of inconsistencies processed
- Each inconsistency number, name, and number of cases with the inconsistency

RESTRICTIONS

If you use a recoded or recode result variable in the ID option, it cannot appear anywhere else in the setup.

OPTIONS

Choose CONSISTENCY CHECK from the command screen and make selections.

ID=variable list

Up to 10 variables may be specified for use as identification for each case containing illegal codes. If not specified, sequential case numbers will be used. ID variables may be alphabetic.

MAXI=n

The maximum number of inconsistencies allowed before stopping.
Default: MAXERR=99.

RECODE=n

Use RECODE n, previously entered via the RECODE command.

VAR=variable numbers

List of variables to print for each inconsistency found. Variable numbers, names, and values are printed.

Condition Statements

Supply one condition statement for each consistency you want tested.

TEST=Rn|Vn Variable for which a value of 1 indicates for a consistency test that an inconsistency has occurred.

VAR=variable numbers

List of variables to print when this inconsistency is found.
Default: None; required for first statement.
Subsequent default: Variable list of previous statement.

NAME='string' Name of inconsistency, up to 40 characters.

EXAMPLE

Run with two rules testing the relationship between V1 and V3 and the relationship between V1 and V2. The name and value of variable V1 will be printed with the case sequence number for each inconsistency detected. The name and value of variable V3 is printed for each inconsistency found by the first test, and the name and value of V2 is printed for each inconsistency found by the second test.

First RECODE is used to create the test conditions, using the RECODE statements:

```
IF V1 LT 0 AND V3 EQ 4 THEN R1=1 ELSE R1=0  
IF V1 EQ 0 AND V2 LT 1 THEN R2=1 ELSE R2=0
```

then CONSISTENCY CHECK is invoked, using recode 1 and using descriptive names for the tests and specifying what variables to display for each consistency found:

```
TEST=R1 NAME='First inconsistency: v1 < 0 and v3 = 4' vars=v3
TEST=R2 NAME='Second inconsistency: v1 = 0 and v2 < 1' vars=v2
```

```

                                CHECKING FOR TWO INCONSISTENCIES

Dataset SAMPLE

Transforming the data by RECODE number 1

ID = 2
V1 Better or Worse=-4

TEST: R1 First inconsistency: V1 < 0 and v3 = 4

VARIABLES: V3 Children=4

-----

ID = 4
V1 Better or Worse=0

TEST: R2 Second inconsistency: V1 = 0 and v2 < 1

VARIABLES: V2 Income (000)=.9

-----

          5 cases processed
          2 inconsistencies
          2 cases have at least one inconsistency

TEST      NAME                                INCONSISTENCIES
-----
1 First inconsistency: V1 < 0 and v3 = 4          1
2 Second inconsistency: V1 = 0 and v2 < 1        1
```

CONJOINT ANALYSIS

MONOTONE ANALYSIS OF FACTORIAL DESIGNS

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

CONJOINT is an additive (main effects) model for performing conjoint analysis with full factorial designs.

Monotone analysis of variance is a procedure for transforming data from a factorial experiment. It searches over all monotone (ascending) transformations of the data, and picks the 'best' one: this means the monotone transformation which results in the greatest percentage of variance being accounted for by the main effects.

The analysis finds utility (part worth) values for each level of the independent variables such that the ranking of main-effects combinations preserves the original ranking of the design as best as possible.

The theory and computational procedure were published by Kruskal (1964) and further analyzed by H. Noguchi and H. Ishii, Faculty of Engineering, Osaka University, Japan.

COMMAND FEATURES

If the data values contain ties, two different kinds of least squares monotone regression are possible. In the PRIMARY approach (REGR=PRIMARY) no restriction is placed by the regression on values which correspond to tied data values; the fitted values need not be equal. The SECONDARY approach (REGR=SECONDARY) requires that if two data values are tied, then their corresponding fitted regression values must be equal.

There does not seem any basis in general for preferring one approach over the other. If there are few ties, it makes little practical difference which approach is used.

PRINTED OUTPUT

Final configuration:

Stress

Iterations

Scatter plot displaying two functions on one plot.

Function one shows the best monotonic transformation of the data (vertical axis) versus the original data values (horizontal axis).

Function two shows the fitted value for that cell based on main effects.
If the two functions coincide, only the Xs appear.

INPUT DATA

The data should be a complete factorial. CONJOINT will sort the data by factor if necessary. If you want a permanent copy of the data in that order, use [SORT DATASET](#) first.

OPTIONS

Choose CONJOINT from the command screen and make selections.

For a Runfile use:	CONJOINT
	Filter statement (optional)
	Job Title
	Keyword choices from below

DEPV=Vn Dependent variable

VAR=(Vnum1:n_1, Vnum2:n2_1,etc.)

Factor variable numbers and levels. Values outside the ranges 1:n_i are discarded. Up to 12 factors may be specified.

CONV=f Stress convergence criterion. It should always be less than or close 1. For greater stringency, a value close to 1, such as 1.001, should be used. For less stringency, a value such as 0.99 should be used.
Default: 1.001

MIN=f Minimum stress level acceptable. Must be less than 1 (100%).
Default: .01

REGR=STANDARD|PRIMARY|SECONDARY

STANDARD: Standard regression.

PRIMARY: Block regression, primary approach.

SECONDARY: Block regression, secondary approach.

Default: PRIMARY

MAXCYCLES=n

Maximum number of iterations.

Default: 50

MAXRECORDS=n

Maximum number of records allowed. Set this to the actual number of expected records to conserve memory if necessary.

DEFAULT: 1,000,000.

RECODE=n Use RECODE n, previously entered via the RECODE command.

REFERENCES

"Analysis of Factorial Experiments by Estimating Monotone Transformations of the Data" (1965) by Joseph B. Kruskal, in Journal of the Royal Statistical Society, Series B, 27(2), pp 251-263.

EXAMPLE

Three factors, 2 two levels each VARS=(V2:2,V3:2,V4:2). This example matches the one in Kruskal (1965),

```
*** CONJOINT ANALYSIS FOR FACTORIAL DESIGNS ***  
  
Example from Kruskal (1965)  
  
Dataset conjoint  
Dependent variable: V1  
Factor variables: V2 V3 V4  
Max cycles: 50  
Stress minimum desired: .01  
Stress test value: .999  
Using block regression with secondary approach  
  
8 cases accepted  
  
ITERATION  STRESS  
      1      0.000000  
  
Zero stress reached  
  
FINAL CONFIGURATION HAS STRESS OF .0 PERCENT.  
      FACTOR              LEVELS  COEFFICIENTS  
V2  Variable 2              2      0.266 -0.266  
V3  Variable 3              2      1.498 -1.498  
V4  Variable 4              2      0.827 -0.827
```


CORRELATIONS

File Assignments:	DATASET	Input data
	MATOUT	Correlation matrix (optional)
	COVAR	Covariance matrix (optional)

GENERAL DESCRIPTION

Computes Pearson product-moment correlation coefficients for all pairs of variables in a list.

For Spearman Rho rank order correlations and other correlations for nominal and ordinal data see [TABLES](#).

COMMAND FEATURES

A CORRELATIONS matrix can be generated using either a case-wise or a "pair-wise deletion" for missing data. CORRELATIONS will also produce a covariance matrix.

Part-Whole Correlations. Optionally computes part-whole correlations, using a second pass of the data and case-wise missing-data deletion.

Partial Correlation Coefficients

Missing Data: Pair-Wise Deletion: The optional paired statistics (see the "Printed Output" section) and each correlation coefficient can be computed from the set of cases which have valid data for both variables (see option DELETE= PAIRS). Thus, a case may be used in the computations for some pairs of variables and not used for other pairs. If there are missing data, correlations may be computed on different subsets of the data. Large amounts of missing data can lead to internal inconsistencies in the correlation matrix, causing difficulties in subsequent multivariate analysis. Consider using [IVEWARE](#) to impute for missing-data and use case-wise deletion instead.

Missing Data: Case-Wise Deletion: Each correlation coefficient can be computed from the set of cases which have valid data on all variables in the local variable list (see option DELETE=CASES). Thus, a case is either used in computation for all pairs of variables or not used at all. This method of handling missing data is called "case-wise deletion."

You can use the [IVEWARE](#) command to first impute missing data if desired.

PRINTED OUTPUT

Covariance Matrix: (Optional: see option PRINT=COV.) A matrix of covariances is printed with variable names and numbers labeling the rows and columns.

Matrix of Paired n's: (Optional: see option PRINT=N.) A matrix of paired n's is printed with variable names and numbers labeling the rows and columns.

Correlation Matrix: The correlations are printed as a matrix with variable names and numbers labeling the rows and columns.

Univariate Statistics: The following are printed for each variable:

Number of "valid cases"
Sum of weights (if weighted)
Mean
Standard deviation (square root of unbiased estimate of population variance)

Paired Statistics: For each pair of variables the following are printed:

Correlation coefficient
Number of valid cases
Sum of weights (if weighted)
Mean of the x variable
Standard deviation of the x variable (square root of unbiased estimate of population variance)
Mean of the y variable
Standard deviation of the y variable (square root of unbiased estimate of population variance)
T-statistic significance measure
Estimated probability of T occurring by chance

Part-Whole Correlations. Optionally computes par-whole correlations. (See PRINT=PCOR). Requires a second pass of the data.

Regression Coefficients: (Optional: see option PRINT= REGR.) For each pair of variables x and y in the variable list, regression coefficients a and c and constant terms b and d in the regression equations $x=ay+b$ and $y=cx+d$ and the standard error of the estimate are printed.

Summary Table of Significant Correlations: (Optional: see option LEVELS.) The T statistic, correlation coefficient, and estimated exact probability of that value of T occurring by chance is printed for each pair of variables.

Standardized Partial Regression Coefficients: The standardized partial regression coefficients (betas) are printed when the partials option is selected. For each coefficient, the row variable is the dependent variable and the column variable is the independent variable.

Partial Correlation Coefficients (optional): The partial correlation coefficients for each pair of variables in the correlation matrix are printed.

Multiple Correlation Coefficients: The multiple correlation coefficient R for each variable, using all the others as predictors, is printed on the diagonal of the partial correlation matrix.

Output Correlation and Covariance Matrices

Creation: The output correlation matrix is produced only if the option WRITE is specified and the variables are not pairs. The output covariance matrix is produced only if option COV=WRITE is specified under the same conditions.

Content: The matrices output from CORRELATIONS contain correlations, means and standard deviations; and covariances, means and standard deviations. Means and standard deviations are unpaired. The matrices contain variable numbers and names obtained from the input dataset. The order of the variables is determined by the order of variables in the list. No diagonal is output for correlation matrices; for covariance matrices, the diagonal contains the variances.

Output Missing Data: CORRELATIONS may generate pseudo correlations of 99.999 and covariances of 1.5 billion, indicating it is impossible to compute a meaningful value. Typically, this means all cases were eliminated due to missing data or that one of the variables was constant valued.

INPUT DATA

Interval scale: A Pearson correlation coefficient assumes that the variables were measured on an interval scale; if a Pearson r is used only to describe a given set of data, with no inference involved, this assumption is not necessary.

Inconsistent triplets. Peculiar distributions of missing data in a dataset can lead to internal inconsistencies in the correlation matrix when pair-wise deletion of missing data is used. If inconsistencies are detected, partial correlations might lead to a singular matrix and partial correlations won't be computed. (See Walker and Lev, pp. 349-345, for a discussion of inconsistent triplets.)

OPTIONS

Choose CORRELATIONS from the command screen and make selections.

For a Runfile use:	CORRELATIONS Filter statement (optional) Job Title Keyword choices from below
--------------------	----------------------------------------------------------------------------------------

CMATRIX=n The number to assign to the covariance matrix.
Default: MATRIX=2 when WRITE is also specified.

CWRITE Write the covariance matrix to the file assigned to MATOUT.

DELETE=PAIRS|CASES
 PAIRS Pair-wise deletion.
 CASES Case-wise deletion.
Default: DELETE=PAIRS.

LEVEL=p Calculate and print a summary table of correlations whose probability of occurrence by chance is less than or equal to p, where $0 < p < 1$.

MATRIX=n Assigns the number n to the matrix so that it may be referenced by a later command. The number n must be a positive integer less than 1000.
Default: 1

PRINT=(N,PAIR,REGR,COV,PW,PARTIALS)

N Print the matrix of paired N's.
 PAIR Print the paired statistics.
 REGR Print the regression coefficients.
 COV Print the covariance matrix.
 PW: Calculate and print the part-whole correlations.
 PARTIALS: Calculate and print the partial correlations.

RECODE=n Use RECODE n, previously entered via the RECODE command.

VARS=(variable numbers) |ALL
 Use the variables specified in the list. If you use VARS=ALL, the matrix will include the weight variable if present.

WT=n Use variable n as a weight variable.

WRITE Write the matrix of correlations, means, and standard deviations to the file assigned to MATOUT.

REFERENCES

Rummel, R. J. *Applied Factor Analysis*. Evanston, IL: Northwestern University Press, 1970. (The algorithm for pair-wise deletion of missing data is described on pp. 258-59.)

EXAMPLE

Conventional symmetrical matrix with pair-wise deletion with all correlations significant at the 5 percent level summarized, computing partial and part-whole correlations.

```

*** PEARSONIAN CORRELATION ANALYSIS ***

          SAMPLE RUN

Dataset SAMPLE

Weighting the data by variable V4

Assigning number 1 to the correlation matrix

Missing-data option: PAIR-WISE deletion

5 cases accepted

Total weight sum: 5.00000

      Name                Variable      N      Mean      StdDev
Income (000)              V2          4      10.725     17.538
Children                  V3          5       2.200       2.168
Assets                    V5          5       2.558       1.266

CORRELATIONS

          V2      V3
Income
(000) Children

```

Children	V3	0.724	
Assets	V5	0.912	0.886

PART-WHOLE CORRELATIONS

X	Y	N	Mean X	StdDev X	t-level	r	p
V2	WHOLE	4	10.725	17.538	1.848	0.7942	0.14
V3	WHOLE	4	2.500	2.380	1.559	0.7406	0.19
V5	WHOLE	4	2.920	1.124	4.012	0.9431	0.02

STANDARDIZED PARTIAL REGRESSION COEFFICIENTS (BETAS)

		V2	V3	V5
	Income (000)	Children	Assets	
Income (000)	V2	0.0000	-0.3938	1.2608
Children	V3	-0.4999	0.0000	1.3422
Assets	V5	0.5673	0.4758	0.0000

PARTIAL CORRELATIONS (Multiple R-squares on diagonal)

		V2	V3	V5
	Income (000)	Children	Assets	
Income (000)	V2	0.9298		
Children	V3	-0.4437	0.9099	
Assets	V5	0.8458	0.7991	0.9690

DESCRIBE

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

DESCRIBE estimates the population means, proportions, subgroup differences, contrasts and linear combinations of means and proportions. A Taylor Series approach is used to obtain variance estimates appropriate for a user-specified complex sample design.

DESCRIBE invokes the SrcWare version of IVEware installed with MicroSiris to perform the analysis.

COMMAND FEATURES

A simple random sample analysis is performed if STRATUM, CLUSTER and WT variables are not specified. Use [USTATS](#) for general univariate statistics for simple random samples.

If a design based analysis involves only a WT variable and no STRATUM or CLUSTER variable, then a pseudo stratification variable and a pseudo cluster variable should be used. When using pseudo variables, all observations in the data set should have the same value for the pseudo STRATUM variable (e.g., 1), while each observation should have a unique value on the pseudo-CLUSTER variable (e.g., observation ID). The pseudo variables can be created with RECODE prior to performing the analysis.

DESCRIBE creates name.set and name.data, where name is the input dataset name, which are submitted to SrcWare. You can save these (SAVE option) for later modification and refinement and use them directly with Srcware. See [Srcware User Guide](#) for details.

Alphabetic variables may not be used.

See the [IVEware User Guide](#) for more information.

PRINTED OUTPUT

For each factor of each variable:

- Covariance of denominator
- Standard error
- Mean
- Confidence interval and T test
- Bias
- Design Effect

RESTRICTIONS

Alphabetic variables may not be used.

REFERENCES

[IVEware](#) was developed by the Survey Methodology Program at The University of Michigan's Survey Research Center, Institute for Social Research.

OPTIONS

Choose IVEWARE from the command screen and make selections.

(The descriptions for most of these keywords were adapted from the [IVEware User Guide](#).)

SAVE Save the IVEWARE setup and data file for later modification and use with the SRCLIB version of IVEWARE.

RECODE=n Use RECODE n, previously entered via the RECODE command.

STRATUM=Vn Use variable n as the stratum variable.

CLUSTER=Vn Use variable n as the cluster variable.

WT=Vn Use variable n as a weight variable.

MEAN=variable numbers
Means, standard errors, and design effects are calculated for the variables.

TABLE=variable numbers
Produce the weighted proportions and their standard errors for all levels of the variable. Cross-tabulations may be indicated with an asterisk, for example:
TABLE=V1*V2.

BY=variable numbers
Used in conjunction with TABLE or MEAN. The analysis is performed for each level of the variable(s) specified in the BY statement.

CONTRAST=Vn[*Vm]
Used in conjunction with the MEAN keyword to compare or estimate linear combinations of cell means. For example, if V1 is Income and V2 is Race and V3 is Gender, then MEAN V1 CONTRAST=V2 produces all the pairwise comparisons of mean Income defined by Race. CONTRAST= V2*V3 produces comparisons of Income means for all combinations of Race and Gender.

PAIRS=list,DIFF=list
Variance estimation method to be used.
PAIR: Use the paired selection method.
DIFF: Use the successive differences method.
Default: Uses the multiples method, useful when there are multiple PSUs within a stratum. You can specify different methods for each stratum. For example, PAIR(15,16,17) DIFF(20,21,27); uses paired differences for strata 15, 16, 17, the successive differences for strata 20, 21,27, and multiples for the rest.

EXAMPLE

Variable 268, no contrasts.

Analysis description:

1 Variables
0 Strata
0 Secus

Strata Model
0 Multiple PSU
0 Paired Selection
0 Successive Differences

327 Cases Read

Problem 1

Degrees of freedom
326

Factor Covariance of denominator
None 0.00000

Mean	Number of	Sum of	Weighted	Standard
V268: Total family inc	Cases	Weights	Mean	Error
	327	327	10421.26	416.0166
	Lower	Upper	T Test	Prob > T
	Bound	Bound		
	9602.843	11239.68	25.05011	0.00000
	Unweighted	Bias	Design	
	Mean		Effect	
	10421.26	0.00000	1.00000	

DISCRIM -- MULTIVARIATE LINEAR DISCRIMINANT ANALYSIS

File Assignments:	DATASET SCORES	Input dataset RECODE statements to compute scores (optional)
-------------------	-------------------	-----------------------------------------------------------------

GENERAL DESCRIPTION

Performs multivariate linear discriminant analysis. Up to 25 factors (groups) can be used. The method of solution used by DISCRIM is based on matrix operations described in Cooley and Lohnes (1971). See also [FACTOR ANALYSIS](#) and [MANOVA](#). The discriminant model can be interpreted as a special case of factor analysis that extracts orthogonal factors of the measurements for displaying and capitalizing upon the differences among criterion groups. Discriminant analysis derives components which best separate the defined groups.

COMMAND FEATURES

Missing Data: Cases with missing-data codes on any of the input variables (dependent, covariate, or factor variables) are excluded. This may result in many excluded cases and constitutes a potential problem you should consider when planning an analysis. Use [IVEWARE](#) to impute missing data to avoid this problem.

Correlation: When significant correlation among the factor variables is present (See LEVEL=f option), a warning is issued.

Group Classification: MicrOsiris generates classification matrices using the discriminant equation DISCRIM derives. If there are three or more groups, DISCRIM chooses the best discriminant function found (the one with the highest chi-square value.) DISCRIM then uses the chosen discriminant equation to compute discriminant scores on all cases in the sample to classify cases in groups.

Two options are available: equal group membership and proportional group membership likelihoods:

Equal: DISCRIM locates the points in the score rank ordered distribution that divide the numbers of cases into n equal parts, where n is the number of groups. DISCRIM then sets temporary codes that identify n equal groups of cases. It also presents the proportion of the total sample that the chosen discriminant equation correctly classifies.

Proportional: DISCRIM rank orders the scores from highest to lowest. DISCRIM then locates the points in the score rank ordered distribution where the count of each group divides the sample into the actual number of matching each actual group size. DISCRIM then sets temporary codes that identify the proportional groups.

Finally, DISCRIM produces an n-by-n cross-tabulation matrix that presents both the count and the proportion of each group that the discriminant equation correctly classifies. It also presents the proportion of the total sample correctly classified.

SPECIAL TERMINOLOGY

Factor Variable: Factor variables are the independent variables; that are used to classify cases into groups. They are sometimes called treatment variables or control variables. DISCRIM uses a GROUP statement to define the factors (groups).

PRINTED OUTPUT

Group and pooled means, standard deviations and Ns.

T matrix: The matrix of sums of squares and cross-products of deviations of all subjects from the grand centroid.

Among-groups sums of squares and cross-products.

Within-groups sums of squares and cross-products.

Univariate F-Ratios, degrees of freedom, ETA squared, and probabilities.

Mahalanobis distance (only when there are two groups).

Wilks' Lambda, generalized correlation ratio and Eta squared.

F-Ratio for test of overall dispersion, degrees of freedom and probability.

Chi-square tests with successive roots removed. This refers to the significance of the remaining functions after removing effect of the preceding functions one by one. Hence "none" refers to nothing removed, 1 to removal of first function effect, 2 after 1 is removed, etc. Only displayed if more than one function exists. With multiple functions it tells the user which one is best.

Coefficient vectors for each discriminant factor. Absolute values of these indicate importance.

Factor structure. Correlations between variables and discriminant functions.

Communalities for each variable. If the communalities are less than 1, the variance for those variables with low communalities is not accounted for in the full set of discriminant functions, thus indicating they contain little information regarding group differences. Displayed only if less than 1.

Percentage of trace of r accounted for by each root.

Centroids.

Classification matrix. Matrix indication correctly classified cases. (see EQUAL|PROP option)

RESTRICTIONS

If a data value is missing for any variable in the analysis, DISCRIM deletes the whole case. Use IVEWARE to impute for missing-data if desired.

OPTIONS

Choose DISCRIM from the command screen and make selections.

For a Runfile use: DISCRIM
Filter statement (optional)
Job Title
Keyword choices from below

RECODE=n Use RECODE n, previously entered via the RECODE command.

VAR=variable numbers
Use the variables specified in the list.

WT=n Use variable n as a weight variable.

GROUPS=(Vn=list1[\$name1]/list2[\$name2]
Defines up to 25 groups of cases where each group is defined by Vn=listn.
The list of acceptable values may include single values and ranges of values
between -32,767 and 32,767 (use RECODE if needed). Cases falling in more
than one group are included only in the first such repetition.

Example: GROUPS= (V1=1\$Primary/2-5\$Other) defines and labels two
groups, one for V1=1 and the other for V1 in the range 2-5. Note that if
group names are absent, sequential numbers are used to label the groups.

STEP=FORWARD|BACKWARD Step-wise analysis. *Default:* All variables entered into the
mode (complete).

FIN=n The F-ratio below which a variable will be entered in the model. Default:
FIN=.001.

FOUT=n The F-ratio above which a variable will remain the model. Default:
FOUT=0.0.

OR

AIN=n The significance level below which a variable will enter the model. Default:
ALPHA=.025.

AOUT=n The significance level below which a variable will remain in the model.
This is the alpha level to remove. Default: ALPHA=.025.

PRINT=(T,SSCP,MEANS)
MEANS Print the means and standard deviations.
SSCP Print the cross-products matrix
T Print the T matrix.

LEVEL=f Check for correlations greater than f or less than -f.

EQUAL|PROP
EACH: Use equal group sizes for classifying groups.

PROP: Use actual group sizes for proportionally classifying groups.
Default: EQUAL.

SCORES=DATASET|RECODE

Create recode to compute scores variable 10000.

DATASET Write a dataset using the recode.

RECODE Create recode only for use in subsequent commands.

REFERENCES

Altman, et al. *Application of Classification Techniques in Business, Banking and Finance*.
Greenwich, CN, Aijai Press, 1981.

Cooley, W.H. and Paul R. Lohnes. *Multivariate Analysis*. Wiley, New York, 1971, pp. 243-250.

EXAMPLE

Two factors, using equal groups for classification.

Equal Groups

Dataset TALENT

Excluding V1>200

Group definitions: V2=2/3 School Size

138 cases accepted

Mahalanobis distance: .37

Frequencies

	1st GROUP	2nd GROUP	Total
N	60	78	138
Percent	43.5	56.5	100.0

Univariate F-ratios

	Among Mean SQ	Within Mean SQ	F-Ratio	NDF1	NDF2	ETA Squared	Probability	
V10	4248.21	1162.49	3.65	1	136	0.0262	0.05	Information Test I
V11	1319.30	284.74	4.63	1	136	0.0329	0.03	Information Test II

Wilks' Lambda = .967 F-Ratio = 2.307 NDF1 = 2 and NDF2 = 135 Probability = .101

Coefficients

		Function 1
Information Test I	V10	0.0032
Information Test II	V11	0.0529

Standardized coefficients

		Function 1
Information Test I	V10	0.1093
Information Test II	V11	0.9041

Factor Structure

1

Information Test I V10 0.884
Information Test II V11 0.998

Communalities

V10 V11
0.782 0.996

Percentage of Trace of R accounted for by each root

1
88.870

Centroids in 1-dimensional discriminant space

Function 1
1st GROUP -0.207
2nd GROUP 0.159

Classification Matrix (Actual cases in row categories classified into predicted column categories)

	1st GROUP	2nd GROUP	Totals	Percent correct
1st GROUP	36	24	60	60.00
2nd GROUP	33	45	78	57.69
Totals	69	69	138	58.69

Above table based on expected equal group size.

EDIT DATASET

File Assignments: DATASET Input dataset

GENERAL DESCRIPTION

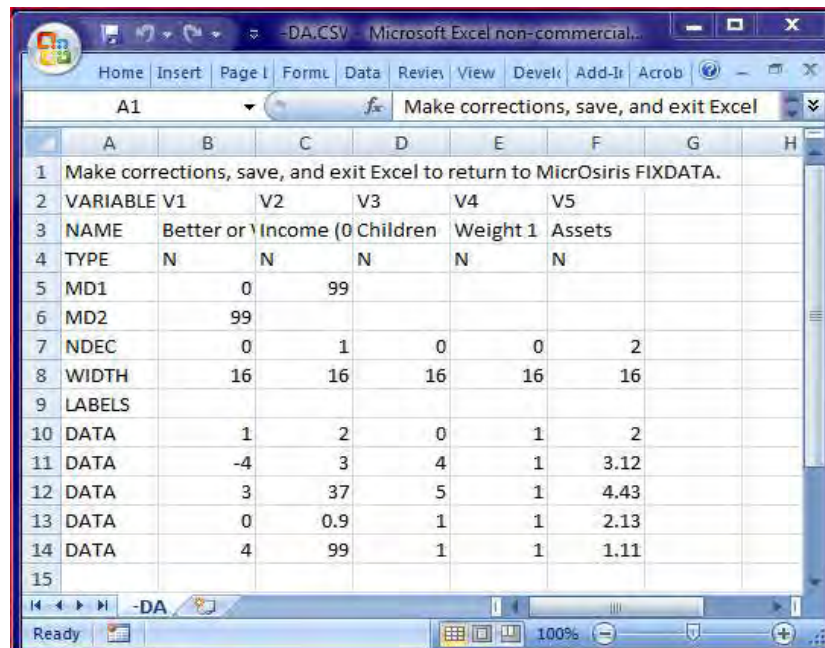
Displays a dataset or just the dictionary portion as an Excel spreadsheet for corrections, adding records, and modifying a dictionary, including adding variable value labels.

EDIT DATASET is especially useful for correcting errors that occur with no particular pattern across the records in the data file. These types of errors are those which may be reported as "bad data" or wild codes by [WILDCODE CHECK](#), or which have been uncovered by consistency checking with RECODE or TABLES.

Consider using RECODE with TRANSFORM to correct systematic errors. For instance, when all occurrences of V1=1 must be changed to 2, use RECODE statement "IF V1 EQ 1 THEN V1=2"

COMMAND FEATURES

MicroSiris converts the dataset into an intermediate CSV file and passed to Microsoft Excel or equivalent. Instructions appear when you move onto a cell.



	A	B	C	D	E	F	G	H
1	Make corrections, save, and exit Excel to return to MicroSiris FIXDATA.							
2	VARIABLE	V1	V2	V3	V4	V5		
3	NAME	Better or	Income (0 Children	Weight 1	Assets			
4	TYPE	N	N	N	N	N		
5	MD1	0	99					
6	MD2	99						
7	NDEC	0	1	0	0	2		
8	WIDTH	16	16	16	16	16		
9	LABELS							
10	DATA	1	2	0	1	2		
11	DATA	-4	3	4	1	3.12		
12	DATA	3	37	5	1	4.43		
13	DATA	0	0.9	1	1	2.13		
14	DATA	4	99	1	1	1.11		
15								

Excel applies no formatting to a [CSV](#) file so it's easier to make corrections if you first right justify and auto size column one, and center row 2.

Variable value labels (LABELS line) are entered in the form: 1=label 1,2=label 3,....

The labels must be entered as one string and repeated for each variable that requires them.

Labels may not contain a double quote (").

Examples: 1=F,2=M
 0=No Car,1=Small,2=Compac,3=Mid Size,4=Large

Make corrections to the dictionary and data rows (columns 2-n) and close the file.

RESTRICTIONS

Do not change variable types or variable widths unless you are sure the dictionary does not match your data file; the results are unpredictable and the dataset may be rendered unusable.

Do not change anything in column one or row one. All numeric variables in the new output file will be double precision floating-point of width 8.

EXPORT DATA TO OTHER SYSTEMS

File Assignments:	DATASET	Input data
	CVS FILE	Output csv file (conditional)
	SASRUN	SAS Runfile (conditional)
	SASDATA	Output datafile (conditional)

GENERAL DESCRIPTION

Creates a [CSV](#) or fixed-length text file for use with other systems, such as SAS, SPSS, database or spreadsheet programs, from a MicroSiris dataset.

A CSV text file contains fields separated by some kind of delimiter, usually a comma; it may use a special quote character to enclose text within fields. A fixed-format text file is a file with no field separators or special quote characters, wherein all records are exactly the same length.

COMMAND FEATURES

Exporting Data to Excel. Select option CSV. Excel will read CSV files directly.

Exporting Data to SPSS. Use CSV format with NAMES. See also [Using SPSS data in MicroSiris](#)

Exporting Data to SAS. Use the SAS option to create a SAS (release 6.04) DATA step Runfile. The file assigned to SASRUN will contain an SAS data step with an INFILE statement pointing to the output file assigned to SASDATA, an INPUT statement for all selected variables, a LABEL statement for each variable, and a SAS FORMAT statement indicating how many decimal places to display. The variable names will be the first eight characters of the MicroSiris variable name (up to the first blank or quotation mark). The SAS filename assigned on the DATA statement will be the first 8 characters of the MicroSiris filename (up to the "." if present). See also [Using SAS data in MicroSiris](#).

SPECIAL TERMINOLOGY

Character Numeric: Each digit of a number, its sign or exponent indicator, occupies one location of the total field reserved for it on a storage device.

OPTIONS

CSV FIXED SAS	Type of output file desired.
CSV	CSV text file with delimiters. (Default)
SAS	Create a SAS DATA step Runfile and matching data file.
<i>Default: CSV</i>	

RECODE=n Use RECODE n, previously entered via the RECODE command.

VAR= (variable numbers)|ALL)

Use the variables specified in the list. If V=ALL is given, all variables in the dictionary are used plus any RECODE variables.

For CSV-format files only

DELIMITER=c

C is the delimiter; EXPORT will separate variable values in the data file with the character "c."

Default: EXPORT uses a comma as the delimiter.

QUOTE=c

c is the QUOTE character; alphabetic strings in the data file are enclosed with character "c" if they contain blanks or the DELIMITER character.

Default: Use quotation marks (") as the QUOTE character.

This is NOT the same as two consecutive primes, i.e., NOT (").

NAMES

Put the variable names in the first row.

FACTOR ANALYSIS

File Assignments:	DATASET	Input data
	MATIN	Input correlation matrix
	VARIMAX	Varimax rotated factor matrix (optional)
	OBLIMIN	Oblimin rotated factor matrix (optional)
	SCORES	RECODE statements for factor scores (optional)

GENERAL DESCRIPTION

A general factor analysis command that includes numerous options for the application of various factor analytic tools. Separate factor analyses may be performed on subsets of variables in a single run.

COMMAND FEATURES

Missing data (raw data only)

Missing data can be handled by either pair-wise deletion or case-wise deletion.

With pair-wise deletion a case is used for computation as long as it has valid data for both variables in the pair (see option DELETE=PAIRS). Thus, a case may be used in the computations for some pairs of variables and not used for other pairs. If there are missing data, correlations may be computed on different subsets of the data. Large amounts of missing data can lead to internal inconsistencies in the correlation matrix causing difficulties or the analysis, including abnormal termination. Consider using IVEWARE to impute for missing-data and use case-wise deletion instead.

With case-wise deletion a case is used for calculation only if it has valid data for all variables in the variable list (see option DELETE=CASES). Thus, a case is either used in computation for all pairs of variables or not used at all. The disadvantage of case-wise deletion is a relatively greater loss of data.

You can use the **IVEWARE** command to first impute missing data if desired.

Output Matrices: The principal-axes factor matrix, the varimax rotated factor matrix, and the oblimin transformation matrix may be saved for later use by MicroSiris commands. Factor matrices may be used as input to COMPARE.

Factor Scores: For each factor solution (principal axes, varimax, and oblimin), factor score coefficients may be computed. You may also request RECODE statements for computing factor scores. These RECODE statements work as follows:

-A data case is REJECTED if there is missing data for any of the input variables.

-The score for Factor 1 will be variable Rn, with name 'FACTOR 1', the score for Factor 2 will be variable Rn+1, with name 'FACTOR 2', etc., where Rn is set by the SCORES option/keyword.

Use TRANSFORM to save the scores permanently as new variables.

Factor Extraction: The factor extraction algorithm follows the principal-axes method of Hotelling. A Wilkinson tridiagonalization technique is used to obtain the eigenvectors of the correlation matrix and their corresponding eigenvalues (characteristic vectors and values). This procedure does not extract factors successively, but always solves for all eigenvalues simultaneously within a given level of accuracy and sorts them into descending order. You can request a specified number of residual correlation matrices to display after factoring.

Principal-Components Analysis: A principal-components analysis is always performed first to determine the proper number of factors and to estimate communalities. The results may be used as the starting point for iterative re-factoring.

Estimation of Communalities: You can specify one of three types of initial communality estimates:

- Unities (principal-components analysis)

- Squared multiple correlations

- User-supplied estimates (these might be from theoretical considerations or prior factor analyses)

In any of these cases, you can direct FACTOR ANALYSIS to re-factor the matrix a specified number of times (see Harman, 1960: p. 85), beginning with the desired initial communality estimates. After each iteration, the current estimates are replaced by the communalities obtained from re-factoring, taking into account only the number of factors to be retained. Iteration continues until the communalities converge within a given tolerance. This iteration procedure may be used to attempt a theoretical fit of the factor model to the data by requiring an empirical stabilization of the factor solution.

Rotation: An orthogonal (varimax) and/or oblique (oblimin) rotation may be performed on the factor matrix. If both are desired, the varimax rotation is performed first and generally speeds convergence in the subsequent oblimin solution.

For either rotation, you may request prior normalization in order to avoid bias caused by unequal communalities, and may adjust certain internal parameters relating to accuracy and convergence tests. The complete class of oblimin solutions is available, including the biquartimin and covarimin solutions.

PRINTED OUTPUT

Means and Standard Deviations: (Optional: see option PRINT=USTATS.)

Covariances: (Optional: see option PRINT=MOMENTS.) The lower left triangle of the variance-covariance matrix).

Correlation Matrix: (Optional: see option PRINT=CORREL.) The correlations matrix is displayed with variable names and numbers labeling the rows and columns.

Characteristic Roots: For all factors, the lambda, percentage of variance, and cumulative percentage are displayed, followed by the sum of the lambdas. This is the result of the principal-components analysis.

Characteristic Vectors: (Optional: see option PRINT= VECTORS.) The first column vector corresponds to the first lambda, etc.

Determinant of the Inverse Matrix.

H-Squared Minimum: A row vector of the lower bounds of the communalities (see also option COM=(HSQ)). These are the squared multiple correlations.

Characteristic Roots of Factor Analysis: If communalities were iterated, the lambdas are displayed for the factor analysis.

Characteristic Vectors of Factor Analysis: (Optional: see option PRINT=VECTORS.) Will be included only if iterative factor analysis was performed.

Factor Matrix: These are the loadings of each factor on each variable. Also displayed for each factor are: the sum of squared loadings, percentage of total variance, percentage of common variance, and cumulative percentages.

Final Communality Estimates: For each variable, the final input and output communality estimates are displayed, followed by their sums.

Squared Loadings: (Optional: see option PRINT=SQL.) Squared entries from the factor matrix. The last column contains the row sums, which are the final output communalities. The column sums are also displayed.

Residual Correlation Matrices: (Optional: see option NRESID.) These are displayed as square symmetrical matrices.

Score Coefficients: (Optional: see option SCORES.) If scores are requested, FACTOR ANALYSIS prints the factor score coefficients in standard score form, the squared multiple correlations and multiple correlations, the raw score coefficients, and the constant terms.

Varimax Rotation: (Optional: see option VARIMAX.) The varimax rotation history is displayed, giving the criterion value and difference for each cycle. The rotated factor matrix and the transformation matrix are both displayed. Score coefficients also may be requested.

Oblimin Rotation: (Optional: see option OBLIMIN.) The oblimin rotation history is displayed, giving the criterion value, difference, vector convergence, and the number of minor cycles for each major cycle. The reference structure V, lambda transformation, reference factor correlations, diagonal matrix D, the primary pattern P, the T transformation, the primary factor correlations and contributions, and the primary structure S are displayed. Score coefficients also may be requested.

INPUT DATA

Raw Data Input: Raw data input (MATRIX=n option not used) consists of a MicrOsiris dataset. Use **IVEWARE** to impute missing data first if desired.

Matrix Input: Matrix input (MATRIX=n option used) consists of a standard symmetric MicrOsiris correlation matrix with means and standard deviations included.

OPTIONS

Choose FACTOR ANALYSIS from the command screen and make selections.

For a Runfile use: FACTOR ANALYSIS
Filter statement (optional)
Job Title
Keyword choices from below

MATRIX=n The number of a correlation matrix produced by CORRELATIONS or assigned to MATIN.

Default: Raw data input.

PRINT=(CORREL,SQL,VECTORS)

CORREL Print the correlations.

SQL Print the matrix of squared loadings.

VECTORS Print the characteristic vectors.

Options for Raw Data Input Only:

DELETE=PAIRS|CASES

PAIRS Pair-wise deletion.

CASES Case-wise deletion.

Default: DELETE=PAIRS.

RECODE=n Use RECODE n, previously entered via the RECODE command.

WT=n Use variable n as a weight variable

VARS=variable numbers Use the variables specified in the list.

COMMUNALITIES=(UNITIES|SMR|ESTIMATES=(d1-dn),CONV=c, HSQ, MAXCYCLE=n)

UNITIES Use 1's for initial communalities (the default).

ESTIMATES=(d1-dn) List of initial communality estimates.

SMR Use squared multiple correlations for initial communalities.

CONV=c Convergent criterion for the communalities. Relevant only if MAXCYCLE is not zero. *Default:* CONV=0.005

HSQ Allow communalities to drop to 0.
Default: Communalities will not be allowed to drop below the squared multiple correlations.

MAXCYCLE=n

Maximum iterations to perform on the communalities.

Default: MAXCYCLE=0

DIAGONALIZATION=(MAXCYCLE=n)

Sets the maximum number of iterations per individual eigenvalue in the

Wilkinson tridiagonalization.

Default: MAXCYCLE=30

KAISER|MINVARIANCE=n

The criterion for determining the number of factors:

KAISER Kaiser's criterion (number of roots greater than 1.0)

MINV=n The minimum percent variance to be explained by the factors taken all together (e.g., MINV=85).

Default: Use the number of factors given by MAXFACTORS.

MAXFACTOR=n

Maximum number of factors desired.

Default: The number of analysis variables.

NRESID=n Display n residual correlation matrices.

Default: NRESID=0.

SCORES=Rn Generate recode statements for the principal axes scores. The RECODE number will be the analysis number and the first factor will be variable Rn. successive factors will be Rn+1, Rn+2, etc.

VARIMAX=(MATRIX=n, WRITE, RAW, NCYCLE=n, MAXCYCLE=n, MINANGLE=a, CONV=c, SCORES=n, TITLE=name)

MATRIX=n Assigns number n to the varimax rotated factor matrix.

Default: None: required when WRITE is specified or matrix will be used by a subsequent command.

RAW Compute the unnormalized solution.

Default: Normalize the factor matrix before rotating.

NCYCLE=n The number of cycles for which the new value of the varimax criterion must equal the old value.

Default: NCYCLE=3

MAXCYCLE=n

Maximum number of cycles allowed. If the criterion does not converge in n cycles, the most recent loadings are used.

Default: MAXCYCLE=50

MINANGLE=a

An angle must be greater than a to be considered different from zero (one minute of arc=0.00116, and one degree=0.06993).

Default: MINA=0.00116

CONV=c Convergence criterion for considering two successive rotation values equal.

Default: CONV=0.0000001

SCORES=Rn

Generate recode statements for the principal axes scores. The RECODE number will be the analysis number and the first factor will be variable Rn. successive factors will be Rn+1, Rn+2, etc.

TITLE=name

1-100 character name for the rotated matrix

WRITE

Write the varimax rotated factor matrix to the file assigned to MATOUT.

OBLIMIN=(MATRIX=n,WRITE,SCORES=n,RAW,NCYCLE=n,MAXCYCLE=n, GAMMA=g,
MINROOT=a,CONV=c,TOL=t,TITLE=name)

MATRIX=n

Assigns number n to the oblimin rotated factor matrix.

Default: None: required when WRITE is specified or matrix will be used by a subsequent command.

WRITE

Write the oblimin transformation factor matrix to the file assigned to MATOUT.

SCORES=Rn

Generate recode statements for the principal axes scores and print the score coefficients. The RECODE number will be the analysis number and the first factor will be variable Rn. successive factors will be Rn+1, Rn+2, etc.

RAW

Compute the unnormalized solution.

Default: Normalize the factor matrix prior to rotation.

NCYCLE=n

Number of major cycles for which the new value of the oblimin criterion must equal the old value.

Default: NCYCLE=3

MAXCYCLE=n

Maximum number of iterations allowed in solving for a given eigenvector. If a vector doesn't converge within n iterations, the values of the last iteration are used.

Default: MAXCYCLE=100

GAMMA=g

The controlling parameter for the oblimin rotation. Any value between 0.0 and 1.0 may be used; of particular interest are: 0.5 Biquartimin method (Carroll) 1.0 Covarimin method (Kaiser)

Default: GAMMA=0.5

MINROOT=a

The number to add to the diagonal of a matrix to ensure that all eigenvalues will be positive.

Default: The MINROOT value will be $0.5 * \text{GAMMA} * (\text{number of factors}) * \text{number of variables squared}$)

CONV=c Convergence criterion for terminating rotation. If the difference in the oblimin criterion on two successive cycles is less than c, the rotation has converged; otherwise, if the current value is at least 1.5 times greater than the previous one, the rotation has diverged and the current analysis is terminated.
Default: CONV=0.05

TOL=t Convergence constant for eigenvector solution.
Default: TOL=0.0001

TITLE=name
1-100 character name for the rotated matrix

REFERENCES

Harman, H. H. "Factor Analysis," in *Handbook of Measurement and Assessment in Computers*, edited by D. K. Whitla. New York: Wiley, 1960. (Harman presents here the derivation and method for the principal factor solution.)

EXAMPLE

Factor analysis of legislative roll call data with a normalized varimax rotation.

ROLL CALL DATA										
Using matrix 1, Roll Call 85 Data, based on 443 cases from RC85.MTX										
PRINCIPAL COMPONENTS ANALYSIS										
THE CHARACTERISTIC ROOTS										
	LAMBDA		PER CENT				CUMULATIVE			
1	3.93956		78.79%				78.79%			
2	0.68259		13.65%				92.44%			
3	0.27184		5.44%				97.88%			
4	0.06649		1.33%				99.21%			
5	0.03951		0.79%				100.00%			
SUM	5.00000									
DETERMINANT (R) = 0.19206E-02										
H-SQUARED MINIMUM										
	V1		V2		V3		V4		V5	
APR04 1957	APR04 1957		APR05 1957		APR17 1957		JUN05 1957			
HR 6387	HR 6287		HR 6387		HR 6871		HRE 259			
0.91487	0.91280		0.94452		0.68172		0.26581			
FACTOR MATRIX LOADINGS										
	FACTOR 1		FACTOR 2		FACTOR 3		FACTOR 4		FACTOR 5	

APR04	1957	HR	6387	V1	-0.95938	-0.11598	-0.16522	0.18111	0.07774
APR04	1957	HR	6287	V2	-0.95542	-0.14934	-0.16106	-0.18335	0.07293
APR05	1957	HR	6387	V3	-0.97379	-0.12102	-0.09625	0.00434	-0.16672
APR17	1957	HR	6871	V4	-0.87940	-0.13042	0.45750	0.00186	0.01863
JUN05	1957	HRE	259	V5	0.62026	-0.78434	-0.00612	0.00715	-0.00275

FACTOR CONTRIBUTIONS

	FACTOR 1	FACTOR 2	FACTOR 3	FACTOR 4	FACTOR 5
SUM SQUARED LOADINGS	3.93956	0.68259	0.27184	0.06649	0.03951
% TOTAL VARIANCE	78.79120	13.65186	5.43683	1.32983	0.79028
% CUM. TOTAL VARIANCE	78.79120	92.44306	97.87989	99.20972	100.00000
% COMMON VARIANCE	78.79120	13.65186	5.43683	1.32983	0.79028
% CUM. COMMON VARIANCE	78.79120	92.44306	97.87989	99.20972	100.00000

COMMUNALITY ESTIMATES

VARIABLE	FINAL INPUT	OUTPUT
V1	1.00000	1.00000
V2	1.00000	1.00000
V3	1.00000	1.00000
V4	1.00000	1.00000
V5	1.00000	1.00000
SUM	5.00000	5.00000

(COMMON VARIANCE)

Factor	Squared Multiple Correlation	Multiple Correlation
1	1.000000	1.000000
2	1.000000	1.000000
3	1.000000	1.000000
4	1.000000	1.000000
5	1.000000	1.000000

IMPORT DATA

GENERAL DESCRIPTION

IMPORT loads data into MicroSiris from an Excel file created with the MicroSiris [Excel template](#), an SPSS PC portable file, or a [CSV](#) (Commas Separated Value) text file or an existing fixed-format file. A fixed-format file is a file with no field separators or special quote characters, wherein all records are exactly the same length.

It is started by selecting the IMPORT button on the command screen.

You can IMPORT data from CSV files created by a spreadsheet program, a database program, an accounting package, SAS, or even a word processor or text editor.

For options SPSS and CSV, IMPORT creates a data dictionary where all numeric variables become floating-point variables in MicroSiris of width 4 (default, single precision, 7 significant digits) or 8 (double precision, 16 significant digits), but MicroSiris always uses double precision for calculations.

Using [TRANSFORM](#) to change variable types and field widths to [character numeric](#) can save considerable space for large datasets with relatively few decimal values.

Due to the maximum output record length of 80000, at most 10000 variables can be imported from SPSS or a CSV file when width 8 is used.

COMMAND FEATURES

CSV files

IMPORT reads records from the input file and assumes values in the order they are found. Variables with no values or consisting of a single period are optionally assigned the default first missing-data code 1,500,000,000 or set to a user specified value. Bad data values detected in the input records or in converting input values to the output record are treated as missing-data and given the default second missing-data code 1,600,000,000. Input values consisting of a single period are considered missing-data.

Importing from SPSS

The input file must be an SPSS PC [portable](#) (.por) file (See [Using SPSS data in MicroSiris](#) for treatment of SPSS missing-data codes.) Note: MicroSiris only imports numeric, date, and alphabetic types. It imports the short names or the long names if present (up to 24 characters) and any variable label codes present in the file.

Importing from EXCEL

Save your file with the file extension .CSV; Excel will automatically create a CSV file that IMPORT can read. See [Importing data using Excel or Works](#) for an alternative.

Importing from SAS

Use the SAS EXPORT proc to create a CSV file from your SAS dataset. Choose the IMPORT button on the command screen and specify CSV. Use the ALPHA list to indicate which variables are alphabetic and specifying their widths.

See also [Using SAS data in Microsirir](#).

Importing from OSIRIS

OSIRIS is a general purpose statistical package written for use on IBM mainframes. It is no longer actively supported. However, an enormous store of survey data is available in OSIRIS format from the Inter-University Consortium for Political and Social Research (ICPSR) at the University of Michigan.

OSIRIS datasets are similar to Microsirir datasets, comprised of a dictionary file and a data file. There are two significant differences: 1) the dictionary file, known as a type 1 dictionary, is a [binary](#) (internal number storage format) file limited to smaller variable numbers and missing-data codes than the Microsirir dictionary. 2) Since OSIRIS was developed on IBM mainframes, the dictionary and data files were coded in EBCDIC (E^xtended Bⁱinary C^oded D^ecimal Iⁿterchange C^ode) instead of ASCII, which is used on PCs.

Microsirir reads and interprets the type 1 dictionaries available from ICPSR and converts the EBCDIC portions to ASCII as needed.

The data file, originally stored in EBCDIC format, comes from ICPSR, usually is already in ASCII format and needs no conversion. If it is not in ASCII format you must use the "Convert to ASCII" button on the command screen.

Fixed-format files

For fixed-format files, IMPORT creates a Microsirir dictionary file according to your specifications to match the existing file. The file may contain character numeric, alphabetic, or floating-point binary values.

OUTPUT DATA

Except for fixed-format files, IMPORT creates a Microsirir dataset where all numeric variables are floating-point variables (TYPE=F) in Microsirir of width 8. For fixed-format files the data fields remain the same and a matching dictionary file is created.

CSV files:

Alphabetic variables, if present, are described using the option ALPHA, e.g., ALPHA=(V1:2,V2:12), which indicates variables V1 and V2 are alphabetic and specifies their widths.

Alphabetic strings exceeding their variable width (8 by default, or specified with the ALPHA option) are truncated to the variable width. Alphabetic strings shorter than their variable width are padded on the right with blanks.

QUOTE characters are not identified as part of an alphabetic string unless consecutive; in that case IMPORT reduces them to a single QUOTE character.

Missing or bad data are converted to the first and second default missing-data codes respectively.

OPTIONS

Options are selected from an interactive window. The choices are:

CSV | SPSS | OSIRIS | FIXED

Type of input file.

For CSV files:

Alphabetic variable lengths:

(Vn1:m1,Vn2-m2,etc.) Describes alphabetic variables, where Vn is the variable number and m is its field width.

Value separator

Specifies the delimiter used in the CSV; variable values are separated in the data file by the delimiter.

Quote character

Alphabetic strings in the data file are enclosed with the character "c" if the contain blanks or the DELIMITER character.

Names

Variable names will be taken from the first row of the CSV file if the Names box is checked.

Decimal places

The number of decimal places for each variable are taken from the second row of the CSV file if the decimal places box is checked and NAMES box is also checked.; otherwise from the first row. If more decimal places are found in the data, the large of the two will be used for the dictionary.

Set blank values to: Missing data or a specific value.

For OSIRIS type 1 files:

Convert data file to ASCII

Indicates the OSIRIS data file is coded as EBCDIC and needs to be converted to ASCII.

For Fixed-format files a screen with the following variable descriptor option appears:

VARS=(variable numbers)

A list of variables being defined (range 1-999999). All variables in the list will have the same attributes defined by the remaining options. The list is sorted before use.

NAME='...' Variable name, up to 24 characters, left justified. If V=list, a unique number is added to the end of NAME for each variable name, starting with 1.

Variable type: Character numeric, alphabetic or floating.

Character numeric corresponds to numbers entered from the keyboard, e.g., with Notepad, WordPad, or a word processor.

Floating responds to floating-point binary.

LOCATION=n The starting location of the variable within each record.

WIDTH=n The width of the variable. Permissible width of a variable depends upon the variable type as shown in the following table:

TYPE	WIDTH
character numeric	1-15
alphabetic	1-999

NDEC=n The number of decimal places implicit in the variable for TYPE='C' if no explicit decimal point exists. Must be in the range 0-9.

MD1=value The first missing-data code for a numeric variable; must be integral value, up to nine digits wide.

MD2=value The second missing-data code for a numeric variable; must be integral value, up to nine digits wide.

LABELS '1=label_1,2=label _2,...,'
Variable value labels. Label_i is the character label to associate with code i. labels may not contain a double quote ("). All codes must be integers.

IMPUTE

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

IMPUTE performs a variety of descriptive and model based analyses accounting for such complex design features as clustering, stratification and weighting including multiple imputation analyses for both descriptive and model-based survey statistics.

IMPUTE invokes the SrcWare version of IVEware installed with MicroSiris to perform the imputation.

IMPUTE returns an imputed data set for further analysis in MicroSiris with other commands.

COMMAND FEATURES

IMPUTE produces imputed values for each individual in the data set conditional on all the values observed for that individual. The approach is to consider imputation on a variable-by-variable basis but to condition on all observed variables. IMPUTE creates imputations through a sequence of multiple regressions, varying the type of regression model by the type of variable being imputed. Covariates include all other variables observed or imputed for that individual. The imputations are defined as draws from the posterior predictive distribution specified by the regression model with a flat or non-informative prior distribution for the parameters in the regression model.

Sets of variables in the dataset are specified as one of the following five types: continuous, binary, categorical (polytomous with more than two categories), counts, and mixed (a continuous variable with a non-zero probability mass at zero).

The types of regression models used are linear, logistic, Poisson, generalized logit or mixed logistic/linear, depending on the type of variable being imputed.

IMPUTE can restrict imputations to subpopulations, and bounded imputed values.

By default, IMPUTE perturbs model coefficients using a multivariate normal approximation of the posterior distribution and the predicted values using the appropriate regression model conditional on the perturbed coefficients. Option PERTURB=SIR uses the Sampling-Importance-Resampling algorithm to generate coefficients from the actual posterior distribution of parameters in the logistic, polytomous and Poisson regression models. This is appropriate in situations where normal approximation to the posterior distribution is not appropriate.

See the [IVEware User Guide](#) for more information.

OUTPUT DATA

A Microsirir imputed set to use with other Microsirir commands. If the MULTIPLES option is used, all multiples of the dataset are included; and two new variables are added, 'Multiple number', and 'Observation number'.

Note: R type recoded variables used in the analysis will be changed to V type variables in the output dictionary. Be sure and use R type variable numbers that do not match any unrecorded variables used.

IMPUTE creates name.set and name.data, where name is the name of the input dataset, which are submitted to SrcWare. You can save these (SAVE option) for later modification and refinement and use them directly with Srcware. See [Srcware User Guide](#) for details.

Alphabetic variables may not be used or transferred.

RESTRICTIONS

Alphabetic variables may not be used.

REFERENCES

[IVEware](#) was developed by the Survey Methodology Program at The University of Michigan's Survey Research Center, Institute for Social Research.

"A multivariate technique for multiply imputing missing values using a sequence of regression models" by Raghunathan, Lepkowski, Van Hoewyk and Solenberger (*Survey Methodology*, June 2001).

OPTIONS

Choose IVEWARE from the command screen and make selections.

PRINT=(DETAILS,COEF)

DETAILS Detailed distribution information for each variable.

COEF Print unperturbed and perturbed coefficients for each iteration of each multiple.

RECODE=n Use RECODE n, previously entered via the RECODE command.

ITERATIONS=n

The number of cycles for the imputation. Specify any number greater than 1. About 10 cycles are sufficient for most imputations. *Default: 2.*

SEED=n Specifies a seed for the random draws from the posterior predictive distribution. n should be greater than zero. A zero seed results in no perturbations of the predicted values or the regression coefficients. *Default: SEED is determined by your computer's internal clock.*

MULTIPLES=n

The number of imputations to be performed. Multiples and iterations determine p (see page 11 of original IVEware documentation). If multiples were specified

as 5 and iterations as 10 then a total of 50 cycles will be performed. After every 10th cycle an imputed data set is created. *Default: 1*

PERTURB=SIR

Use the Sampling-Importance-Resampling algorithm to generate coefficients from the actual posterior distribution of parameters in the logistic, Poisson and polytomous regression models.

Default: Use a multivariate normal approximation of the posterior distribution and the predicted values using the appropriate regression model conditional on the perturbed coefficients.

VARS=variable numbers List of continuous variables.

CATVARS=variable numbers List of categorical variables.

MIXVARS=variable numbers

Mixed variables are both categorical and continuous. In a mixed variable a value of zero is treated as a discrete category, while values greater than zero are considered continuous. Alcohol consumption is an example of a mixed variable. A two stage model is use to impute the missing values. First, a logistic regression model is used to impute zero vs. non-zero status. Conditional on imputing a non-zero status, a normal linear regression model is used to impute non-zero values.

COUNTVARS=variable numbers

Count variables have non-negative integer values. A Poisson regression model is used to impute the missing values. The number of annual doctor visits is an example of a COUNT variable.

TRANSFER=variable numbers

Transfer variables are carried over to the imputed data set, but are not imputed nor used as predictors in the imputation model.

INTERACT=Vn_1*Vn_2, Vn_3*Vn_4,etc.

List of interactions. These are labeled XACT1-XACTn in the SrcWare ouput.

RESTRICT=Vn(logical expression)

Restrict the imputation of a variable to those observations that satisfy the logical expression. For example, suppose that the variable V1 indicates the number of years an individual smoked, and the variable V2 takes the value 1 for a current smoker, 2 for a former smoker or 0 for someone who never smoked. Then V1(V2=1,2) imputes V1 values only for current and former smokers. Restrictions on more than one variable may be combined as follows: V1(V2=1,2) V3(V4=2) V5(V6=1). When the restriction is not met, the value of the restricted variable will be set to zero for continuous variables and one higher than the highest observed code for categorical variables.

IMPUTE does not check the logical expression for syntax; this is done by the IVEware software.

BOUNDS=Vn(logical expression)

Used to restricting the range of values to be imputed for a variable. For example,

if V1 is the number of years an individual smoked and V4 is age, then V1 (> 0, <= V2-12) ensures that the imputed values for V1 are between 0 and the individual's age minus 12. Smoking is assumed not to begin before the age of 12. More than one variable can be included in the BOUNDS statement, e.g., V1 (>0, <= V2-12) V14(>0))

IMPUTE does not check the logical expression for syntax; this is done by the IVEware software.

MAXPRED=n Maximum number of predictors (stepwise regression performed).

MINRSQD=decimal number

Minimum marginal r-squared for a stepwise regression. (Minimum initial marginal r-squared for a logistic regression, and minimum initial r-squares for any code being predicted for a polytomous regression.) A small decimal number like 0.005 can build large, time-consuming, regression models while 0.25 will include a smaller number of predictors in the regression models. If neither MAXPRED nor MINRSQD is set no stepwise regression is performed.

MINCODI=n Specifies the minimum proportional change in any regression coefficient to continue the logistic regression iteration process. Applies to the convergence criterion for the logistic, polytomous and Poisson regression models.

EXAMPLE

Imputing values for a categorical variable.

```

*** IMPUTE (IVEWARE) ***

Dataset C:\DEVELOPMENT\PROJECTS\TESTDATA\SCF

1 Variables and 327 cases written

Executing IVEware...

Building dictionary for imputed data file...

IMPUTED PROCEDURE

Setup listing:

DATAIN SCF_data;
PRINT=STANDARD;
DEFAULT CONTINUOUS;
CATEGORICAL V37;
ITERATIONS 2;

OUTPUT FOR IMPUTED

Variable          Observed          Imputed    Double counted
      V37: Race      326              1              0

Variable V37

      Code      Observed          Imputed          Combined
              Freq    Per      Freq    Per      Freq    Per

```

1	299	91.72	1	100.00	300	91.74
2	24	7.36	0	0.00	24	7.34
3	3	0.92	0	0.00	3	0.92
Total	326	100.00	1	100.00	327	100.00

Dataset: C:\DEVELOPMENT\PROJECTS\TESTDATA\SCF_imputed
Standard

Variables: 1
Codeframes: 1
Observations: 327

Created imputed dataset SCF_imputed

INDEX RELIABILITY

File Assignments

DATASET
MATIN

Input data (conditional)
Input matrix (conditional)

GENERAL DESCRIPTION

Computes various statistics for the internal consistency reliability and criterion-related validity of composite measures. The command works from the correlation matrix and standard deviations for the items, allowing great flexibility in trying out various ways of composing indices. A typical use is to compute index alphas and item-to-total correlations during data reduction. However, the combination of the ALPHA and ITEM options with the COMPOS or CRITER options can be used to improve indices and to improve their relationship to a criterion index or item. See also [ITEM ANALYSIS](#).

COMMAND FEATURES

INDEX RELIABILITY produces, optionally:

Alpha	Cronbach's alpha coefficient of index reliability.
Item	Item-to-total index correlation coefficient.
Compos	Correlation between two sets of composite indices.
Scott	Scott's homogeneity ratio coefficient.
Step	A step-wise item analysis.
Criter	Correlation coefficients of an index with a criterion measure.

INPUT DATA

A Pearson correlation coefficient matrix of item responses , with means and standard deviations.

PRINTED OUTPUT

The output consists of any of the six statistics listed under COMMAND FEATURES above, plus the correlation and variance matrices if requested.

OPTIONS

Choose INDEX RELIABILITY from the command screen and make selections.

For a Runfile use:	INDEX RELIABILITY Filter statement (optional) Job Title Keyword choices from below
--------------------	---------------------------------------------------------------------------------------------

MATRIX=n The number of a correlation matrix produced by CORRELATIONS or assigned to MATIN.
 Default: Raw data input.

PRINT=(CORR,COVAR)
 CORR: Print the input correlation matrix.
 COVAR: Print the covariance matrix.

REVERSE=variable list Reverse the signs of the item variables in the list.

Options for Raw Data Input

DELETE=PAIRS|CASES
 PAIRS Pair-wise deletion.
 CASES Case-wise deletion.
 Default: DELETE=PAIRS.

RECODE=n Use RECODE n, previously entered via the RECODE command.

WT=n Use variable n as a weight variable

Analysis Statements (as many as required)

ALPHA|ITEM|COMPOS|CRITER|STEP|SCOTT
 ALPHA: Cronbach's alpha.
 ITEM: Item-to-total index correlation coefficient.
 COMPOS: Correlation between two sets of composite indices. Requires
 VARS=list1/list2 where list1 and list2 have the same number of
 variables.
 SCOTT: Scott's homogeneity ratio coefficient.
 STEP: Stepwise item analysis.
 CRITER: Correlation coefficients of an index with a criterion measure.
 Default: ALPHA.

VARS=variable numbers Use the variables specified in the list.

CVAR=Vn Criterion variable for CRITER.

MIN=n Step minimum for stepwise item analysis.

NAME=string A 1- to 60-character name for the analysis or scale.

EXAMPLE

Cronbach's alpha, item-to-total index correlation, and Scott's homogeneity ratio coefficient.

```
*** INDEX RELIABILITY ANALYSIS ***  
  
CHECKING INDEX RELIABILITY  
  
Using matrix 1 CORRELATIONS
```


ANALYSIS 1 Cronbach Alpha

V1(Better or Worse),V2(Income(000)),V3(Children)

Alpha Coefficient = .4362

ANALYSIS 2 Item-to-Total Index Correlation Coefficient Example

V1(Better or Worse),V2(Income(000)),V3(Children)

Item-Total-Correlation Coefficients for 3 Items:

Variable	Correlation Coefficient	
V1	.6363	Better or Worse
V2	.9902	Income(000)
V3	.5852	Children

ANALYSIS 3 Scott's homogeneity ratio coefficient

V1 Better or Worse, V2 Income(000), V3 Children

Scott's Homogeneity Ratio = .6376

ITEM ANALYSIS

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

Performs an internal consistency analysis of a questionnaire scored on the Likert 5-point scale according to Kuder and Richardson's equation for scale reliability. The program computes item reliability for each question as well as overall questionnaire reliability. See also [INDEX RELIABILITY](#).

PRINTED OUTPUT

Respondent summary. The sum, average test score, variance and standard deviation for each respondent are optionally printed (see PRINT=RSUM).

Test summary. The average test score, variance, standard deviation, Kuder-Richardson reliability coefficient and Cronbach's alpha.

Variable statistics. The average, sum, variance, standard deviation, and test.

INPUT DATA

Input data should be similar to the Likert 5-point scale (5=strongly agree, 4=moderately agree, 3=undecided, 2=moderately disagree, 1=strongly disagree)

OPTIONS

Choose ITEM ANALYSIS from the command screen and make selections.

For a Runfile use:	ITEM ANALYSIS Filter statement (optional) Job Title Keyword choices from below
--------------------	-----------------------------------------------------------------------------------------

ID=variable number

A variable number identifying each respondent when PRINT=RSUM is specified. If ID is not specified, sequential numbers are used.

PRINT=RSUM Print the respondent summary.

RECODE=n Use RECODE n, previously entered via the RECODE command.

VARS=variable numbers Use the variables specified in the list

REFERENCES

Kuder, G. F. and M. W. Richardson, *Psychometrika*, Vol. 2, No. 3, Sept. 1937.

EXAMPLES

Example 1: This example uses data randomly generated by RECODE.

Recode statements: V1=TRUNC(RAND(1)*6)
 V2=TRUNC(RAND(1)*6)
 V3=TRUNC(RAND(1)*6)
 V4=TRUNC(RAND(1)*6)
 V5=TRUNC(RAND(1)*6)

ITEM ANALYSIS -- ITEM ANALYSIS COMMAND						
TEST USING RANDOMLY GENERATED DATA						
Dataset SCF						
Transforming the data by RECODE number 1						
Respondent Summary						
ID	Sum of	Average		Standard		
Number	Test Scores	Score	Variance	Deviation		
1	11	2.200	1.700	1.30		
2	22	4.400	.800	.89		
3	16	3.200	3.200	1.78		
Total case count: 5						
2 cases deleted due to missing data						
3 cases used in the analysis						
TEST SUMMARY						
Average		Standard	K-R Test	Cronbach		
Score	Variance	Deviation	Reliability	Alpha		
16.333	20.222	4.497	.8653	.8654		
VARIABLE SUMMARY						
Average	Sum of		Standard	Test		
Response	Responses	Variance	Deviation	Reliability	Name	
V1	1.667	5	.89	.94	.832	Better or Worse
V2	3.333	10	1.56	1.25	.500	Income (000)
V3	4.000	12	2.00	1.41	.693	Children
V4	2.667	8	1.56	1.25	.945	Weight 1
V5	4.667	14	.22	.47	.803	Assets

IVEWARE -- IMPUTATION AND VARIANCE ESTIMATION

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

IVEWARE invokes the Srcware version of IVEware installed with MicroSiris that performs imputations of missing values using the Sequential Regression Imputation method.

IVEWARE also performs a variety of descriptive and model based analyses accounting for such complex design features as clustering, stratification and weighting including multiple imputation analyses for both descriptive and model-based survey statistics.

COMMAND FEATURES

There are three IVEware modules invoked by IVEWAREⁱⁱ:

IMPUTE uses a multivariate sequential regression approach to imputing item missing values. The IMPUTE command returns an imputed data set for further analysis in MicroSiris with other commands, e.g., SEARCH, MNA, MCA, REGRESSION, LOGIT_LINEAR, and the DESCRIBE and REGRESS modules.

DESCRIBE estimates the population means, proportions, subgroup differences, contrasts and linear combinations of means and proportions and variance estimates appropriate for a user specified complex sample design.

REGRESS fits linear, logistic, polytomous, Poisson, Tobit and proportional hazard regression models for data resulting from a complex sample design, using a Jackknife repeated replication to estimate the sampling variances. See also **REGRESSION** and **LOGIT_LINEAR**.

See the **IVEware User Guide** for more information. MicroSiris variable numbers are used as the short 8-character variable names in the Guide. The IVEWARE command inserts the MicroSiris variable names in the output next to the variable numbers, so you will see both in the MicroSiris output.

RESTRICTIONS

Alphabetic variables may not be used.

REFERENCES

IVEware was developed by the Survey Methodology Program at The University of Michigan's Survey Research Center, Institute for Social Research.

"A multivariate technique for multiply imputing missing values using a sequence of regression models" by Raghunathan, Lepkowski, Van Hoewyk and Solenberger (*Survey Methodology*, June 2001).

LIFE TABLE ANALYSIS

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

LIFETABLE can be used to describe a distribution of survival times. LIFETABLE uses a multiple-decrement actuarial life table approach for estimating survival, termination, and hazard rates. For each termination category LIFETABLE optionally calculates net rates, allowing for the presence of competing risks, and gross rates, which estimate the rates in the absence of competing risks. The standard errors are computed according to formulas given in Appendix X-2 of Friedman and Takeshita (1969); the Poole formulas are used.

SPECIAL TERMINOLOGY

Duration variable. A variable giving the length of time between the beginning and end of exposure, or the time of observation if no termination event has occurred.

Termination variable. A variable giving the reason for termination. Any code outside the range specified is assumed to be a continuation. The last two codes are assumed to mean "lost to follow-up" and "continuing as of the last observation."

PRINTED OUTPUT

Data matrix. (Optional: see option PRINT.) The matrix calculated from the data or recalculated from the original input matrix may be printed. Termination variable value labels, if present in the dictionary. Note that the last two code categories are for "lost to follow-up" and "continuing," and are labeled "Lost" and "Continuing" if they are not in the dictionary.

Overall totals. The overall numbers entered, withdrawn, exposed, terminated, and survived for each interval.

Overall rates. The overall survival, termination, hazard rates, and standard errors for each interval plus the cumulative survival rates, termination rates, and standard errors.

Category rates. (Optional: see option PRINT.) For each termination category the survival rates, termination rates, hazard rates, and standard errors for each interval, as well as the cumulative survival rates, termination rates, and standard errors. Termination variable value labels are used if present in the dictionary.

OPTIONS

Choose LIFE TABLE from the command screen and make selections.

For a Runfile use: LIFE TABLE
Filter statement (optional)
Job Title
Keyword choices from below

DCODE=list Duration codes to include in the analysis.

DVAR=n Duration variable.

PRINT=(MATRIX,CATE)

 MATR Print the data matrix.

 CATE Print the rates for each termination category.

RECODE=n Use RECODE n, previously entered via the RECODE command.

TCODE=list Termination codes to include in the analysis.

TVAR=n Termination variable.

WT=n Use variable n as a weight variable.

REFERENCES

Chaing, Chin Long. *The Life Table and its Applications*. Florida: Robert E. Krieger Publishing Company, 1984.

Potter, R. G. "Application of Life Table Techniques to Measurement of Contraceptive Effectiveness." *Demography*, Vol. 3, 1966, no. 2.

EXAMPLE

Calculate life/death rates from raw data.

Options: dvar=v1 dcode=1-13 tvar=V7 tcode=1-5 wt=v8 p=matrix

*** Life Table Analysis ***

Contraception

Dataset LIFETABLE\LIFE

Duration Variable is: Month

Termination variable is: Termination Category

Data are weighted by: Category Weight

63 Input entries

2 others skipped due to invalid weight

***** Tables scaled by N/(sum of weights)

DATA MATRIX

	1	2	3	4	5
	Pregnancy	Expulsion	Removal	Lost	Continued
1	.10	1.58	2.16	.09	1.09
2	.25	1.00	.97	.09	1.28
3	.24	.90	1.04	.03	.63
4	.28	.78	1.04	.01	.63
5	.32	.53	.73	.02	1.00
6	.17	.42	.61	.08	1.00
7	.27	.41	.77	.05	1.98
8	.22	.40	.63	.02	1.32
9	.14	.36	.53	.02	1.19
10	.21	.27	.49	.00	1.60
11	.14	.22	.50	.00	2.06
12	.11	.15	.48	.02	2.75
13	1.08	.82	3.22	.09	21.41

OVERALL TOTALS

	NUMBER ENTERED	NUMBER WITHDREW	NUMBER EXPOSED	NUMBER TERMINATED	NUMBER SURVIVED
1	63.0	1.2	62.4	3.8	58.0
2	58.0	1.4	57.3	2.2	54.4
3	54.4	.7	54.1	2.2	51.5
4	51.5	.6	51.2	2.1	48.8
5	48.8	1.0	48.3	1.6	46.2
6	46.2	1.1	45.7	1.2	43.9
7	43.9	2.0	42.9	1.5	40.4
8	40.4	1.3	39.8	1.2	37.9
9	37.9	1.2	37.3	1.0	35.6
10	35.6	1.6	34.8	1.0	33.1
11	33.1	2.1	32.0	.9	30.1
12	30.1	2.8	28.8	.7	26.6
13	26.6	21.5	15.9	5.1	.0

OVERALL RATES

	UNIT TERMINATION RATE	UNIT SURVIVAL RATE	STANDARD ERROR	CUMULATIVE TERMINATION RATE	CUMULATIVE SURVIVAL RATE	STANDARD ERROR	HAZARD RATE	STANDARD ERROR
1	0.0616740	0.9383260	0.0304504	0.0616740	0.9383260	0.0304504	0.0636364	0.0324190
2	0.0387257	0.9612743	0.0254913	0.0980113	0.9019887	0.0378012	0.0394903	0.0265079
3	0.0403403	0.9596598	0.0267615	0.1343978	0.8656022	0.0435734	0.0411707	0.0278746
4	0.0410845	0.9589155	0.0277312	0.1699606	0.8300394	0.0481875	0.0419462	0.0289066
5	0.0327837	0.9672163	0.0256235	0.1971723	0.8028277	0.0512312	0.0333300	0.0264846
6	0.0261628	0.9738372	0.0236221	0.2181766	0.7818235	0.0533736	0.0265096	0.0242525
7	0.0338047	0.9661953	0.0275893	0.2446059	0.7553942	0.0558987	0.0343859	0.0285461
8	0.0312239	0.9687760	0.0275766	0.2681922	0.7318078	0.0580217	0.0317191	0.0284583
9	0.0277318	0.9722682	0.0268990	0.2884866	0.7115134	0.0597485	0.0281218	0.0276607
10	0.0277664	0.9722336	0.0278409	0.3082428	0.6917572	0.0613742	0.0281573	0.0286303
11	0.0269350	0.9730650	0.0286052	0.3268753	0.6731247	0.0629140	0.0273027	0.0293915
12	0.0260468	0.9739532	0.0297021	0.3444081	0.6555920	0.0644545	0.0263905	0.0304911
13	0.3225806	0.6774194	0.1173404	0.5558893	0.4441107	0.0884549	0.3846154	0.1668108

LIST DATASET

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

Prints a set of variables from a MicroSiris dataset. You can use LIST DATASET to list temporary RECODE result variables to check their correctness.

COMMAND FEATURES

Dictionary Listing: You can list a MicroSiris dictionary with LIST DATASET without listing the data file by specifying the CASES=0 option.

Data Listing: LIST DATASET prints datasets in column format, wherein the values for a variable are printed in a column extending for as many pages as necessary and including values for all cases selected for printing. A block sketch of column format follows:

V	V	V	V
xxx	xxxx	x	xxxxxxxxxx
xxx	xxxx	x	xxxxxxxxxx

The V headings on the columns represent variable numbers and the x's represent variable values. If you request more variables than will fit within the width of a page, LIST DATASET makes several passes through the data, listing as many variables per page as it can each time. Decimal valued variables are printed with an explicit decimal point and the number of decimal places stated in the dictionary. RECODE result variables are rounded to the nearest hundredth before printing.

Note: Blanks in the listing indicates blanks or bad data values were found but no missing-data codes are defined in the dictionary.

INPUT DATA

Input is any MicroSiris dataset. If you just want to print the dictionary, set CASES=0.

OPTIONS

Choose LIST DATASET from the command screen and make selections.

CASES=(b,e,i)|0

Print a subset of the data, evaluated after application of the filter.

0: Print the dictionary only

b: Begin printing at record number b.

e: End printing at record number e.

i: Print only every i-th record between b and e, beginning with record b.
Default: i=1.

CHECK_ALPHAS

Check alphabetic variables for uniqueness if used in RECODE, which looks at only the first 8 characters. See [Alphabetic Recoding](#).

PRINT=(DICT|CODES,SINGLE|DOUBLE,ZEROS)

DICT Print the input dictionary.

CODES Print the input dictionary including variable value labels.

DOUBLE Double space between data lines.

ZEROS Print leading zeros for all variables.

Default: Don't print the dictionary, and single space data lines.

RECODE=n Use RECODE n, previously entered via the RECODE command.

VARS=(variable numbers)|ALL

Use the variables specified in the list. If V=ALL is given, all variables in the dictionary are used plus any RECODE variables.

EXAMPLE

Listing four variables. Every case is printed (the default).

*** DATASET LISTING COMMAND ***					
List all cases for four variables					
Dataset C:\MICROSIRIS\DATA\SAMPLE					
	V1	V2	V3	V5	
	Better	Income			
RECORD	or Worse	(000)	Children	Assets	
1	1	2.0	0	2.00	
2	-4	3.0	4	3.12	
3	3	37.0	5	4.43	
4	0	0.9	1	2.13	
5	4	99.0	1	1.11	
5 CASES READ					
5 CASES LISTED					

LOGIT LINEAR -- DICHOTOMOUS REGRESSION ANALYSIS

File Assignments:	DATASET	Input data
	DATAOUT	Residuals output dataset(optional)

GENERAL DESCRIPTION

Computes a Maximum Likelihood regression for a dichotomous dependent variable using either a linear or logit model. May also be used to analyze multi-way contingency tables whenever one dimension can be thought of as a dichotomous dependent variable. Both a logit and a linear analysis can be performed in the same run for comparison purposes if desired.

Logit regression yields results equivalent to logistic regression.

Note that Maximum Likelihood linear regression is not the same as Least Squares regression (produced by the REGRESSION command), which does not dichotomize the dependent variable; and coefficients are not be expected to be identical.

See IVEware [REGRESS](#) for polytomous, Poisson, Tobit and proportional hazard regression models and for data resulting from a complex sample design.

Use IVEWARE or [USTATS](#) first if you want to include interaction effects or impute for missing-data.

COMMAND FEATURES

Unlike MCA, MNA, and REGRESSION, which use least squares to fit the data to the dependent variable, LOGIT LINEAR takes into consideration that the mean of the dichotomous dependent variable is restricted and that its variance may not be constant across cases. LOGIT LINEAR can handle both interval- and nominal-scale independent variables in the same model. Two models for regression analysis are available; the logit model is the default, and a linear model may be selected with the LINEAR option.

The dependent variable is automatically made dichotomous by transforming all values differing from 1 to the value 0, and then fitting the data using the appropriate model to investigate the probability that the dependent variable is equal to 1, i.e., probability ($y=1$). Contrast IVEWARE [REGRESS](#) logistic regression which models probability ($y=0$), which is mathematically equivalent, but conceptually different.

LOGIT LINEAR does this by choosing regression coefficients which minimize the predictive error. Estimation of the regression coefficients is done on an iterative basis using a modified Newton-Raphson version of the maximum likelihood solution until the convergence criterion is less than a predefined minimum set by the CRITERION option. The number of iterations attempted is set by the MAXI option. An abnormal termination will result if convergence does not occur before the maximum number of iterations is exceeded. See *The Regression of a Dichotomous Variable* by W. DuMouchel for elaboration. See also [Bard, 1974], [Cox, 1070], [Grizzle, 1971, pp. 1057-1062], [Nerlove, 1973], and [Truett, 1967].

Nominal-scale independent variables can be transformed into dummy variables by using the CATEGORICAL option and specifying a categorical variable definition statement. All variables in the VARS list are considered model 2 independent variables. If the MODEL1 option is specified to select a subset of independent variables, regression coefficients for these variables will be estimated first; and then regression coefficients for the remainder of the model 2 variables will be estimated to test if the model 2 variables not included in the model 1 list produce a significant reduction in the predictive error.

If the LSA option is specified, LOGIT LINEAR explores the likelihood function in each direction from the maximum likelihood estimate, and compares it with normal distribution approximation. This option can add significant computing time to the command.

Categorical predictor variables.

There are times you may want to include a categorical variable in the model such as gender or education level. You cannot enter them directly because they are not continuously measured variables, but they can be represented by dummy variables. For example, if variable V32 is "Education of Head" with categories 1=0-11th Grade, 2=Completed HS, 3=Some College, 4=College Degree, 5=Graduate Degree, you can create four dummy variables with the dummy statement

V32(2-5) (See keyword CAT)

0-11th Grade will not have a dummy variable but instead is represented by the other four dummy variables all being equal to zero. For each of these, the regression compares the category in question to the base case 0-11th Grade. See Draper and Smith (1981, p. 134) for a discussion of dummy predictors.

SPECIAL TERMINOLOGY

Predictive error. Measure used to determine the precision of the regression coefficient estimates. Mathematically, one minus the geometric mean of the fitted probabilities of the sample values of the dependent variable.

PRINTED OUTPUT

Computation report. For each iteration of the estimation procedure, the following are printed:

- Iteration number
- Number of tries (evaluations of the likelihood function) before the improved estimate was accepted
- Predictive power of the improved estimate
- Convergence criterion attained by the improved estimate
- Cosine of the angle between the next correction vector and the vector of derivatives of the likelihood function
- Cosine of the angle between the next and last correction vectors

Table of coefficients. For each independent variable the following are printed:

- Regression coefficient (B)
- Standard error of the regression coefficient

Ratio of the regression coefficient to its standard error ($B/SE(B)$)

Partial fraction of predictive error explained by the predictor

Model 1 regression coefficients (see the MODEL1 option)

For categorical variables, a chi-square statistic to test whether every coefficient for each level of the variable is 0

Analysis of predictive error table. This table consists of two parts. Part one contains the overall proportion that $Y=1$, and for each model:

Number of parameters

Predictive power

Predictive power of the model taking into account the degrees of freedom used

Part two is a table of the predictive error due to regression (i.e., model 2), due to error, and the total, along with the following for each model:

Percent predictive error of the total

Chi-square to test if every regression coefficient is 0

Degrees of freedom

Sum of squares

Percent sum of squares of total

Table of adjusted probability $Y=1$ ($P(Y=1)$). The following estimates are printed:

$P(Y=1)$ for fitted values of independent variables

Non-simultaneous 95% confidence interval

Description of the distribution of independent variables for each of the following values:

for each interval scaled independent variable, the predictor mean plus and minus its standard deviation, holding all other independent variables at their mean

for categorical independent variables, each level holding all other independent variables at their mean

Goodness of fit analysis. For each independent variable, a two-way contingency table of the dependent variable and the independent variable is constructed. (Analytic independent variables are divided into four levels by the mean and the mean plus and minus its standard deviation.) The following will be printed for each level of each independent variable:

Number of cases

Observed $P(Y=1)$ (unadjusted)

Marginal predictive error (assuming cases have $P(Y=1)$ equal to observed value)

Model 2 predictive error (uses full model to compute $P(Y=1)$ for each case)

Percentage reduction of the model 2 over the marginal predictive error

Likelihood surface analysis. By moving each regression coefficient (omitting the last level of categorical independent variables) above and below the maximum likelihood estimate to where the likelihood function would be 5% if approximation with a normal distribution were accurate, the following is printed for each parameter:

Resulting parameter values

Actual values of the likelihood function at each parameter value

Comparison of LOGIT versus LINEAR

If both LOGIT and LINEAR are selected, a comparison of the predictive power of the two methods is produced.

Plot: A plot of the actual data and predicted values using Excel is optional if only one of LOGIT and LINEAR is selected. Because Excel uses no more than 255 points for a chart, a random selection of approximately 255 data points are used.

INPUT DATA

LOGIT LINEAR provides two methods for weighting data. The SWEIGHT option designates a weight variable indicating the frequency of each case within the sample. This option is generally selected to analyze a multi-way contingency table where values for the dependent variable and single independent variable identify each cell and the weight variable is the cell frequency. The PWEIGHT option designates a population weight variable that is generally inversely proportional to the probability of selection for each case. If the PWEIGHT option is selected, analysis is performed for both unweighted and weighted data.

RESTRICTIONS

An independent variable that predicts the dependent variable perfectly is not permitted in the analysis. Perfect association occurs for an analytic independent variable when the ranges of sample values for cases having $Y=1$ and $Y=0$ do not overlap, and for a categorical independent variable when a bivariate table between the dichotomous dependent variable and the levels of the predictor defined in the categorical variable definition statement produces an empty cell.

The problem can be solved for the analytic independent variable by not including it in the analysis and for the categorical independent variable by collapsing levels or filtering out cases containing the offending category.

OPTIONS

Choose LOGIT LINEAR from the command screen and make selections.

For a Runfile use:	LOGIT LINEAR Filter statement (optional) Job Title Keyword choices from below
--------------------	----------------------------------------------------------------------------------------

CAT='stmt' Stmt is a list of categorical variables and valid codes to transform into dummy predictors. For each variable every code specified is transformed into a dummy predictor, e.g., V100(5,6,1,2),V101(1-6,7). LOGIT LINEAR maintains the order of the values in creating dummy variables but renumbers the values from 1.

CRITERION=n
Numeric value to use as the convergence criterion.
Default: .01.

DELETE=(MD1,MD2)

MD1	Delete all cases where any independent variable equals its first missing-data code.
MD2	Delete all cases where any independent variable equals its second missing-data code.

DDELETE=(MD1,MD2)

MD1	Delete all cases where any dependent variable equals its first missing-data code.
MD2	Delete all cases where any dependent variable equals its second missing-data code.

DEPV=variable number The dependent variable.

LOGIT Perform a logit analysis.

LINEAR Perform a linear analysis.

LSA Explore the likelihood surface.

MAXI=n Maximum number of iterations
Default: 25.

MODEL1=(variable numbers)
 List of independent variables from VARS list to use as model 1 independent variables.

PLOT Plot predicted vs. actual. Can't be used if RECODE specified.

RECODE=n Use RECODE n, previously entered via the RECODE command.

RESIDUALS=DATASET|RECODE
 Create recode to compute residuals with predicted value variable number 10000 and residual variable number 10001.
 DATASET Write a dataset using the recode.
 RECODE Create recode only for use in subsequent commands.

SWEIGHT=Vn|PWEIGHT=Vn
 Type and number of weight variable to use (optional).
 SWEIGHT=Use Vn as sample weights.
 PWEIGHT=Use Vn as population weights.
 Analysis is performed for unweighted data also.

VARS=(variable numbers)
 The list of independent variables.

Categorical Variable Description

Example: CAT
 V100(5,6,1,2),V101(1-6,7)

EXAMPLES

Example 1: Logit regression model, deleting all missing-data from the analysis and exploring the likelihood surface.

Options: DEP V=V1 VARS=V2-V3 DEL=(MD1,MD2) DDEL=(MD1,MD2) LSA

*** MAXIMUM LIKLIHOOD DICHOTOMOUS LOGIT-LINEAR REGRESSION ANALYSS ***						
LOGIT REGRESSION WITH LIKLIHOOD SURFACE						
A LOGIT model is used to fit the data in the form:						
$\text{LOG}(P(Y=1)/(1-P(Y=1))) = B_1 + B_2 \cdot X_2 + \dots$						
Dataset DREG3						
Cases with MD1 or MD2 values for the independent variables are deleted						
Cases with MD1 or MD2 values for the dependent variables are deleted						
The dependent variable Y is V1 Better or worse						
23 Cases accepted						
1 Cases deleted due to missing data						
22 Cases used in the analysis						
ITER	TRIES	PRED POWER	CONVERGENCE	COS(C:S)	COS(C:CLAST)	
1	1	0.51128	0.00003	0.95347	0.49937	
TABLE OF COEFFICIENTS						
VAR	PARTIAL %	MODEL-2 B	STD ERROR	B/SE	MODEL-0 B	
V0	1.3	0.6616	0.8695	0.76	0.3677	Constant
V2	0.4	-0.0143	0.0336	-0.43	0.0000	Income (000)
V3	0.0	0.0081	0.3012	0.03	0.0000	Children
ANALYSIS OF PREDICTIVE POWER FOR THE REGRESSION						
PROPORTION OF Y=1 IN SAMPLE: 0.591						
	NUMBER OF PARAMETERS	PREDICTIVE POWER	"ADJUSTED" PREDICTIVE POWER			
MODEL-0	1	0.508	0.500			
MODEL-2	3	0.511	0.500			
SOURCE	PREDICTION ERROR	PERCENT OF TOTAL	CHI-SQ VALUE	DEGREES FREEDOM	SUM OF SQUARES	PERCENT OF TOTAL
DUE TO MODEL-2	0.003	0.6	0.25	2	0.64E-01	1.2
MODEL-2 ERROR	0.489	99.4	.	.	5.3	98.8
MODEL-0 ERROR(TOTAL)	0.492	100.0	.	.	5.3	100.0
TABLE OF ADJUSTED P(Y=1)						
VALUE OF X	P(Y=1)	95% CONF LIMIT	DIST. OF X MEAN	ST DEV	PROP	
X = MEAN (X)			0.592	0.381	0.773	

V2	6.1410	0.645	0.319	0.875	21.8318	Income (000)
.	37.5227	0.537	0.235	0.814	15.6909	
V3	1.0164	0.588	0.271	0.846	2.7727	Children
.	4.5291	0.595	0.278	0.849	1.7563	
GOODNESS OF FIT ANALYSIS BY PREDICTOR VARIABLE						
STRATUM	ASSOCIATED WITH	N	OBSERVED P(Y=1)	PREDICTION MARGINAL	ERROR MODEL-2	%REDUCE NAME
V2	V LO	4	0.500	0.500	0.526	-5.2 Income (000)
	LOW	5	0.800	0.394	0.436	-10.8
	HIGH	11	0.545	0.498	0.502	-0.8
	V HI	2	0.500	0.500	0.462	7.6
	OVERALL	22	0.591	0.477	0.489	-2.6
V3	V LO	7	0.714	0.450	0.460	-2.1 Children
	LOW	3	0.667	0.471	0.468	0.7
	HIGH	7	0.286	0.450	0.538	-19.6
	V HI	5	0.800	0.394	0.467	-18.7
	OVERALL	22	0.591	0.441	0.489	-10.9
LIKELIHOOD SURFACE ANALYSIS						
EACH PARAMETER IS MOVED FROM ITS ESTIMATE FOR MODEL-2 TO WHERE NORMAL THEORY PREDICTS L/LMAX=.05						
PARAM	LOW LIMIT	L/LMAX	HIGH LIMIT	L/MAX	NAME	
V0 0	-0.696	0.0474	1.439	0.0669	Constant	
V2 0	-0.082	0.0841	0.054	0.0585	Income (000)	
V3 0	-0.602	0.0651	0.618	0.0572	Children	

Example 2: Linear regression model using categorical variables.

Options: DEP V=V1 VARS=V2-V3 DEL=(MD1,MD2) DDEL=(MD1,MD2) CAT LIN
Categorical variable
statement: V3(1,4), V4(2,3)

*** MAXIMUM LIKLIHOOD DICHOTOMOUS LOGIT-LINEAR REGRESSION ANALYSIS ***						
Dataset LOGIT\DREG3						
For the independent variable, values are deleted for cases with MD1 or MD2						
For the dependent variable, values are deleted for cases with MD1 or MD2						
Using a Maximum Likelihood LINEAR model to fit the data in the form:						
PROB(Y=1) = B1 + B2*X2 + B3*X3 + ...						
INPUT DICTIONARY						
V1 Better or worse	TYPE	LOC	WID	DEC	MD1	MD2
0 Worse, 1 Better	C	1	1	0		9

V2 Income (000) C 2 3 1 99
V3 Children C 5 1 0 9
0 None, 1 One child, 2 Two children, 3 Three children, 4 Four children, 5 Five children
V4 Weight 1 C 6 1 0 9

Dependent variable Y is V1 Better or worse

22 cases accepted

1 case deleted for missing data

ITER	TRIES	PREDICTIVE POWER	CONVERGENCE	COS(C:S)	COS(C:CLAST)
1	1	0.54183	0.00205	0.51396	-0.57694

TABLE OF COEFFICIENTS

VAR	PARTIAL %	MODEL-2 B	STD ERROR	B/SE	MODEL-0 B
V0	17.5	0.6527	0.2209	2.95	0.3677 Constant
V2	0.4	-0.0031	0.0084	-0.37	0.0000 Income (000)
V3	4.9	(CHISQ=1.98 2 DF)			Children
CAT 1	0.2	0.0655	0.2576	0.25	0.0000 CODE 1 One child
CAT 2	4.5	-0.3764	0.2812	-1.34	0.0000 CODE 4 Four children
CAT 3	1.3	0.0573	0.0828	0.69	0.0000
V4	4.2	(CHISQ=1.66 2 DF)			Weight 1
CAT 1	3.9	0.1933	0.1560	1.24	0.0000 CODE 2
CAT 2	1.6	-0.1378	0.1749	-0.79	0.0000 CODE 3
CAT 3	0.5	-0.0585	0.1339	-0.44	0.0000

ANALYSIS OF PREDICTIVE POWER FOR THE REGRESSION PROPORTION OF Y=1 IN SAMPLE: 0.591

	NUMBER OF PARAMETERS	PREDICTIVE POWER	ADJUSTED PREDICTIVE POWER
MODEL-0	1	0.5084	0.5000
MODEL-2	6	0.5419	0.5000

SOURCE	PREDICTION ERROR	PERCENT OF TOTAL	CHI-SQ VALUE	DEGREES FREEDOM	SUM OF SQUARES	PERCENT OF TOTAL	P(CHI)
DUE TO MODEL-2	0.034	6.8	2.81	5	0.64	12.1	0.7
MODEL-2 ERROR	0.458	93.2			4.7	87.9	
MODEL-0 ERROR(TOTAL)	0.492	100.0			5.3	100.0	

TABLE OF ADJUSTED P(Y=1)

VALUE OF X	P(Y=1)	95% CONF	LIM	DIST OF X MEAN/ST DEV/PROP
X = MEAN(X)	0.584	0.382	0.787	
V2 6.1410	0.633	0.287	0.980	21.8318 Income (000)
37.5227	0.535	0.224	0.846	15.6909
V3 Children				
CAT 1	0.650	0.136	1.164	0.2273 CODE 1 One child
CAT 2	0.208	-0.413	0.829	0.1364 CODE 4 Four children
CAT 3	0.641	0.377	0.906	0.6364
V4 Weight 1				
CAT 1	0.777	0.483	1.072	0.3182 CODE 2
CAT 2	0.446	0.034	0.859	0.2727 CODE 3
CAT 3	0.526	0.153	0.899	0.4091

GOODNESS OF FIT ANALYSIS BY PREDICTOR VARIABLE

STRATUM		OBSERVED		PREDICTION	ERROR	%REDUCE	
ASSOCIATED	WITH	N	P(Y=1)	MARGINAL	MODEL-2		
V2	V LO	4	0.500	0.500	0.444	11.3	Income (000)
	LOW	5	0.800	0.394	0.414	-5.1	
	HIGH	11	0.545	0.498	0.459	7.8	
	V HI	2	0.500	0.500	0.572	-14.5	
	OVERALL	22	0.591	0.477	0.458	3.9	
V3	Children						
CAT 1		5	0.600	0.490	0.492	-0.5	CODE 1 One child
CAT 2		3	0.333	0.471	0.467	0.9	CODE 4 Four children
CAT 3		14	0.643	0.479	0.443	7.4	
	OVERALL	22	0.591	0.480	0.458	4.6	
V4	Weight 1						
CAT 1		7	0.714	0.450	0.457	-1.4	CODE 2
CAT 2		6	0.500	0.500	0.508	-1.6	CODE 3
CAT 3		9	0.556	0.497	0.423	14.8	
	OVERALL	22	0.591	0.483	0.458	5.2	

MANOVA -- MULTIVARIATE ANALYSIS OF VARIANCE

File Assignments:	DATASET	Input data
	MATRIX	Design matrix (optional)
	CMAT1...12	Factor contrast matrices (optional)

GENERAL DESCRIPTION

Performs a univariate or multivariate analyses of variance and covariance using a general linear hypothesis model. Up to twelve factors (independent variables) can be used. If more than one dependent variable is specified, both univariate and multivariate analyses are performed.

MANOVA performs an exact solution with either equal or unequal numbers of observations in the cells. The method of solution used by MANOVA is based on matrix operations described in Bock (1963). The method applies to any design, univariate or multivariate, complete or incomplete, with proportionate or disproportionate cell frequencies and with or without covariates.

SPECIAL USES

Multiple Univariate F-Ratios: If a multivariate analysis is performed (say on six dependent variables), a by-product of the computations is a univariate analysis. Thus, you can use MANOVA to "mass generate" F-ratios. However, MANOVA deletes cases with missing data on any variable in the analysis. Therefore, the number of cases on which an individual F-test is based may be needlessly small if there are missing data for other variables in the analysis. Use **IVEWARE** to impute for missing data to avoid this problem.

A conventional analysis of variance table will not be printed, but you can construct it with a small amount of computation from information printed--the F-ratios, their degrees of freedom, and the within-mean-square errors. (The latter appear on the diagonal of the error dispersion matrix.)

MANOVA vs. ANOVA or MCA: MANOVA is the only Microsir command for multivariate analysis of variance and the preferred command for multifactor analysis of variance. ANOVA is recommended for one-way (i.e., single factor) univariate analysis of variance. MCA, which handles multifactor univariate problems, may be required for certain problems--MCA has no limitation with respect to empty cells, and accepts more than 12 predictors. The basic analytic model of MCA is different from that of MANOVA; one important difference is that MCA is normally insensitive to interaction effects.

SPECIAL TERMINOLOGY

Covariate: Covariates are variables used to statistically adjust ("correct") the measurements on dependent variables. Covariance analysis is usually considered in the context of controlling for the effects of extrinsic sources of variation. You can also consider it as a method for studying the relationship between dependent variables and variables measured by the investigator which

cannot be incorporated in the ways of classification. Regression methods are used to partial out the effects of the covariates. See Cochran and Cox (1957, p. 52), Winer (1962, chapter 11), or Overall and Klett (1972, p. 321) and Bock and Haggard (1968, p. 103).

Factor Variable: Factor variables are the independent variables; they are used to classify cases into groups. They are sometimes called treatment variables or control variables.

Multifactor: A multifactor (or n-way) design has more than one factor variable, i.e., more than one variable is used to assign cases to groups. In a single factor (one-way) design only one variable is used to assign cases to groups.

Multivariate: A multivariate analysis has multiple dependent variables. See Bock and Haggard (1968), Bock (1966), and Overall and Klett (1972) for examples of multivariate analysis of variance design. In one of the Overall and Klett examples, a one-factor multivariate analysis of variance was used to determine whether psychiatric patients assigned to different drug treatments differed significantly in symptom characteristics. As is typical in a multivariate design, the investigators found it necessary to include more than one dependent variable (the symptom characteristics) in order to evaluate the effect of the independent variable or variables (the drug treatment).

Univariate: A univariate analysis has one independent variable.

Principal Components of the Hypothesis: For each hypothesis of interest (e.g., that there is no classroom effect), there is an appropriate partition of the total sum of products. In addition to using this hypothesis matrix in a multivariate test, MANOVA uses factor analytic techniques to determine the composite functions having maximum utility for explaining the variance in the hypotheses' effect. See Overall and Klett (1972, chapter 3) for a general discussion of principal components.

COMMAND FEATURES

Missing Data: Cases with missing-data codes on any of the input variables (dependent, covariate, or factor variables) are excluded. This may result in many excluded cases and constitutes a potential problem you should consider when planning an analysis. Unequal cell sizes are permitted, however. Use **IVEWARE** to impute for missing data to avoid this problem.

Hierarchical Regression Model: MANOVA uses a hierarchical regression approach to analysis of variance. (See Williams, 1972, p. 78; the discussion of method three in Overall and Klett, 1972, chapter 18; or Bock and Haggard, 1968, p. 107.) There is an important consequence: if there is more than one factor variable, and if there is a disproportionate number of cases in the cells formed by the cross-classification of the control variables, then you must consider the order in which the factors are defined. Presence of disproportional subclass numbers confounds the main effects, and you must choose the order in which the confounded effects should be eliminated. This choice (unless special ordering is invoked) is accomplished by the order in which you define the factors. When using standard ordering, the first listed effect will be tested with all other main effects eliminated. (The general rule is that each test eliminates effects listed afterward.) For a standard two-way analysis, the interaction term is not affected by the order of the factors; more generally, for a standard n-way analysis, only the nth-order interaction term is unaffected. This problem exists for both univariate and multivariate analyses.

Design Matrix and Contrast Matrices: To use regression methods for the analysis of variance, there must be a design matrix specifying treatment classifications and interaction. If requested, this design matrix is constructed by MANOVA using Kronecker products involving specified one-way contrast matrices. (See contrast options in the next paragraph.) Alternatively, you may supply the design matrix. For a general discussion of contrasts, see Cochran and Cox (1957, chapter 5); Tables 5.1 and 5.2 in the cited article are examples of design matrices. Also see Kempthorne (1952, p. 236).

Contrast Options: There are two fixed and two flexible options for setting up contrasts. Nominal contrasts can be generated by MANOVA and are most commonly selected. They are the customary deviations of row and column means from the grand mean and the generalizations of these for the interaction contrasts. These are typically used when you only need the overall tests of significance. MANOVA can also generate Helmert contrasts; they are discussed in Bock (1963, p. 103). If, for a given factor, you don't want to use either of the contrast matrices built into MANOVA, you can supply a one-way contrast matrix. Use this option when you want single degrees of freedom contrasts, or when you are sub-grouping degrees of freedom. The fourth contrast option is to bypass contrast matrices and supply a design matrix instead.

Standard N-Way Univariate Analyses: MANOVA is the only MicroSiris command that performs traditional multifactor univariate analysis of variance (i.e., ordinary two-way or three-way ANOVA). For a univariate problem the printout includes the familiar ANOVA table.

Augmentation of Within-Cells Sum of Squares. You can augment the within-cells (error) sum of squares from the orthogonal estimates. Thus you can use MANOVA for Latin squares and for pooling of interaction terms with error. (See option ERROR.) Latin squares are discussed in Winer (1962, chapter 10), where a discussion of pooling begins on page 202.

Reordering and/or Pooling Orthogonal Estimates: A conventional ordering of orthogonal estimates is built into MANOVA. However, orthogonal estimates may be rearranged into some other order. (See the REORDER option.) Further, it is possible to pool several orthogonal estimates, such as several interaction terms, for simultaneous testing, or to partition the cluster of orthogonal estimates for a given effect into smaller clusters for separate testing. (See the DEGFR option.)

PRINTED OUTPUT

Cell Means and Ns: (Optional: see option PRINT=MEANS.) For each cell, MANOVA prints N and the mean for each dependent variable and covariate. The means are not adjusted for any covariates. Cells are labeled consecutively starting with "1 1" (for a two-factor design) regardless of actual codes of factor variables. In indexing the cells, the indices of the last factors are the minor indices (fastest moving).

Basis of Design: (Optional: see option PRINT=BASIS.) This is the design matrix MANOVA generated or you supplied, depending on the option chosen. The effects equations are in columns, beginning with the main effect in column one. If REORDER was specified, the matrix is printed after reordering.

Inter-correlations among the Normal Equations Coefficients: (Optional: see option PRINT=CORR.)

The Error Correlation Matrix: In a multivariate analysis of variance, the error term is a variance-covariance matrix. This is that error term (before adjustment for covariates, if any) reduced to a correlation matrix.

Principal Components of the Error Correlation Matrix: The components are in columns. These are the components of the error term (before adjustment for any covariates) of the analysis.

Error Dispersion Matrix and Standard Errors of Estimation: This is the error term, a variance-covariance matrix, for the analysis. The matrix is adjusted for covariates, if any. Each diagonal element of the matrix is exactly what would appear in a conventional analysis of variance table as the within mean square error for the variable. Degrees of freedom for the matrix are printed, adjusted for augmentation if that was requested. Standard errors of estimation correspond to the standard deviations in Bock and Haggard (1968, Table 13).

If Covariance Analysis is requested:

Adjusted error dispersion matrix reduced to correlations. This is the error term--a variance-covariance matrix--reduced to a correlation matrix after adjustments for covariates.

Statement of regression analysis.

Principal components of the error correlation matrix. The components are in columns. These are the components of the error term of the analysis after adjustment for covariates.

For Univariate Analyses:

ANOVA table: Degrees of freedom, sum of squares, mean squares, F-ratios, total sums of squares and probability of chance occurrence of the F-ratios are printed.

For Multivariate Analyses:

The items listed below are printed for each effect. Adjustments are made for covariates, if any. The order of effects in the multivariate printout is exactly the reverse of that indicated by the Names statement; that is, for the conventional analysis, the test for the highest order interaction is printed first and the test for the grand mean is printed last. (The test of the grand mean is not usually interpreted; it is a test that the grand mean is zero.)

F-ratio for likelihood ratio criterion: Rao's approximation is used. See Rao (1965, p. 472). This is a multivariate test of significance of the overall effect for all the dependent variables simultaneously. The probability of chance occurrence of this F-ratio is also printed.

Canonical variances of the principal components of the hypothesis: These are the roots, or eigenvalues, of the hypothesis matrix.

Coefficients of the principal components of the hypothesis: These are the correlations between the variables and the components of the hypothesis matrix. The number of nonzero components for any effect will be the minimum of the degrees of freedom and the number of dependent variables. The components characterize the differences between the levels represented by the hypothesis.

Contrast component scores for estimated effects: These are the scores of the hypothesis for the contrasts used in the design. They are analogous to the column means in a univariate analysis of variance and can be used in the same manner to locate variables and contrasts that give unusual departures from the null hypothesis.

Cumulative Bartlett's tests on the roots: See Rao (1965, p. 474). This is an approximate test for remaining roots after eliminating the first, second, third, etc.

F-ratios for univariate tests: Exactly the F-ratios which would be obtained in a conventional univariate analysis. The probabilities of chance occurrence of these F-ratios are also printed.

INPUT DATA

The dependent variable(s) should be measured on an interval scale. In order to generate results that allow inferences to a specified population, you must assume that the dependent vector variable is multivariate normal in distribution with the same variance-covariance matrix for each group. (This latter assumption is the multivariate analogue of the assumption of homogeneity of variances.)

If more than one dependent variable is used, they may or may not be related. Normally, in a true multivariate analysis, they will be related. However, if you use MANOVA to mass generate F-ratios for the dependent variables, weight, spelling score, and interest in baseball (for example) could be obtained in a single analysis.

In the dataset, there must exist factor (grouping) variables, coded 1 through n, which can be used to designate the proper cell for the case. For example: in a two-factor design, sex (coded 1 and 2) could be one grouping variable and classroom (coded 1, 2, 3, and 4) could be a second. Use RECODE to generate these variables if necessary.

RESTRICTIONS

The maximum number of factors is 12.

If a data value is missing for any variable in the analysis, MANOVA deletes the whole case.

All cells in the design must have one or more entries.

OPTIONS

Choose MANOVA from the command screen and make selections.

For a Runfile use:	MANOVA Filter statement (optional) Job Title Keyword choices from below
--------------------	----------------------------------------------------------------------------------

RECODE=n Use RECODE n, previously entered via the RECODE command.

VARs=Dependent variable numbers
 Use the variables specified in the list.

COVARs=list The list of variables to treat as covariates.
 Default: No covariate analysis will be done.

DMATRIX=n Use matrix n as the design matrix. The matrix must have as its basis the rows (the way design matrices are usually presented in texts; see Cochran and Cox, 1957, p. 154.) The number of rows and columns required is calculated from the number of levels for each factor. Thus for a 3x2 factored

design in which the interaction effect will be partitioned to test a particular hypothesis, the design matrix would be:

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	0.0	0.0	1.0	1.0
-1.0	-1.0	1.0	1.0	0.0	0.0
1.0	-1.0	0.0	0.0	-1.0	1.0
0.5	-0.5	-1.0	1.0	0.5	-0.5

You must enter the design matrix with the MATRIX command prior to issuing the MANOVA command.

Default: MANOVA constructs the design matrix according to the contrast specifications on the Factors.

ERROR=(WSS|AUGMENT=(list))

WSS Use within-sum of squares as the error term.

AUGMENT=(list)

Augment within-sum of squares by the columns of the orthogonal estimates matrix, indicated by list.

Default: ERROR=WSS

PRINT=(BASIS,MEANS)

BASIS Print the basis of design matrix.

MEANS Print the cell means and n's.

REORDER=(list)

Reorder the orthogonal estimates according to list (see special ordering section below).

Factors

Up to 12 Factors may be supplied, one factor per statement.

FACTOR=(Vn,m)

Defines FACTOR Vn with codes 1 through m. Thus FACTOR=(V1,3) defines a factor with three levels, one for V1=1, another for V1=2, and a third for V1=3. Cases with values outside the range 1-m are discarded.

CONTRAST=(NOMINAL|HELMRT|CMAT=n)

Designates what type of contrasts to use.

NOMINAL Effect mean deviated from the grand mean, e.g., M(1)-GM, M(2)-GM, etc. These are the customary deviations from the grand mean and are ordinarily used when there is no subdivision of the degrees of freedom for effects.

HELMRT Helmert contrasts. Mean of effect 1 deviated from the sum of means 1 through r when r levels are involved.

CMAT=k Use matrix k assigned to CMATi, where i is the ordinal factor number. The matrix is an n x (n + 1) matrix, where n is the

number of levels of the factor. The "1 1" entry must be 1.0, and the rest of the first column must contain zeros. The matrix for a three-level factor with deviation contrasts would be:

1	.3333	.3333	.3333
0	.6667	.3333	.3333
0	-.3333	.6667	-.3333

Default: CONTRAST=NOMINAL

Names

These identify the tests performed. Ordinarily there will be a test statement for the grand mean, followed by a test statement for each main effect, and finally, one for each possible interaction. (Note: Test results are displayed in reverse order)

NAME='test name' A 1- to 16-character name for the test.

DEGFR=n The natural grouping of degrees of freedom (or hypothesis parameter equations) occurs when the conventional order of statistical tests is used. (Conventional order: the order in which sums of squares are typically discussed; described in the section Special Ordering.) DEGFR is used only to change the grouping, e.g., to pool several interaction terms and test them simultaneously or to partition the degrees of freedom of some effect into two or more parts. When using the DEGFR option, be sure to use it on all test statements, including a degree of freedom for the grand mean.
Default: Use the natural grouping of degrees of freedom.

For ordinary factorial analysis, the required test statements are:

Grand mean: e.g., use GRAND MEAN.

Main effects: One for each factor. Supply these main effects statements in factor order.

Interactions: the two-way interactions, followed by the three-way interactions, etc. Within each level of interaction, supply the statements in forward order.

Special ordering

The conventional ordering of orthogonal estimates of effects (e.g., mean, C, B, A, BxC, AxC, AxB, AxBxC, for a three-factor design) may be rearranged into some other order (e.g., mean, A, B, AxB, C, AxC, BxC, AxBxC). Furthermore, it is possible to break down each cluster of orthogonal estimates into smaller clusters and arrange them into some unusual order. If special ordering is used for any reason, take care that the order for the degrees of freedom agrees with the design as it will finally appear (use the DEGFR option on the name statement if necessary). Likewise, names should agree with the new design.

To specify a special ordering, locate the column number of the orthogonal estimates in the conventional order, then determine in what order these columns should finally appear.

Example: A has 3 degrees of freedom, B has 1 degree of freedom.

<u>Conventional order</u>	<u>Orthog. Estimate</u>	<u>New Order</u>
1	GRAND MEAN	1
2	B(1)	8
3	A(1)	7
4	A(2)	2
5	A(3)	3
6	A(1)xB(1)	6
7	A(2)xB(1)	4
8	A(3)xB(1)	5

Then REORDER=(1,8,7,2,3,6,4,5)

REFERENCES

Bock, R. D. "Programming Univariate and Multivariate Analysis of Variance." *Technometrics*, 1963, pp. 5, 95-117. (MANOVA uses the method of analysis outlined here.)

Bock, R. D. "Contributions of Multivariate Experimental Designs to Educational Research." In *Handbook of Multivariate Experimental Psychology*. Edited by R. B. Cattell. Chicago: Rand McNally, 1966, pp. 820-40. (Worked examples of three applications of multivariate analysis of variance.)

Bock, R. D. and E. A. Haggard. "The Use of Multivariate Analysis of Variance in Behavioral Research." In *Handbook of Measurement and Assessment in Behavioral Sciences*. Edited by D. K. Whitla. Reading, Mass: Addison-Wesley, 1968, pp. 100-42.

Cochran, W. G. and G. Cox. *Experimental Design*. Second ed. New York: Wiley, 1957. Corrected printing, 1968. (Univariate only.)

Overall, J. E. and C. J. Klett. *Applied Multivariate Analysis*. New York: McGraw-Hill, 1972. Reprinted: Krieger, 1983. pp. 321, 430, 441-68.

Rao, C. R. *Linear Statistical Inference and its Applications*. Second ed. New York: Wiley, 1973.

Winer, B. J. *Statistical Principles in Experimental Design*. Second ed. New York: McGraw-Hill, 1971. (Univariate.)

EXAMPLES

Example 1: Two factors, multivariate; variable 1 is the sex factor variable and variable 2 is the classroom factor variable.

Options: VARS=v3-v8
 Factors: FACTOR=(V1,2) CONTRAST=NOMINAL
 FACTOR=(V2,2) CONTRAST=NOMINAL
 Names: NAME='Grand Mean'
 NAME=Classroom
 NAME=Sex
 NAME='Sex X Classroom'

*** MULTIVARIATE ANALYSIS OF VARIANCE ***

Two-factor example; 5 variates, no covariates

Dataset CRAMER

Number of covariates: 0

Number of factors: 2

The error term is the within-cells sum of squares

Number of tests: 4

Factor 1 has 2 levels and nominal contrasts

Factor 2 has 2 levels and nominal contrasts

Error Correlation Matrix

		V3 D1	V4 D2	V5 D3	V6 D4	V7 D5	V8 D6
D1	V3	1.0000	-0.0631	-0.1710	0.0799	0.2951	-0.0972
D2	V4	-0.0631	1.0000	0.6330	-0.1219	-0.1217	-0.0583
D3	V5	-0.1710	0.6330	1.0000	0.0208	0.0759	-0.2612
D4	V6	0.0799	-0.1219	0.0208	1.0000	0.0742	-0.1165
D5	V7	0.2951	-0.1217	0.0759	0.0742	1.0000	-0.4402
D6	V8	-0.0972	-0.0583	-0.2612	-0.1165	-0.4402	1.0000

The Principal Components

		First	Second	Third	Fourth	Fifth	Sixth
D1	V3	-0.1540	0.5768	-0.3564	0.6924	-0.1505	0.1203
D2	V4	0.8055	-0.3127	-0.1253	0.3411	-0.0157	-0.3481
D3	V5	0.9069	-0.0928	0.0966	0.0487	0.1435	0.3698
D4	V6	-0.0342	0.3443	0.8849	0.2998	0.0550	-0.0654
D5	V7	0.1704	0.8094	-0.1989	-0.1886	0.4764	-0.1177
D6	V8	-0.4609	-0.6481	-0.0489	0.3685	0.4772	0.0403

		First	Second	Third	Fourth	Fifth	Sixth
Roots		1.7375	1.6327	0.9771	0.8594	0.5012	0.2921

The components of Wilks and Bartlett's tests are orthogonal rotations of the above

Error Dispersion Matrix

		V3 D1	V4 D2	V5 D3	V6 D4	V7 D5	V8 D6
D1	V3	2.20621E+04	-2.13381E+02	-6.65711E+02	1.30771E+03	3.41111E+01	-1.17528E+02
D2	V4	-2.13381E+02	5.18097E+02	3.77614E+02	-3.05628E+02	-2.15556E+00	-1.08028E+01
D3	V5	-6.65711E+02	3.77614E+02	6.86839E+02	6.01528E+01	1.54722E+00	-5.57333E+01
D4	V6	1.30771E+03	-3.05628E+02	6.01528E+01	1.21400E+04	6.35833E+00	-1.04544E+02
D5	V7	3.41111E+01	-2.15556E+00	1.54722E+00	6.35833E+00	6.05556E-01	-2.78889E+00
D6	V8	-1.17528E+02	-1.08028E+01	-5.57333E+01	-1.04544E+02	-2.78889E+00	6.62778E+01

The error dispersion matrix has 36 degrees of freedom.

Standard Errors

3 4 5 6 7 8

D1	D2	D3	D4	D5	D6
1.48533E+02	2.27617E+01	2.62076E+01	1.10182E+02	7.78175E-01	8.14112E+00

STATISTICS FOR TEST 4 SEX x CLASSROOM
 Variates 6

Hypothesis Equations 1

Principal Components 1

F-Ratio for the likelihood ratio criterion: 0.557
 having 6 and 31 degrees of freedom with probability 0.76

The Canonical Variances of the Principal Components of the Hypothesis

1	3.8800
---	--------

The Coefficients of the Principal Components of the Hypothesis

	First
V3	0.0249
V4	0.2645
V5	0.4962
V6	-0.0714
V7	0.8252
V8	-0.6902

Contrast Component Scores for Estimated Effects

	First
1	0.3114

For main effects and nominal contrasts, the score eliminated by taking degrees of freedom

	First
2	-0.3114

Cumulative Bartlett's Tests on the Roots

Roots	Chi-Square	Deg. of Fr.	Probability
1 through 1	3.3778	6	0.76

For univariate tests, F-Ratios for the adjusted variates

	V3	V4	V5	V6	V7	V8
	D1	D2	D3	D4	D5	D6
F-Ratio	0.002	0.271	0.955	0.020	2.642	1.848
P(F)	0.960	0.612	0.337	0.884	0.109	0.179

These ratios have 1 and 36 degrees of freedom.

STATISTICS FOR TEST 3 SEX

Variates 6

Hypothesis Equations 1

Principal Components 1

F-Ratio for the likelihood ratio criterion: 3.286
 having 6 and 31 degrees of freedom with probability 0.01

The Canonical Variances of the Principal Components of the Hypothesis

1 22.8986

The Coefficients of the Principal Components of the Hypothesis

First
V3 0.0636
V4 0.4457
V5 0.4488
V6 0.0540
V7 0.6794
V8 0.0406

Contrast Component Scores for Estimated Effects

First
1 0.7566

For main effects and nominal contrasts, the score eliminated by taking degrees of freedom

First
2 -0.7566

Cumulative Bartlett's Tests on the Roots

Roots	Chi-Square	Deg. of Fr.	Probability
1 through 1	16.2458	6	0.01

For univariate tests, F-Ratios for the adjusted variates

	V3 D1	V4 D2	V5 D3	V6 D4	V7 D5	V8 D6
F-Ratio	0.093	4.548	4.613	0.067	10.569	0.038
P(F)	0.760	0.038	0.036	0.793	0.003	0.841

STATISTICS FOR TEST 2 CLASSROOM

Variates 6

Hypothesis Equations 1

Principal Components 1

F-Ratio for the likelihood ratio criterion: 0.726
having 6 and 31 degrees of freedom with probability 0.63

The Canonical Variances of the Principal Components of the Hypothesis

1 5.0570

The Coefficients of the Principal Components of the Hypothesis

First
V3 0.2263
V4 0.5406
V5 0.7888
V6 0.0638
V7 -0.1807
V8 0.1555

Contrast Component Scores for Estimated Effects

First
1 0.3556

For main effects and nominal contrasts, the score eliminated by taking degrees of freedom

First
2 -0.3556

Cumulative Bartlett's Tests on the Roots

Roots	Chi-Square	Deg. of Fr.	Probability
1 through 1	4.3376	6	0.63

For univariate tests, F-Ratios for the adjusted variates

	V3	V4	V5	V6	V7	V8
	D1	D2	D3	D4	D5	D6
F-Ratio	0.259	1.478	3.146	0.021	0.165	0.122
P(F)	0.620	0.230	0.081	0.882	0.689	0.729

These ratios have 1 and 36 degrees of freedom.

STATISTICS FOR TEST 4 GRAND MEAN

Variates 6

Hypothesis Equations 1

Principal Components 1

F-Ratio for the likelihood ratio criterion: 570.835
having 6 and 31 degrees of freedom with probability 0.00

The Canonical Variances of the Principal Components of the Hypothesis

1 3,977.4280

The Coefficients of the Principal Components of the Hypothesis

	First
V3	0.4489
V4	0.4471
V5	0.2732
V6	0.4457
V7	0.4446
V8	0.1275

Contrast Component Scores for Estimated Effects

First
1 9.9717

For main effects and nominal contrasts, the score eliminated by taking degrees of freedom

First
2 -9.9717

Cumulative Bartlett's Tests on the Roots

Roots	Chi-Square	Deg. of Fr.	Probability
1 through 1	155.5581	6	0.00

For univariate tests, F-Ratios for the adjusted variates

	V3	V4	V5	V6	V7	V8
	D1	D2	D3	D4	D5	D6
F-Ratio	801.419	795.000	296.894	789.972	786.220	64.651
P(F)	0.000	0.000	0.000	0.000	0.000	0.000

These ratios have 1 and 36 degrees of freedom.

Example 2: Two factors, univariate and using RECODE. Variable 7 is sex, variable 8 is classroom. The main effects and interaction effects are in reverse order.

Recode statements: IF MDATA(V7,V8) THEN REJECT
 IF V7 GT 1 THEN V7=2
 IF V8 GT 3 THEN V8=2 ELSE V8=1

Options: recode=1 v=4
Factors: factor=(v7,2)
 factor=(v8,2) contrast=nominal
Names: name='GRAND MEAN'
 name=SEX
 name=CLASSROOM
 name='CLASSROOM X SEX'
 END

```
*** MULTIVARIATE ANALYSIS OF VARIANCE ***  
  
UNIVARIATE TEST WITH MORE PRINT OPTIONS  
  
Dataset OIVT  
  
Transforming the data by RECODE 1 read from MANOVA.RUN  
  
Number of covariates: 0  
  
Number of factors: 2  
  
The error term is the within-cells sum of squares  
  
The number of tests is 4  
  
Factor 1 has 2 levels and nominal contrasts  
  
Factor 2 has 2 levels and nominal contrasts  
  
Intercorrelations among normal equations coefficients  
  
      1      2      3      4  
1  1.0000  
2 -0.1015  1.0000  
3  0.4338 -0.1631  1.0000  
4 -0.1631  0.4338 -0.1015  1.0000  
  
      DF  Sum Squares  Mean Squares  F-Ratio  Prob.  
ANOVA Error      321      135.114      0.421  
EDUC X MARITAL    1       0.000      0.000      0.000      0.982
```


EDUCATION	1	50.358	50.358	119.639	0.000
MARITAL STATUS	1	0.479	0.479	1.137	0.287
GRAND MEAN	1	1284.049	1284.049	3050.604	0.000

MATRANS -- MATRIX TRANSFORMATIONS

File Assignments:	MATIN	Input matrix
	MATOUT	Output matrix (conditional)

GENERAL DESCRIPTION

Creates a new matrix from an existing MicrOsiris matrix, prints a matrix, or subsets a matrix. It can reverse the signs of matrix elements, invert a symmetrical matrix, create dissimilarities from correlations and similarities, and make a rectangular matrix from a symmetrical one.

OPTIONS

MATRIX=*n* *n* is the number of the input matrix produced by CORRELATIONS or another MicrOsiris command or assigned to MATIN.

Default: MATRIX=1.

ROWV=*variable numbers*|ALL

Output row variables.

Default: ALL.

COLV=*variable numbers*

Output column variables.

Default: Input column variable numbers for a rectangular matrix, and the ROWV variables for a symmetrical matrix. Do not specify COLV for a symmetrical matrix.

MATOUT=*n* Output matrix number:

ABSOLUTE Take the absolute value of each matrix element.

DISS=F1|F2|F3|F4

Convert correlations(*r*) or similarities(*s*) to dissimilarities by:

F1: $D = 1 - r/2$ (strong negative *r* considered very dissimilar; best option).

F2: $D = 1 - \text{abs}(r)$ (strong negative *r* assigned small dissimilarity; ok)

F3: $D = 1 - s$

F4: $D = \text{sqrt}(1 - s)$

POSITIVE Flip rows and columns signs to maximize the number of positive signs. Negative values are sometimes a result of scoring conventions rather than an indication of an intrinsically negative relationship between variables. POSITIVE removes the effect of these arbitrary measurement conventions by reducing the number of negative signs in the matrix as much as possible by switching the signs of the matrix elements one row and column at a time. Negative measurements

remaining after this process represent intrinsically negative relationships that cannot be removed by changes in the scoring conventions for a single variable.

- N=n The number of cases represented in the matrix.
Default: Use the value stored in the input matrix. Ignored for [CSV](#) files.
- REVERSE Reverse the sign of each matrix element, except diagonal of symmetric matrix.
- RECTANGULAR Create a rectangular matrix from a symmetrical matrix.
- SQUARE Square each matrix element.
- PRINT=(MATIN,OUTM)
 MATIN: Print input matrix.
 OUTM: Print output matrix.
- TITLE='string' A 1- to-100 character title for the matrix.

EXAMPLE

Print MicroSiris matrix 1 stored in the file CORRELATIONS after reversing the off-diagonal entries, showing both input and output matrices.

```

*** MATRIX TRANSFORMATIONS ***

Using matrix 1 from file CORRELATIONS.MTX

Creating matrix number 5 in file -MTX1.MTX

Reversing the signs of the input matrix

Input matrix CORRELATIONS

```

	V1	V2	V3
	Better or Worse	Income (000)	Children
Better or Worse V1	0.100000E+01		
Income (000) V2	0.702950E+00	0.100000E+01	
Children V3	-0.275411E+00	0.723788E+00	0.100000E+01

```


```

	Means	Standard Deviations
Better or Worse V1	0.100000E+01	0.355903E+01
Income (000) V2	0.107250E+02	0.175376E+02
Children V3	0.220000E+01	0.216795E+01

```

Output matrix

```

	V1	V2	V3
	Better or Worse	Income (000)	Children
Better or Worse V1	0.100000E+01		
Income (000) V2	-0.702950E+00	0.100000E+01	
Children V3	0.275411E+00	-0.723788E+00	0.100000E+01

```


```

	Means	Standard Deviations
Better or Worse V1	0.100000E+01	0.355903E+01

Income (000)	V2	0.107250E+02	0.175376E+02
Children	V3	0.220000E+01	0.216795E+01

MCA -- MULTIPLE CLASSIFICATION ANALYSIS

File Assignments:	DATASET	Input data
	DATAOUT	Residuals output dataset(optional)

GENERAL DESCRIPTION

MCA examines the relationships between several categorical independent variables and a single dependent variable using an additive model. The technique handles predictors with no better than nominal measurement and interrelationships of any form among predictors or between a predictor and the dependent variable. The dependent variable should be an interval-scaled variable without extreme skewness or a dichotomous variable with two frequencies which are not extremely unequal. MCA determines the effects of each predictor before and after adjustment for its inter-correlations with other predictors in the analysis. It also provides information about the bivariate and multivariate relationships between the predictors and the dependent variable. See Andrews, et al., *Multiple Classification Analysis*, for a complete description of the methodology used.

COMMAND FEATURES

Missing Data: Cases with missing data on the independent variables may be eliminated (see DELETE option). Cases with missing data on the dependent variable are automatically excluded from the analysis.

Analysis of Variance: If only one independent variable is specified MCA performs a one-way analysis of variance. Unlike **MANOVA**, MCA has no limitation with respect to empty cells, and accepts more than 12 predictors. The basic analytic model of MCA is different from that of MANOVA; one important difference is that MCA is normally insensitive to interaction effects.

PRINTED OUTPUT

Dependent Variable Statistics: For the dependent variable (Y):

- Grand mean
- Standard deviation (square root of unbiased estimator of the population variance.)
- Sum of Y
- Sum of Y-squared
- Total sum of squares
- Explained sum of squares
- Residual sum of squares
- Number of cases used in the analysis
- The sum of weights

Independent Variable Category Statistics: For each category of an independent variable:

The number of cases (raw, weighted, and percentages)

Mean and standard deviation
Deviation of the category mean (unadjusted and adjusted)
Adjusted class mean MCA coefficient
Eta and eta squared
Partial beta and beta-squared coefficients
Unadjusted and adjusted sum of squares
Bivariate frequency tables for every pair of predictors (optional)

One-Way Analysis of Variance Summary Statistics: If only one independent variable is specified, the following are printed:

Eta squared
Adjustment factor
Adjusted eta and eta squared
Total sum of squares
Between-mean sum of squares
Within-groups sum of squares
F value (degrees of freedom are printed)

Plot(optional): A plot of the actual data and predicted values using Excel. Because Excel uses up to 255 points for a chart, a random selection of approximately 255 data points is used.

Interpretation of Results

(From Multiple Classification Analysis, Andrews, Morgan, et al, 1973)

The major interpretation in MCA is of the adjusted and unadjusted coefficients printed out for each subclass. In a population where there was no correlation among the predictors, the observations in one class of characteristic A would be distributed over all classes of the other characteristics in a fashion identical to the way in which those in other classes of A were distributed. Hence, the unadjusted mean Y for each subclass of A would be an unbiased estimate of the effect of belonging to that class of characteristic A. In the real world, however, characteristics are correlated. Young people are more likely in lower income groups, and in higher education groups than are older people. The multivariate process is essentially one of adjusting for these "non-orthogonalities." The adjusted means are estimates of what the mean would have been if the group had been exactly like the total population in its distribution over all the other predictor classifications. It is useful not only to have the "pure" effects of each class adjusted for all the other characteristics, but also to see how these adjusted effects differ from the unadjusted effects.

Both the adjusted and unadjusted coefficients are expressed by the program as deviations from the overall mean, and are constrained so that their sum, weighted by the proportion in each subclass, is zero.

The adjusted coefficients for any predictor may be considered an estimate of the effect of that predictor alone "holding constant" all other predictors in the analysis. Differences between the adjusted and unadjusted coefficients can be analyzed, and explanations for these differences may often be found in the two-way tables of predictors. It is often valuable to compare the coefficients within a predictor to see whether there is a pattern or, possibly, a lack of pattern which is of theoretical interest.

The coefficients for the predictors do not provide definitive information about logical priorities, chains of causation, or about interaction effects. It is possible for the program to assign considerable explanatory power to a variable late in a causal chain, such as an attitude, when much of the credit "really" belongs to a logically prior, but not as powerful variable, such as race.

Interaction effects of two or more predictors on the dependent variable will not be revealed by the program, since the assumption is that the effects of all the predictors are additive, i.e. the effect for predictor A is assumed to be the same for one class of predictor B as it is for every other class.

When certain categories are closely overlapped, it is often particularly enlightening to reverse the order of these two predictors. Since the iterative process makes successive approximations to each predictor on the basis of the latest estimates of other predictors, changing the order of predictors results in an entirely new solution as far as the program is concerned.

A difficulty in using the adjusted coefficients as a presentational device is that the additivity assumptions may lead to absurd adjusted means for some groups (less than zero, for instance) if the assumption is inappropriate for the data being analyzed. This is particularly likely when the dependent variable is a dichotomy, such as home ownership. Clearly, it is not sensible to predict that less than 0 percent of a subgroup own a home.

Presentation of Results

It is most informative to the reader to present first the η s and β s, measures of the relative importance of each predictor singly and in competition with the others, and then to present the unadjusted and adjusted sub-group averages, together with a detailed description of what the subclasses represent and with the number of cases in each. (The number of cases should be included because it is an indicator of the potential variability of the estimates.) Multiple R^2 unadjusted and multiple R^2 adjusted are also usually reported.

We recommend that the results be given in the form of unadjusted and adjusted subgroup averages rather than in the form of deviations because the user finds it easier to scan unadjusted and adjusted subgroup averages than positive and negative deviations. However, the adjusted deviations can be included for convenience in seeing the net effects of each predictor. As noted above, a complication of subgroup averages is that occasionally the expected value is impossible (e.g. negative although the dependent variable is a variable with no negative values); if impossible expected values are presented, a short explanatory note should be included.

Examples of presentation of MCA results can be found in Barfield and Morgan (1969), Blumenthal, Kahn, Andrews and Head (1972), Johnston and Bachman (1972), Johnston (1973), Katona, Strumpel and Zahn (1971), Morgan, David, Cohen and Brazes' (1962), Mueller (1969), and Pelz and Andrews (1966).

INPUT DATA

The dependent variables must be measured on an interval scale or must be a dichotomy. Predictor variables must be categorical, preferably with six or fewer categories.

RESTRICTIONS

Predictor codes must be in the range 0 - 999.

OPTIONS

Choose MCA from the command screen and make selections.

For a Runfile use: MCA
 Filter statement (optional)
 Job Title
 Keyword choices from below

DEPV=variable number The dependent variable.

VAR=List of independent variables.

One-way analysis of variance is performed if only one variable is specified.

DELETE=(MD1,MD2)

MD1 Delete all cases where any independent variable equals its first missing-data code.

MD2 Delete all cases where any independent variable equals its second missing-data code.

TEST=%MEAN|CUTOFF|%RATIO.

Convergence test desired. If not specified, MCA iterates until the maximum number of iterations (MAXI) is exceeded. (see CRITERION)

%MEAN Test if the change in all coefficients after each iteration is below a specified fraction of the grand mean.

CUTOFF Test if the change in all coefficients after each iteration is less than a specified value.

%RATIO Test if the change is less than a specified fraction of the ratio of the standard deviation of the dependent variable to its mean.

CRITERION=n

Tolerance (0.0-1.0) of the convergence test selected.

Default: CRITERION=.005.

MAXI=n The maximum number of iterations.

Default: 25 iterations.

PLOT Plot predicted vs. actual.

PRINT=(TABLES,TRACE)

TABLES Print pair-wise cross-tabulations of independent variables.

TRACE Print the coefficients from all iterations.

RECODE=n Use RECODE n, previously entered via the RECODE command.

RESIDUALS=DATASET|RECODE

Create recode to compute residuals with predicted value variable number 10000 and residual variable number 10001.

DATASET Write a dataset using the recode.
RECODE Create recode only for use in subsequent commands.

WT=n Use variable n as a weight variable

REFERENCES

Andrews, F. M., J. N. Morgan, J. A. Sonquist and L. Klem. *Multiple Classification Analysis*. Second edition. Ann Arbor: Institute for Social Research, The University of Michigan, 1973.

EXAMPLE

Predicting income (V268) from occupation, marital status, and education.

```

*** MULTIPLE CLASSIFICATION ANALYSIS ***

PREDICTING INCOME

DATASET scf

For the independent variables, values are deleted with MD1 or MD2

The iteration maximum is 25

The convergence test is %MEAN

The tolerance factor is .00500

    299 cases were used in the analysis.

RESULTS BASED ON ITERATION      6

DEPENDENT VARIABLE (Y) =   V268   TOTAL FAMILY INC

MEAN                                10528.32
STANDARD DEVIATION                 7553.407

SUM OF Y                           3147968.
SUM OF Y SQUARE                    .5014490E+11

TOTAL SUM OF SQUARES               .1700208E+11
EXPLAINED SUM OF SQUARES          .8352816E+10
RESIDUAL SUM OF SQUARES           .8649263E+10

NUMBER OF CASES                     299

PREDICTOR  V251   OCCUPATION B

      NO OF  SUM OF      CLASS      UNADJUSTED
CLASS  CASES  WEIGHTS  %      MEAN      DEVIATION FROM
      0     68     68  22.7  4592.206  -5936.115  -4256.094
      1     30     30  10.0  16396.07  5867.746   1165.547
      2     22     22   7.4  19716.09  9187.770   7577.927
      3     14     14   4.7  15615.71  5087.393   3987.124
      4     22     22   7.4   9988.636 -539.6847   547.4017
      5     42     42  14.0  12596.05  2067.727  1663.999
      6     36     36  12.0  10407.06 -121.2655   461.7471
      7     36     36  12.0   7910.333 -2617.988 -1574.841

```

8	21	21	7.0	11960.00	1431.679	1774.740
9	8	8	2.7	4009.000	-6519.321	-5901.890
CLASS	ADJUSTED MEAN		STANDARD DEVIATION			
0	6272.228		4161.586			
1	11693.87		9158.358			
2	18106.25		6896.417			
3	14515.45		11944.88			
4	11075.72		5269.902			
5	12192.32		5372.033			
6	10990.07		4254.318			
7	8953.480		5063.992			
8	12303.06		6163.097			
9	4626.431		2196.427			
ETA SQUARED =		.380238	BETA SQUARED		.195452	
ETA =		.616634	BETA		.442099	
ETA SQUARED (ADJ) =		.360938				
ETA (ADJ) =		.600781				
UNADJUSTED DEVIATION SS =		.646484E+10				
ADJUSTED DEVIATION SS =		.332309E+10				
PREDICTOR V30 MARITAL STATUS						
CLASS	NO OF CASES	SUM OF WEIGHTS	%	CLASS MEAN	UNADJUSTED DEVIATION FROM GRAND MEAN	COEFFICIENT
1	221	221	73.9	12449.90	1921.575	1123.470
2	17	17	5.7	7115.882	-3412.439	-2828.932
3	41	41	13.7	3732.463	-6795.858	-2956.380
4	16	16	5.4	5748.750	-4779.571	-4603.841
5	4	4	1.3	7640.000	-2888.321	-1330.495
CLASS	ADJUSTED MEAN		STANDARD DEVIATION			
1	11651.79		7563.060			
2	7699.389		4465.809			
3	7571.941		2752.520			
4	5924.480		4340.339			
5	9197.826		8306.206			
ETA SQUARED =		.194470	BETA SQUARE		.658475E-01	
ETA =		.440988	BETA		.256608	
ETA SQUARED (ADJ) =		.183511				
ETA (ADJ) =		.428382				
UNADJUSTED DEVIATION SS =		.330640E+10				
ADJUSTED DEVIATION SS =		.111955E+10				
PREDICTOR SUMMARY STATISTICS						
PREDICTOR V32 EDUC OF HEAD						
CLASS	NO OF CASES	SUM OF WEIGHTS	%	CLASS MEAN	UNADJUSTED DEVIATION FROM GRAND MEAN	COEFFICIENT
1	16	16	5.4	5973.375	-4554.946	-564.7311
2	71	71	23.7	6579.493	-3948.828	-2085.182
3	44	44	14.7	11013.86	485.5426	397.8526
4	70	70	23.4	10257.70	-270.6211	-789.0604
5	37	37	12.4	11210.03	681.7060	-1273.955
6	30	30	10.0	14161.87	3633.546	2836.744

7	17	17	5.7	16022.71	5494.385	3034.737
8	14	14	4.7	19327.71	8799.393	7518.277

CLASS	ADJUSTED MEAN	STANDARD DEVIATION
1	9963.590	6006.004
2	8443.139	4868.404
3	10926.17	8730.284
4	9739.261	6009.121
5	9254.365	5760.727
6	13365.06	7470.542
7	13563.06	6769.267
8	18046.60	12470.24

ETA SQUARE =	.203802	BETA SQUARED	.949135E-01
ETA =	.451445	BETA	.308080

ETA SQUARED (ADJ) =	.184650
ETA (ADJ) =	.429709

UNADJUSTED DEVIATION SS =	.346507E+10
ADJUSTED DEVIATION SS =	.161373E+10

ANALYSIS SUMMARY STATISTICS

DEPENDENT VARIABLE (Y) = V268 TOTAL FAMILY INC

R-SQUARED(UNADJUSTED) = PROP. OF VARIATION EXPLAINED BY FITTED MODEL: .49128

ADJUSTMENT FOR DEGREES OF FREEDOM = 1.07194

*** MULTIPLE R (ADJUSTED) = .67430 MULTIPLE R-SQUARED (ADJUSTED) = .45468

LISTING OF BETAS IN DESCENDING ORDER

RANK	VAR. NO.	NAME	BETA
1	V251	OCCUPATION B	.442099
2	V32	EDUC OF HEAD	.308080
3	V30	MARITAL STATUS	.256608

MERGE DATASETS

File Assignments:	DATASET1	First input data file
	DATASET2	Second input data file
	DATAOUT	Output data

GENERAL DESCRIPTION

Adds variables and records to datasets. Adding variables is usually required when processing data from a survey in which the same respondents were interviewed two or more times, leading to the creation of separate files for each set of variables. Adding records is required when respondents are interviewed in separate batches and must be combined. Note: You can also use FIX_DATASET with the Excel option to add records as well as to correct existing ones.

COMMAND FEATURES

MERGE can produce the intersection of two files (output only matched records), the union of two files (output one record for each unique ID in either input file as well as the matched records), or one record for each record in the first file. MERGE can pad unmatched records in either data set with missing-data codes.

To add records with the same variables, specify MATCH=UNION, specify the same variables in both VARS and ADDVARS, and do not specify RENUMBER.

RESTRICTIONS

The datasets must be in ascending sort order.

OPTIONS

Choose MERGE from the command screen and make selections.

ID=n Matching variable in each datafile. The ID variable may not be alphabetic or negative, or have fractional values.

MATCH=INTER|UNION|F1

INTER The output file is the intersection of the two input files.

UNION The output file is the union of the two input files.

F1 The output file is one record for each record in file 1 (DATASET1).

DUP Records with duplicate IDs in file 1 will each match the first record in file 2 with the same ID (a "many to one" match). For instance, file 1 might represent counties and file 2 represent states; you expect many counties per state. MERGE generates one output record per county with its associated

state data when DUP is specified. If DUP is not specified, all records are considered unique.

PRINT=(DICT,OUTD,NOMATCH)

DICT Print the input dictionary.

OUTD Print the output dictionary.

NOMATCH Prints the ID variable for unmatched records in either dataset.

PAD=MD1|MD2

For unmatched records, assign each variable:

MD1 its MD1 value.

MD2 its MD2 value.

Default: PAD=MD1.

RENUMBER=n

Renumber the variables, starting with n.

VARS=variable list

The variables to take from DATASET1. The ID variable is automatically included.

ADDVARS=variable list

The variables to take from DATASET2. The ID variable is automatically excluded.

EXAMPLE

The following example illustrates merging two files. The output dataset will contain one case for each unique ID in either dataset plus one for each pair of matched records (union). Cases in either file that do not match any case in the other dataset will be padded with missing-data values for the missing variables. Both datasets must be in sort order and must have the same ID variable numbers.

Options: print=(dict,outd) pad=md1 id=v1 match=union vars=2-4 addv=v5

*** MERGE DATASETS ***							
MERGING TWO FILES							
Dataset1 AFILE							
Dataset2 BFILE							
Creating dataset MFILE							
DICTIONARY 1							
	TYPE	LOC	WID	DEC	MD1	MD2	
V1 Interview number	C	1	4	0	0	99	
V2 Income (000)	C	5	4	1	990		
V3 Children	C	9	4	0			
V4 Weight 1	C	13	4	0			

DICTIONARY 2

	TYPE	LOC	WID	DEC	MD1	MD2
V1 Interview number	C	1	4	0	0	99
V5 Assets	C	17	4	2		

File 1: No match for ID 0

File 2: No match for ID 2

File 2: No match for ID 8

OUTPUT DICTIONARY

	TYPE	LOC	WID	DEC	MD1	MD2
V1 Interview number	C	1	4	0	0	99
V2 Income (000)	C	5	4	1	990	
V3 Children	C	9	4	0		
V4 Weight 1	C	13	4	0		
V5 Assets	C	17	4	2		

The output file is the union of both input files

5 variables and 7 cases written; record length: 20

1 unmatched record(s) found in file 1

2 unmatched record(s) found in file 2

Unmatched records padded with MD 1

MFILE set as default input dataset

MINISSA -- SMALLEST SPACE ANALYSIS

Guttman-Lingoes Smallest Space Analysis

File Assignments:	MATIN	Input data matrix or initial configuration matrix
	ADDPTS	Additional points matrix (optional)
	MATOUT	Output configuration matrix (optional)

GENERAL DESCRIPTION

MINISSA (Michigan Israel Netherlands Smallest Space Aalysis) is a non-metric multidimensional scaling command. Input to MINISSA is a matrix of similarity or dissimilarity coefficients (e.g., Pearson's r). The output is a geometric representation of the matrix in m dimensions. MINISSA constructs a configuration of points in space using information about the order relations among the coefficients. Because it is usually possible to satisfy the order relations of the coefficients in fewer dimensions than would be necessary to reproduce the metric information, the technique is called smallest space analysis.

An example may clarify what MINISSA does. Suppose we had a matrix of correlations between four variables, A, B, C, and D.

	A	B	C
B	.6		
C	.5	.8	
D	.4	.5	.1

Variable B and C are most similar (.8), variables A and B are next most similar (.6), etc. The correlation coefficients for each pair of variables have the following ranks:

Variable Pair	Correlation	Rank
BC	.8	1
AB	.6	2
AC,BD	.5	3.5 (because of tie)
AD	.4	5
CD	.1	6

There is no 1-dimensional solution (i.e., no way of placing the four variables on a line) which satisfies these order relations among similarities (B and C must be closest, A and B next closest, etc.). However, the following 2-dimensional solution satisfies the order relations:

C	A
B	
	D

Two things should be noted about this example. First, that the solution is one of many possible 2-dimensional solutions, which demonstrates that it is necessary to have many more variables than dimensions. (Using ten variables for example, which yields 45 inequalities, one should obtain a unique configuration in two dimensions.) Second, there are tied data--two input coefficients have the same value. In the solution diagrammed, the equal coefficients are represented by equal distances (between the pairs AC and BD); MINISSA, however, does not require that equal coefficients be represented by equal distances but rather treats ties as an indeterminate order relation and breaks them optimally.

Variables "fit" in a given number of dimensions if they can be portrayed in the correct relative position. Although the smallest space distance model is formulated as if it were necessary to get a perfect solution--where the mapping in a given number of dimensions reflects perfectly the order relations among the coefficients--adequate, albeit imperfect, solutions are allowed. MINISSA computes and prints two measures of goodness of fit: the coefficient of alienation and Kruskal's stress coefficient.

SPECIAL USES

Because only ordinal constraints are used for a solution, MINISSA can be used as an alternative to factor analysis when data do not meet the assumptions for factor analysis. Sometimes, a smallest space analysis is preferred to a factor analytic solution even when data meet the latter's assumptions; the fewer number of dimensions which usually result from a MINISSA smallest space analysis may make the results easier to interpret.

Probably the most useful result from MINISSA is the actual mapping of the data in multidimensional space. Clusters of variables can be spotted; dimensions can be discovered and labeled. For these reasons, smallest space analysis is often used prior to building an index.

COMMAND FEATURES

Input Configuration: The user may supply an initial configuration. Otherwise a Guttman-Lingoes initial configuration (Guttman, 1968, pp. 33, 469-506) is used to begin the computation.

Additional Points: One may add points to an existing configuration by inputting a fixed configuration together with coefficients that relate the new variables to each variable in the initial configuration (see example 2).

Scaling Algorithm: The program starts with an initial configuration and iterates over successive trial configurations each time measuring the fit of the configuration to the original coefficients. The "goodness of fit" is measured with the Guttman-Lingoes coefficient of alienation or with Kruskal's stress coefficient (Kruskal, 1964, pp. 29, 115-129).

Dimensionality and Metric: Solutions may be obtained in 1 to 10 dimensions. There is also a choice of distance metric to use: Euclidean or Cityblock.

Ties: Equal values among the input coefficients are treated as ties and given equal rank. The algorithm however does not require that equal coefficients be represented as equal distances, but rather treats ties as an indeterminate order relation and breaks them optimally.

Monotonicity: Normally the program assumes global monotonicity, i.e., all errors are treated equally. However, local monotonicity may be specified in which case errors in small distances are weighted more than errors in large distances.

Missing Data: There is an option to skip cells in the input data matrix which have a value equal to a specified missing-data code (MDCODE=n). Use MDCODE=99.999 for MicrOsiris correlation matrices.

SPECIAL TERMINOLOGY

Guttman-Lingoes Coefficient of Alienation: A measure of goodness of fit. When this coefficient is minimized, strong (no ties in the input data) or semi-strong (ties in the input data) monotonicity is satisfied. (Lingoes and Roskam, 1973, pp. 26-29, 45-49).

Kruskal's Stress Coefficient: A measure of goodness of fit similar to the Guttman-Lingoes coefficient of alienation. Using Kruskal's stress coefficient, semi-weak or weak monotonicity is satisfied. (Lingoes and Roskam, 1973, pp. 26-29, 45,49).

Monotonicity: Global monotonicity refers to treating all errors equally. Local monotonicity refers to weighting errors in small distances more heavily. The different kinds of monotonicity are described in Guttman, *Psychometrika*, 1968.

Smallest Space Analysis (SSA): Method of analysis in which the order relationships among a set of coefficients are usually satisfied in fewer dimensions than would be necessary to reproduce the metric information.

PRINTED OUTPUT

For each solution:

The coordinates of each variable in space are printed together with the 'centrality index'. The centrality index can be used to weed out outliers whose effect might distort the representation. Variables that have a lot in common with other variables will have low numbers on the centrality index. (See Lingoes and Roskam, 1971, p. 143.)

Coefficient of alienation: a summary statistic that measures the goodness of fit of the data in the space. (See Lingoes and Roskam, 1971, p. 47, equation 73.)

Kruskal's stress, a measure of goodness of fit similar to the coefficient of alienation. (See Lingoes and Roskam, 1971, p. 26, equations 25-27.)

Vector plots: For each pair of dimensions, the variables are plotted. Thus, for each solution, there are $M*(M-1)/2$ plots, where M is the number of dimensions.

The matrix of distances between variables for this solution, i.e., the derived coefficients is printed if PRINT=DIST is specified. If however, you request that the program determine the proper dimensionality (MIND=0), only one solution will include the plot.

INPUT DATA

The input to MINISSA consists of a data matrix with or without an initial configuration matrix, or an initial configuration matrix with an additional-points matrix.

Usually the input to a smallest space analysis is a matrix of correlation coefficients, e.g., a matrix of taus, a matrix of gammas or a matrix of Pearson r's. The two formal requirements are: 1) that the matrix is symmetric and, 2) that order holds over all the elements of the matrix.

Data Matrix: A symmetrical MicroSiris matrix.

Configuration Matrix: Optional. Provides a starting configuration. The rows should match the variables and the columns the dimensions. It is usually produced by a previous use of MINISSA. Note that the number of dimensions must be equal to MAXD (See Options).

Additional Points Matrix: The coefficients in this matrix are measures of proximity relating new variables to a set of variables in an already existing configuration. Each row represents one additional point. The columns represent the variables in the initial configuration. When an additional matrix is used, the configuration for the original variables remains fixed.

OUTPUT DATA

Output Configuration Matrix: When WRITE is specified, the output configuration matrix is written to MATOUT. This matrix consists of one row for each variable in the configuration, with the columns representing the dimensions numbered 1 through n. The matrix may be input to MINISSA in a subsequent MINISSA for further iterations. Note that as many matrices will be written as solutions attempted (i.e., MAXD - MIND + 1 matrices).

OPTIONS

Choose MINISSA from the command screen and make selections.

For a Runfile use:	MINISSA Filter statement (optional) Job Title Keyword choices from below
--------------------	-----------------------------------------------------------------------------------

VAR=(variable numbers)|ALL Use the variables specified in the list.

MATRIX=n Input data matrix number.

AMATRIX=n Use additional points matrix. Can use only with an initial configuration matrix that is to remain fixed.

CMATRIX=n Use initial configuration n. When CMATRIX is specified, an additional-points matrix or an input data matrix must be supplied.

CUT=x Value above or below which input coefficients are considered tied.

EUCLID|CITYBLOCK

EUCLID: Use Euclidean metric.

CITY: Use Cityblock metric.

Default: EUCLID.

GLMIN|KMIN GLMIN: Guttman-Lingoes coefficient of alienation will be minimized.
KMIN: Kruskal's stress coefficient will be minimized.

Default: GLMIN.

MIND=n Minimum number of dimensions. If MIND=0 or is not specified MINISSA determines the minimum dimension. MIND must be less than or equal to MAXD and in the range 0 to 10.

Default: MIND=2.

MAXD=n Maximum number of dimensions. Must be in the range 1 to 10.

Default: MAXD=2

MAXTIES=n The maximum number of ties allowed.

Default: MAXTIES=100.

MDCODE=x Missing-data value; delete from the analysis, those cells of the data matrix which contain the value x. If MDCODE is not specified, the program will not check for missing-data.

MONOTONICITY=GLOBAL|LOCAL

GLOBAL: All errors treated equally.

LOCAL: Errors in small distances weighted more than in large distances

Default: GLOBAL.

PRINT=(MATRIX,DIST,CONF)

MATRIX: Print input data matrix.

DIST: Print the derived distance matrix.

CONF: Print input configuration matrix.

SIMILARITIES|DISSIMILARITIES

SIMI: Large coefficients indicate that points are similar or close.

DISS: Large coefficients indicate that points are dissimilar or far.

Default: SIM.

TIES=EQUAL|LARGE|SMALL

EQUAL: Treat only equal values as ties.

LARGE: Treat large distances as tied. That is, for dissimilarities, treat all values greater than the CUT value as tied; for similarities, treat all values less than CUT value as tied.

SMALL: Treat small distances as tied. That is, for dissimilarities, treat all value less than CUT value as tied; for similarities, treat all values great than the CUT value as tied.

WRITE Write the output configuration of to the file assigned to MATOUT.

OUTM=n The number use for the output configuration matrix; required when WRITE is also specified or the output configuration matrix is to be used in a subsequent command. *Default:* OUTM=999.

TITLE Output matrix title

EXAMPLE

Map of Michigan: A smallest space analysis on 14 Michigan cities. The data are straight-line mileages between cities in the State of Michigan. The plot is easily recognizable as map of Michigan (turned sideways so that north is to the left). The configuration matrix is written to file ssa_config assigned to MATOUT.

Options: MAXD=2 MIND=2 DISS PRINT=MATRIX

```
*** GUTTMAN-LINGOES SMALLEST SPACE ANALYSIS ***

      MAPPING MICHIGAN CITIES

Using matrix file MINISSA\MICHMAP

Creating matrix file MINISSA\CONFIG.SSA

Using input matrix 1

The input matrix consists of dissimilarities

MIN dimensions = 2   MAX dimensions = 2

Output configuration matrix number: 2

Coefficient for minimization is Guttman-Lingoes

Distances are euclidean

Equal values are considered ties

Matrix 1 read in successfully.

Matrix 3 read in successfully.

ORIGINAL COEFFICIENTS
```

		V1	V2	V3	V4	V5	V6	V7
		ALPENA	ANN ARBOR	BENTON HARBOR	CHEBOYGAN	DETROIT	GRAND RAPIDS	GRAYLING
ALPENA	V1	0.000						
ANN ARBOR	V2	189.000	0.000					
BENTON HARBOR	V3	251.000	149.000	0.000				
CHEBOYGAN	V4	64.000	231.000	261.000	0.000			
DETROIT	V5	186.000	35.000	174.000	237.000	0.000		
GRAND RAPIDS	V6	178.000	107.000	72.000	190.000	138.000	0.000	
GRAYLING	V7	64.000	168.000	195.000	68.000	178.000	124.000	0.000
KALAMAZOO	V8	216.000	92.000	47.000	234.000	128.000	47.000	167.000
LANSING	V9	167.000	51.000	106.000	198.000	80.000	58.000	131.000
LUDINGTON	V10	165.000	175.000	126.000	150.000	201.000	76.000	98.000
PORT AUSTIN	V11	72.000	125.000	218.000	130.000	115.000	150.000	93.000
PORT HURON	V12	150.000	82.000	212.000	207.000	53.000	161.000	160.000
SAGINAW	V13	113.000	78.000	156.000	153.000	86.000	98.000	92.000
TRAVERSE CITY	V14	108.000	192.000	186.000	82.000	208.000	121.000	45.000

ORIGINAL COEFFICIENTS - continued

		V8 KALAMAZOO	V9 LANSING	V10 LUDINGTON	V11 PORT AUSTIN	V12 PORT HURON	V13 SAGINAW	V14 TRAVERSE CITY
KALAMAZOO	V8	0.000						
LANSING	V9	60.000	0.000					
LUDINGTON	V10	121.000	125.000	0.000				
PORT AUSTIN	V11	175.000	117.000	169.000	0.000			
PORT HURON	V12	165.000	107.000	209.000	78.000	0.000		
SAGINAW	V13	112.000	56.000	129.000	62.000	81.000	0.000	
TRAVERSE CITY	V14	168.000	148.000	68.000	137.000	198.000	122.000	0.000

GUTTMAN-LINGOES SMALLEST SPACE COORDINATES FOR M = 2 (semi-strong monotonicity)

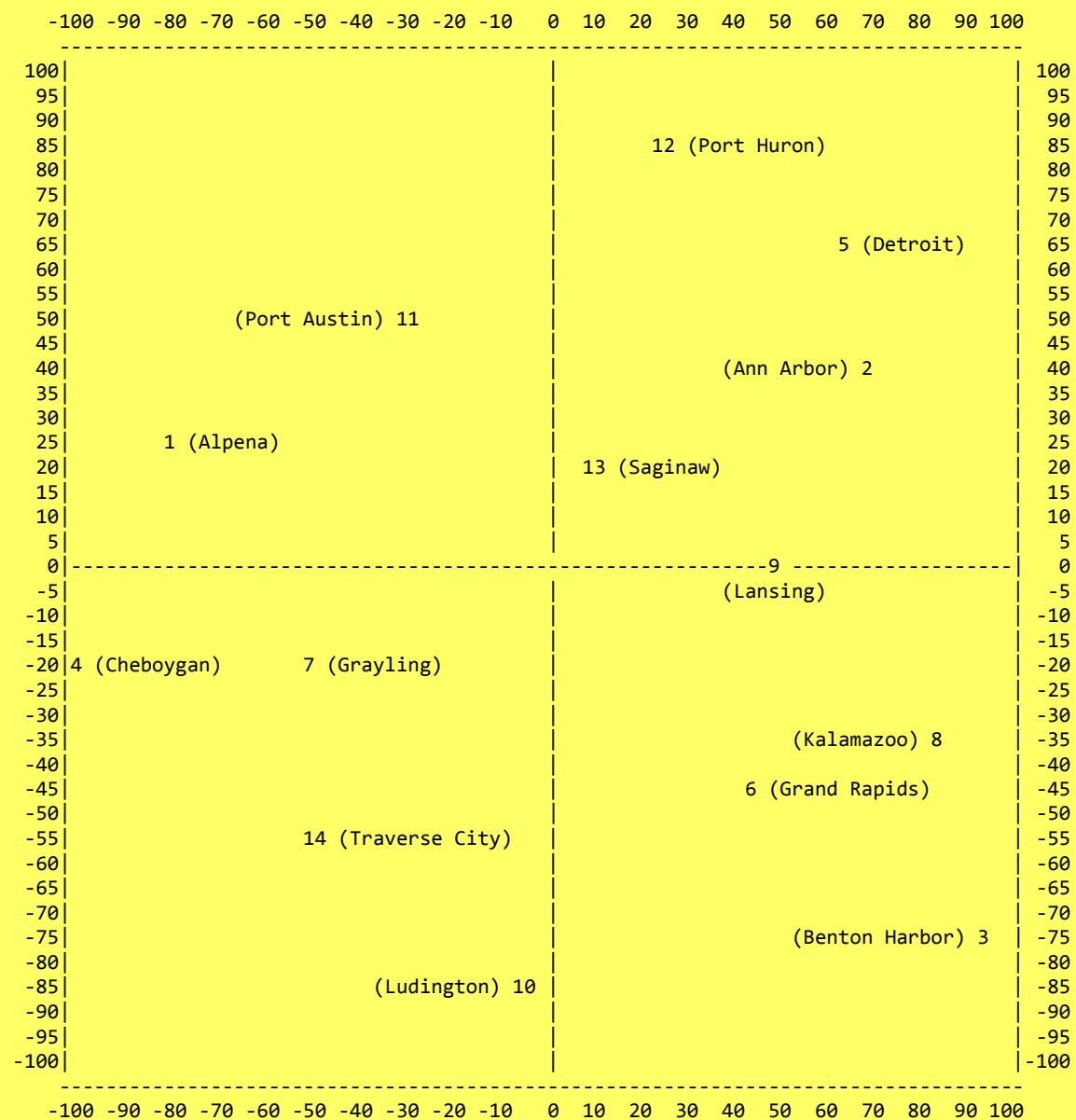
		DIMENSION 1	DIMENSION 2	CENTRALITY INDEX
ALPENA	V1	-0.742	0.242	0.900
ANN ARBOR	V2	0.747	0.369	0.773
BENTON HARBOR	V3	1.000	-0.726	1.115
CHEBOYGAN	V4	-1.000	-0.174	1.110
DETROIT	V5	0.686	0.617	0.893
GRAND RAPIDS	V6	0.482	-0.448	0.540
GRAYLING	V7	-0.450	-0.197	0.571
KALAMAZOO	V8	0.856	-0.350	0.805
LANSING	V9	0.546	-0.006	0.445
LUDINGTON	V10	0.022	-0.814	0.757
PORT AUSTIN	V11	-0.231	0.492	0.647
PORT HURON	V12	0.292	0.814	0.895
SAGINAW	V13	0.148	0.194	0.259
TRAVERSE CITY	V14	-0.438	-0.559	0.737

Guttman-Lingoes coefficient of alienation = 0.00141 in 22 iterations

Kruskal's stress: 0.00076

PLOT LABEL	VARIABLE	NAME
1	V1	ALPENA
2	V2	ANN ARBOR
3	V3	BENTON HARBOR
4	V4	CHEBOYGAN
5	V5	DETROIT
6	V6	GRAND RAPIDS
7	V7	GRAYLING
8	V8	KALAMAZOO
9	V9	LANSING
10	V10	LUDINGTON
11	V11	PORT AUSTIN
12	V12	PORT HURON
13	V13	SAGINAW
14	V14	TRAVERSE CITY

Vector 2(row) plotted against vector 1(column)



MNA -- MULTIVARIATE NOMINAL ANALYSIS

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

MNA performs a multivariate analysis of nominal-scale dependent variables, using a series of parallel dummy-variable regressions derived from each of the dependent variable codes, dichotomized to a 0-1 variable. The program's major use is to give an additive multivariate model showing the relationship between a set of predictors and the dependent variable in terms of a set of coefficients analogous to MCA coefficients.

The advantage MNA has over other techniques applicable to the same data is the simplicity and direct interpretability of the MNA coefficients and the categorical prediction algorithm. See Andrews and Messenger, *Multivariate Nominal Scale Analysis* for a complete description of the MNA technique.

COMMAND FEATURES

Statistics: MNA computes the univariate distribution of the dependent variable, gives (in effect) a bivariate distribution of the dependent variable with each predictor, and computes and prints the multivariate "MNA coefficients." Bivariate statistics are the bivariate theta and the code specific and generalized eta-square; they provide two alternatives for measuring the strength of the simple bivariate relationship between a specific predictor and the dependent variable. The program also prints a series of statistics for each predictor called "Beta Square." These indicate the relative importance of the predictor when holding all other independent variables constant. Multivariate statistics are the multivariate theta and the code specific and generalized R-square.

Missing data: Cases with missing data on the dependent variable may be eliminated with the DDELETE=(MD1,MD2) option. Cases with missing data on the independent variables may be eliminated with the DELETE=(MD1,MD2) KEYWORD.

RESIDUALS OUTPUT

For each case input to MNA there is a set of residual scores--one score for each dependent variable code. Residuals are defined as the difference between the dummy variable score belonging to the particular code and the forecast value for that code. Thus the residual scores are the differences between the MNA-derived probabilities and 0 or 1, depending on whether or not the object actually did fall in the designated category of the dependent variable.

SPECIAL TERMINOLOGY

MNA coefficients: An array of statistics, one for each pair of dependent variable and predictor variable codes, which are transformed dummy variable regression coefficients. They are transformed to include a coefficient for each predictor variable code.

Classification matrix: A matrix indicating the pattern of correct categorical predictions made by MNA. Rows are actual codes and columns are predicted codes.

Forecast: A set of predictions (summing to 1.0), where prediction $P(i)$ is the probability estimate of the occurrence of the i -th dependent variable code, given the predictor variable characteristics present.

Generalized eta-square and R-square: Extensions of the bivariate eta-square and multivariate R-square. These statistics are appropriate for assessing the strength of relationship on the entire set of dummy dependent variable codes. You can view them as a variance-weighted average of the standard statistics over the complete set of dummy dependent variables.

Theta (bivariate and multivariate): Statistics representing the proportion of cases that are correctly classed when using a prediction-to-the-mode strategy in the bivariate sense and prediction-to-the-maximum forecast in the multivariate sense.

PRINTED OUTPUT

See [Interpreting MNA Output](#) for help in understanding the following statistics.

Information on the analysis

- Numbers of cases eliminated due to missing data on the dependent variable and range of valid codes

- Non-empty predictor codes

- Minimum number of significant digits in solution vectors

Dependent Variable Statistics

- Frequency distribution

- Weighted frequency distribution

- Weighted frequency distribution expressed as a percent

- R-squared (for each dependent variable code)

- Adjusted R-squared (for each dependent variable code)

Predictor Variable Statistics

Frequency for each code

- Weighted frequency for each code

- Weighted frequency expressed as a percent for each code

For each predictor code:

- Weighted frequency marginal for each code of the dependent variable (Y) expressed as percents

- Adjusted percents (sums of percents and coefficients) for each dependent variable code

- Coefficients for each code of the dependent variable

- Theta

- Eta-squared (for each dependent variable code)

- Beta-squared (for each dependent variable code)

- Generalized eta-squared

Joint and Multivariate Prediction

Generalized R-squared

Joint theta (proportion of cases correctly classed)

Classification matrix

Rows of the matrix indicate actual codes; columns indicate predicted codes.

INTERPRETING MNA OUTPUT

Consult the example printout at the end of this write-up as noted in the following discussions. See *Multivariate Nominal Scale Analysis* (Andrews and Messenger, 1973) for a complete description of how to interpret MNA results.

Examination Strategies

In looking at a large number of detail statistics from MNA, two things are of particular interest: 1) large coefficients, and 2) large differences between the percents and the adjusted percents.

If an independent variable is ordinal scale, the occurrence of monotonic change across successive coefficients or percentages may also be of interest. This occurs in the example in the way V46, "Better or worse a year from now" affects the likelihood of the first car being a compact.

Theta Statistic

The multivariate statistic Theta indicates the proportion of cases correctly classified after taking into account each respondent's scores on all dependent variables. In the example, Theta is .8043 indicating that 80% of the cases could be correctly classified after taking into account each respondent's scores on all independent variables. This is a gain of more than 10 percentage points over the mode of the overall percentage distribution (69.6% for "Large" car).

Identifying the mode is important; it shows that even if you know nothing about the respondents, you could predict the first car for everyone is large and be correct 69.6% of the time. Relationships of the independent variables to the dependent variable act to increase predictability above this 69.6% level.

The bivariate Theta statistic indicates the proportion correctly classified for a single independent variable.

Forecasts and the Proportion Classed Correctly

For any case a forecast can be derived. The forecast consists of a set of probabilities; it shows the likelihood of that case falling into each category of the dependent variable. You compute the probability for each category by summing the coefficients relevant to that case and adding in the overall percent. Assume we have a person who earns \$40,000 a year, is 28 years old, single, has a college degree, expects to be about as well off next year, expects his/her income to be a little bit more next year, and holds a professional position. The forecast is computed as shown in the table below:

Size of First Car	Small	Compact	Mid-Size	Large
Overall Percents	7.2	8.7	14.5	69.6
Coeff: \$40,000/yr	-5.05	8.10	2.30	-5.35
Coeff: 28 Years old	11.41	-.25	13.13	-24.29
Coeff: Single	10.40	-2.10	-1.11	-7.19
Coeff: College Degree	15.64	-4.17	1.23	-12.69
Coeff: About the Same.	6.73	-4.91	-4.25	2.44
Coeff: A Little More Income	2.03	-.97	-2.74	1.68
Coeff: Professional	-19.31	1.05	15.73	2.52
Forecast:	29.10	5.45	38.78	26.69

The forecast gives a set of predicted scores for each case; you predict a case to be in the dependent variable for which the probability is highest. The person represented in the table above would be assigned the "Mid-Size" category.

INPUT DATA

MNA is designed to analyze a nominally scaled dependent variable with three or more code categories. The MNA coefficients and summary statistics are identical to (or generalizations of) those that would be produced by parallel MCA runs dichotomizing each dependent variable code against the others.

RESTRICTIONS

Code categories must be in the range 0 to 999.

OPTIONS

Choose MNA from the command screen and make selections.

For a Runfile use:	MNA
	Filter statement (optional)
	Job Title
	Keyword choices from below

DEPV=n The dependent variable.

MAXD=n The maximum number of dependent variable codes.
Default: MAXD=10

VAR=(variable list) The list of independent variables.

DDELETE=(MD1,MD2)

MD1 Delete all cases where the dependent variable equals its first missing-data code.

MD2 Delete all cases where the dependent variable equals its second missing-data code.

DELETE=(MD1,MD2)

MD1 Delete all cases where any independent variable equals its first missing-data code.

MD2 Delete all cases where any independent variable equals its second missing-data code.

RECODE=n Use RECODE n, previously entered via the RECODE command.

RESIDUALS=DATASET|RECODE

Create recode to compute residuals with residual variable numbers 10001-1000n corresponding to the first, second, etc. dependent variable codes.

DATASET Write a dataset using the recode.

RECODE Create recode only for use in subsequent commands.

WT=n Use variable n as a weight variable

REFERENCES

Andrews, F. M., J. N. Morgan, J. A. Sonquist and L. Klem. *Multiple Classification Analysis*. Second edition. Ann Arbor: Institute for Social Research, The University of Michigan, 1973.

Andrews, F. M. and R. C. Messenger. *Multivariate Nominal Scale Analysis*. Ann Arbor: Institute for Social Research, The University of Michigan, 1973.

EXAMPLE

Explaining size of first car for childless families. Predictors are income (bracketed), age of head of household (bracketed), education, and feelings of "well-offness."

Recode statements: R1=BRAC(V268,<1=1,1-4500=2,4501-9500=3,9501-15000=4, -
15001-21000=5,>21000=6)
R2=BRAC(V20,0-30=1,31-45=2,46-60=3,>60=4)
NAME R1'Bracketed income',R2'Bracketed age'

*** MNA -- MULTIVARIATE NOMINAL ANALYSIS ***

EXPLAINING SIZE OF FIRST CAR FOR CHILDLESS FAMILIES

Dataset SCF

Including V193=1-8 AND V26=0

Transforming the data with RECODE 1 read from MNA\MNA_EXAMPLE1.RUN

For the dependent variable, values are deleted for cases with MD1 or MD2

138 cases accepted

PREDICTOR	NON-EMPTY CODES							
R1 Bracketed income	2	3	4	5	6			
R2 Bracketed age	1	2	3	4				
V32 Educ of head	1	2	3	4	5	6	7	8
V46 B/w year from now	1	3	5	8				

Minimum number of significant digits in the solution vectors is 4

Dependent variable V193 Size of car

Code	1	2	3	5	
	Small	Compact	Mid-Size	Large	Total
Frequency	10	12	20	96	138
Percent	7.25	8.70	14.49	69.57	100.00
R-squared	0.0991	0.2162	0.1400	0.1986	
Adjusted	0.0000	0.1051	0.0182	0.0851	

R1 Bracketed income

	1	2	3	5
	Small	Compact	Mid-Size	Large
ETA Squared	0.0164	0.0514	0.0047	0.0407
BETA Squared	0.0164	0.0437	0.0031	0.0196

Generalized ETA Squared .0299
Bivariate THETA .6957

CODE		Size of car					
			1	2	3	5	
			Small	Compact	Mid-Size	Large	
2		Percent	2.94	8.82	11.76	76.47	
	N	34	Adj Pct	3.80	13.12	12.88	70.20
	Pct	24.64	Coeff.	-3.45	4.42	-1.61	0.64
3		Percent	6.67	2.22	15.56	75.56	
	N	45	Adj Pct	5.90	5.38	16.61	72.11
	Pct	32.61	Coeff.	-1.34	-3.31	2.11	2.54
4		Percent	12.12	15.15	15.15	57.58	
	N	33	Adj Pct	12.73	9.80	13.22	64.24
	Pct	23.91	Coeff.	5.48	1.11	-1.27	-5.32
5		Percent	6.25	18.75	18.75	56.25	
	N	16	Adj Pct	6.10	16.47	16.67	60.76
	Pct	11.59	Coeff.	-1.15	7.77	2.18	-8.80
6		Percent	10.00	0.00	10.00	80.00	
	N	10	Adj Pct	8.74	-7.52	11.17	87.61
	Pct	7.25	Coeff.	1.50	-16.21	-3.33	18.04

R2 Bracketed age

	1	2	3	5
	Small	Compact	Mid-Size	Large
ETA Squared	0.0210	0.0282	0.0580	0.1140
BETA Squared	0.0191	0.0073	0.0494	0.0522

Generalized ETA Squared .0725
Bivariate THETA .6957

CODE		Size of car					
			1	2	3	5	
			Small	Compact	Mid-Size	Large	
1		Percent	14.29	19.05	33.33	33.33	
	N	21	Adj Pct	14.51	8.81	31.72	44.96
	Pct	15.22	Coeff.	7.26	0.12	17.22	-24.60
2		Percent	0.00	10.00	20.00	70.00	
	N	10	Adj Pct	4.71	3.23	14.82	77.25
	Pct	7.25	Coeff.	-2.54	-5.47	0.33	7.68

3			Percent	8.51	8.51	8.51	74.47
	N	47	Adj Pct	8.37	11.56	7.67	72.40
	Pct	34.06	Coeff.	1.12	2.86	-6.82	2.83
4			Percent	5.00	5.00	11.67	78.33
	N	60	Adj Pct	4.25	7.32	13.75	74.68
	Pct	43.48	Coeff.	-3.00	-1.37	-0.74	5.11

V32 Educ of head

	1	2	3	5
	Small	Compact	Mid-Size	Large
ETA Squared	0.0564	0.1053	0.0594	0.0587
BETA Squared	0.0545	0.0811	0.0698	0.0323

Generalized ETA Squared .0662
Bivariate THETA .6957

CODE			Size of car			
			1	2	3	5
			Small	Compact	Mid-Size	Large
1	0-8th Grade	Percent	0.00	0.00	0.00	100.00
	N	5	Adj Pct	4.08	0.10	96.25
	Pct	3.62	Coeff.	-3.16	-8.60	26.68
2	9th Grade	Percent	7.32	7.32	7.32	78.05
	N	41	Adj Pct	8.80	7.63	71.70
	Pct	29.71	Coeff.	1.56	-1.06	2.13
3	10th Grade	Percent	4.55	0.00	22.73	72.73
	N	22	Adj Pct	2.94	4.21	66.21
	Pct	15.94	Coeff.	-4.31	-4.49	-3.36
4	11th Grade	Percent	3.70	3.70	22.22	70.37
	N	27	Adj Pct	4.35	0.99	73.83
	Pct	19.57	Coeff.	-2.90	-7.71	4.27
5	Completed HS	Percent	0.00	25.00	12.50	62.50
	N	16	Adj Pct	-1.62	21.20	73.93
	Pct	11.59	Coeff.	-8.87	12.51	4.37
6	Some College	Percent	17.65	23.53	5.88	52.94
	N	17	Adj Pct	15.99	23.28	60.72
	Pct	12.32	Coeff.	8.74	14.58	-8.85
7	College Degree	Percent	16.67	0.00	33.33	50.00
	N	6	Adj Pct	14.48	4.09	51.28
	Pct	4.35	Coeff.	7.24	-4.60	-18.29
8	Graduate Degree	Percent	25.00	0.00	25.00	50.00
	N	4	Adj Pct	25.95	1.92	51.58
	Pct	2.90	Coeff.	18.70	-6.78	-17.98

V46 B/w year from now

	1	2	3	5
	Small	Compact	Mid-Size	Large
ETA Squared	0.0063	0.1015	0.0297	0.0828
BETA Squared	0.0193	0.0794	0.0266	0.0513

Generalized ETA Squared .0616
Bivariate THETA .6957

CODE			Size of car			
			1	2	3	5
			Small	Compact	Mid-Size	Large
1	Much Worse	Percent	3.85	26.92	26.92	42.31
	N	26	Adj Pct	1.08	24.66	48.24

	Pct	18.84	Coeff.	-6.16	15.96	11.53	-21.33
3	A Little Worse		Percent	7.04	5.63	11.27	76.06
	N	71	Adj Pct	7.14	6.34	11.48	75.03
	Pct	51.45	Coeff.	-0.10	-2.35	-3.01	5.46
5	About the Same		Percent	9.09	4.55	13.64	72.73
	N	22	Adj Pct	9.76	4.59	14.79	70.86
	Pct	15.94	Coeff.	2.51	-4.11	0.30	1.29
8	Much Better		Percent	10.53	0.00	10.53	78.95
	N	19	Adj Pct	13.15	0.39	9.62	76.84
	Pct	13.77	Coeff.	5.90	-8.30	-4.87	7.28

*** MULTIVARIATE STATISTICS ***

Generalized R-Squared .1726 Multivariate Theta .7609

CASES CORRECTLY CLASSIFIED

	1	2	3	5
	Small	Compact	Mid-Size	Large
	0.000	5.000	5.000	95.000
PROPORTION	0.000	0.417	0.250	0.990

Total cases correctly classified: 105 out of 138 Proportion: .761

ACTUAL(rows) vs. PREDICTED(columns) CLASSIFICATION MATRIX

		1	2	3	5	
		Small	Compact	Mid-Size	Large	Total
Small	1	0	1	1	8	10
	ROW %	0.00	10.00	10.00	80.00	100.00
Compact	2	0	5	0	7	12
	ROW %	0.00	41.67	0.00	58.33	100.00
Mid-Size	3	0	0	5	15	20
	ROW %	0.00	0.00	25.00	75.00	100.00
Large	5	0	1	0	95	96
	ROW %	0.00	1.04	0.00	98.96	100.00
Total		0	7	6	125	
ROW %		0.00	5.07	4.35	90.58	100.00

NON-PARAMETRIC STATISTICS

See [TABLES.](#)

PROBABILITY CALCULATOR

An interactive command that calculates probabilities for F, t, chi-square, and normal statistics specified by filling in the blanks on the prompt screen.

Example 1: Calculate probability of a given t-statistic

The screenshot shows the 'Probability' dialog box with the 'Statistic' section containing four radio buttons: 'T' (selected), 'F-test', 'Chi-square', and 'Normal Z'. The 'Value' field contains '11.4173'. The 'Degrees of Freedom' field contains '1'. The 'Degrees of Freedom for denominator of F' field is empty. The 'Estimated probability' field shows '0.0551'. An 'OK' button is in the top right corner.

Example 2: Find probability for a chi-square value with 4 degrees of freedom.

The screenshot shows the 'Probability' dialog box with the 'Statistic' section containing four radio buttons: 'T', 'F-test', 'Chi-square' (selected), and 'Normal Z'. The 'Value' field contains '10.61'. The 'Degrees of Freedom' field contains '4'. The 'Degrees of Freedom for denominator of F' field is empty. The 'Estimated probability' field shows '0.0314'. An 'OK' button is in the top right corner.

Example 3: For probability of an F-statistic, degrees of freedom for both numerator and denominator must be specified in the PROB command.

The screenshot shows the 'Probability' dialog box with the 'Statistic' section containing four radio buttons: 'T', 'F-test' (selected), 'Chi-square', and 'Normal Z'. The 'Value' field contains '2.826'. The 'Degrees of Freedom' field contains '3'. The 'Degrees of Freedom for denominator of F' field contains '160'. The 'Estimated probability' field shows '0.0396'. An 'OK' button is in the top right corner.

RECODE

GENERAL DESCRIPTION

The MicroSiris RECODE command provides recoding of data. Uses include collapsing of variable categories and creation of user-defined measures (variables).

Recode variables are created with the RECODE command and subsequent commands refer to them as if they existed in the original dataset. RECODE provides only temporary recoding except when used with the [TRANSFORM](#) command which creates a permanently recoded dataset.

The RECODE command analyzes and decodes recode statements. Actual computation takes place during execution of a subsequent PGMNAME command as the data is read, case by case. Reference to a variable in a recode statement is actually a reference to the value of that variable for the particular case being examined. Execution flows sequentially (i.e., the first statement is processed, then the second, third, etc.) except as modified by control statements. When all statements are processed, the recoded case is passed to the command using the data. When the command asks for another case, recoding begins again with the next case.

RECODE processes only those cases passing the filter, if one is present. It can act as an additional filter through use of the REJECT and ENDFILE statements.

A single RECODE command defines one or more independent sets of RECODE statements. Each set consists of a recode number, a set of recode statements, and an END statement. PGMNAME commands specify which set to use via the RECODE option in the command setup. The recode number statement (RECODE n), and END statements are described in the [STATEMENT AND FUNCTION DESCRIPTIONS](#) section. The RECODE n statement may be omitted if only one recode set is desired--the recode number then defaults to 1.

The actual data recoding defined by RECODE takes place only when "RECODE n" is used in a subsequent procedure command setup.

Although you may enter RECODE statements interactively, this can become very tedious; and correcting errors can be complicated or impossible when conditional or program flow statements are entered. The best way to create recodes is to use a text editor like notepad to create a RECODE file. Then use the created file as a Runfile or in another Runfile with the [INCLUDE statement](#) parameter.

SPECIAL TERMINOLOGY

Argument Lists: The one or more constants, variables, expressions or special strings of characters that follow a function reference (V10 in the example under [Function Reference](#)).

Arithmetic Expressions: Expressions that yield numeric values. Examples are:

2*V10	2 multiplied by the value of V10.
44	the constant 44.
R3	the value of R3.
R67/V98+25	25 more than the value of R67 divided by the value of V98.

Arithmetic Functions: Functions that yield numeric values.

ABS, BRAC, LOG, MD1,MD2, SQRT, TABLE, and TRUNC.

Arithmetic Operators: Used between (or with) arithmetic operands. They are:

- (negation)
- EXP x (exponentiation of a positive number to the power x, where x is a decimal or integer number and $-180 \leq x \leq 174$, e.g., 2 EXP 3 is the cube of 2)
- * (multiplication)
- / (division)
- +
- (subtraction)

Assignment Statements: Contain the assignment symbol (=). An example is: R5=V16/15.

Constants: Fixed, unvarying quantities such as numbers or letters.

Control Statements: Influence the order in which RECODE carries out its instructions. They are: executed when RECODE processes the data file. The control statements are: GO TO, CONTINUE, RETURN, REJECT, ENDFILE, ERROR, and RELEASE.

Definition Statements: Influence what RECODE does at the time it processes the RECODE setup. Definition statements are: CARRY, LABEL, NAME and TABLE.

Expression: A representation of a value. A single constant, variable, or function reference is an expression. Combinations of constants, variables, function references, and other expressions by operators are expressions. RECODE evaluates arithmetic and logical expressions.

Floating-Point Constants: Numbers written with decimal points (e.g., 1.5, 2.0, -543.21).

Function: A pre-defined sequence of operations, which yields a single arithmetic or logical value; e.g., the ABS function yields the absolute value of a number.

Function Reference: Consists of the name of the function and its argument list (described above). An example is:

ABS(V10) The absolute value of V10.

Function Return: The value returned after a function is performed. The "function reference" itself can generally be used as an operand in an expression, although some functions (such as RECODE) may not be combined with other statements or functions.

Integer Constants: Numbers without decimal fractions (e.g., 3, 44, -99).

Logical Expression: An expression which assumes a "true" or "false" value. Examples are:

(V62 GT 10) OR (R5 EQ V333)

True if either of the relational operations results in a true value; false if both of the relational operations result in a false value.

R5 EQ V333 True if R5 is equal to V333; false otherwise.

MDATA(V10,R20)

True if V10 is a missing-data code or if R20 is a missing-data code.

MDATA (V3) AND V9 GT 2

True if both V3 is a missing-data code and V9 is larger than 2; false otherwise.

Logical Functions: Functions that yield truth values when referenced (i.e., "true" or "false"). Logical functions are INLIST and MDATA.

Logical Operators: Used between logical operands. These are:

NOT
AND
OR

Operator: A symbol that expresses an operation to be performed on constants, variables, or expressions. RECODE recognizes arithmetic, relational, and logical operators.

Operands: Constants, variables, or expressions combined with operators.

Relational Operators: Used to determine whether or not two arithmetic values have a particular relationship to one another. The relational operators are:

LT	(less than)
LE	(less than or equal)
GT	(greater than)
GE	(greater than or equal)
EQ	(equal)
NE	(not equal)
<	Used only in BRAC statement
>	Used only in BRAC statement

Variables: Variables in RECODE are defined as follows:

Input Variables (Vn). "V" followed by a number. Vn represents the value of a variable as defined by the input dictionary (e.g., V10 is the value of input variable 10). A variable value may be changed by RECODE (e.g., V10=R3).

Result Variables (Rn). "R" followed by a number. Rn represents the value of a variable created by RECODE (e.g., R5 is the value of result variable 5).

R-variables may be used with any command (e.g., WT=R5 or V=R10-R20.) R- and V-variables used in RECODE must be in the range 1-99999.

Univariate Recoding

Sometimes called "code collapsing" or "bracketing", univariate recoding allows a result or input variable to be assigned a new value based on stated rules applied to the current value of that variable. For example, the following statements illustrate two methods of recoding a variable, V3, that has values 0, 1- 9, such that codes 1-5 remain unchanged, code 0 becomes 9, and codes 6-9 become 6:

R1=BRAC(V3, 1=1, 2=2, 3=3, 4=4, 5=5, 0=9, 6-9=6)

or

```

R1=V3
IF V3 IN LIST(6-9) THEN R1=6
IF V3 EQ 0 THEN R1=9

```

Use whichever method you find easiest. Of the above examples, using the BRAC function is probably the most straightforward and understandable method. If only one value of a variable needs modification, however, specifying all the values (as you must when using BRAC) is cumbersome. Instead, you could effectively use an IF statement:

```
IF V3 EQ 0 THEN R1=9 ELSE R1=V3
```

Or, if the original value of the input variable is not needed:

```
IF V3 EQ 0 THEN V3=9
```

Multivariate Recoding

Multivariate recoding allows the V- or R-variables to be assigned values based on rules applied to the current values of two or more variables. For example, suppose the following table represents a desired bivariate recoding:

		V2 (Columns)					
		1	2	3	5	6	7
V1 (Rows)	1	1	2	5	6	4	4
	2	2	1	2	6	7	4
	3	1	1	1	2	4	4
	7	3	3	3	3	9	9

A RECODE TABLE statement is used to represent the above table, and a subsequent assignment statement makes use of it as follows:

```

TABLE 1, PAD=4, COLS 1-3, 5-7, ROWS 1(2,5,6), 2(1,2,6,7), 3(1,1,2),7(3,3,3,3,9,9)
R1=TABLE(TAB=1,V1,V2)

```

Bivariate recoding is also possible with the IF statement.

Arithmetic Computation. Arithmetic computations create new variables or replace old ones according to user-defined statements similar to algebraic formulas. For example, a variable may be expressed as a percentage of another variable, or the logarithm of a variable may be computed:

```

R1=V6/V23 * 100      (percentage)
R1=LOG(V6)            (logarithm)

```

You must make sure your arithmetic computations are possible and meaningful. Thus, in the above examples, if V23 could have a value of zero, the computation is better expressed as:

```
IF V23 EQ 0 THEN R1=0 ELSE R1=V6/V23 * 100
```

Further, if variable V6 had a first missing-data code, the second computation could be:

```
IF MDATA(V6) THEN R2=MD1(R2) ELSE R2=LOG(V6)
```

Evaluation of Expressions. Some RECODE statements involve arithmetic or logical expressions. Parentheses may be used to determine the order in which the operations in an expression are evaluated, e.g., $(3*(4+2))$. If there is no such indication, there is an implied order among the operations, called the precedence order. For example, the precedence order of arithmetic operations causes $3 * 4 + 2$ to equal 14 rather than 18. RECODE evaluates expressions from left to right and with the following order of operations:

Precedence order of arithmetic operators:

- Expression in parentheses
- Function references
- EXP (exponentiation)
- (negation)
- / and * (division and multiplication)
- + and - (addition and subtraction)

No two arithmetic operators in an expression may be contiguous. Use parentheses to separate operators when necessary; e.g., "...+Vn*-k+...", should be "...+Vn*(-k)+...".

Precedence order of relational and logical operators:

- Expression in parentheses
- Logical function references
- LT, LE, GT, GE, EQ, NE
- NOT
- AND
- OR

Alphabetic Recoding

You may recode alphabetic variables to numeric values with RECODE. For example, if variable V1 has codes ABCD and DEFG, the following statements could be used to convert them to the numeric values 1 and 2:

```
IF V5 EQ 'ABCD' THEN V5 = 1 AND GO TO LAST
IF V5 EQ 'DEFG' THEN V5 = 2
LAST      CONTINUE
```

Alphabetic values are enclosed in single quotes. They may be up to eight characters long; if a variable contains longer strings only the first eight characters are compared. Doubled single quotes will be replaced by a single quote. Thus "QUOTE " becomes 'QUOTE'=. You can check for uniqueness of the recoded values using LIST DATASET with the [Check Alphas option](#).

When alphabetic variables are referenced in RECODE, MicroSiris assumes they are used to create new non-alphabetic variables or be recoded to numeric values for use in statistical analysis. Attempting to use for analysis an unrecoded alphabetic value in RECODE leads to unpredictable results. However, you may recode the first eight characters of an alphabetic variable to a string of eight different characters, using the TRANSFORM command if you don't change the mode (type) of the variable. The fundamental concept of alphabetic recoding is:

If the output mode remains alphabetic, then you may recode only to alphabetic values; otherwise, you may only recode to numerical values.

To illustrate this, consider the following setup:

```
RECODE
  IF V5 EQ 'ABCD' THEN R1=1
  IF V5 EQ 'ABCD' THEN V5 = 'WXYZ'
END
LIST DATASET DATASET=DATA
JOB TITLE
RECODE 1 V=R1,V5
```

In this case, LIST DATASET prints what you expect: 'WXYZ' for V5 and 1 for R1. But if V5 is set to 1 and R1 set to 'WXYZ', you get missing data or garbage for both variables because LIST DATASET expects V5 to be alphabetic as indicated in the dictionary and R1 to be numeric. The same recode used in USTATS produces garbage for V5, since USTATS assumes you recoded any alphabetic variables to numeric, or you wouldn't be trying to numerically analyze them. With USTATS, unlike LIST DATASET, setting V5 to 1 therefore works.

Aggregation

Aggregation allows you to process groups of cases as the unit of analysis rather than the cases individually. This is done by arithmetically combining data values across the cases in any group, e.g., computing the sum, mean, or standard deviation of the values of the variables of interest across records in the group. In the following example, assume that data was collected at a county level for a group of states, but that the desired unit of analysis is the state. Thus, the data must be aggregated to the state level.

Assume you want to sum a single item of data, V2, and that V1 is the state code in the input records. The input data is sorted by states.

Clearly, V2 should be summed until the state code changes, and one record should be sent to the program using the recode each time the state code changes. But there are two special concerns: you don't want to send a meaningless record at the time the first state code appears (i.e., the first record), and you don't want to lose the record for the last state at end-of-file. The following code illustrates the way to deal with these special cases and accomplish aggregation

```
      CARRY (R1,R2)
&
      IF R1 NE V1 OR EOF THEN GO TO L1      !Check for new state
&
&      SUM V2 FOR EACH RECORD IN THE CURRENT STATE
&
      R2=R2+V2
      REJECT
&
&      SEND STATE AND SUMMATION VARIABLE TO THE COMMAND
&      AND RE-INITIALIZE
&
```

```

L1      R100=R1
        R200=R2
        R1=V1
        R2=V2
        IF R100 EQ 0 AND R200 EQ 0 THEN REJECT

```

We use the CARRY statement to ensure that the variables R1 and R2 are not destroyed each time a new case is read. It initializes the variables R1 and R2 to zero. Whenever a new state appears (i.e., R1, the old state, not equal to V1, the new state), a new path must be established. Statement two does this. Note that end-of-file (the EOF function) must be treated as a new state.

If the path to output an old state and initialize a new state is not taken, the current value of V2 is added to the previous sum R2 and the case is rejected. REJECT tells RECODE to get a new case without sending the current one to the command using the data.

L1 marks the output path. First, the old state code R1 is loaded into variable R100 and the summation variable into variable R200. The R1 and R2 are initialized for the new state. Finally, R100 and R200 are sent to the command using the data, but only if the state currently being processed is not the phantom state 0, which occurs when the first case is read and both R1 and R2 are zero.

Interactive Use

When RECODE detects an error, it prints an explanatory message and then asks you for a replacement statement. Thus the complete recode need not be entered again in its entirety. However, some errors, such as missing labels, are undetectable until the complete RECODE setup is entered. Therefore, **recode statements should be stored in a file** so that the Runfile assignment on the command prompt screen may be used (see [RUNFILES](#)). When errors are detected, they may be quickly corrected in the file and the RECODE command issued again.

Testing Recode Statements

Errors in logic not detectable by RECODE can occur. To check the intended results against those generated by RECODE, use the statements with LIST DATASET as follows:

- Filter the data (perhaps on an ID variable) so only a few cases or only critical cases are used.
- Specify in the variable list only the recoded variables and the variables used in their derivation.

The data values for the variables in question can thus be inspected.

RECODE and Filtering

Filtering takes place prior to recoding, so result variables may not be referenced in a filter, and input variables are filtered according to their original values. Use the REJECT statement to accomplish filtering with result variables and input variables that are transformed by RECODE.

Coding RECODE Statements

Statement Format

RECODE statements are format free. Labels must begin in column 1. Unlabeled statements begin in column 2 or beyond.

Field Definition

Label

Labels (1 to 4 characters), if used, must begin in column 1. Labels may be a number or any set of characters except blanks, the letters HELP or the letters STOP, and must not begin with an ampersand or a question mark. Labels allow control statements to refer to specific statements, e.g., "GO TO CALC." A label cannot be given on a LABEL, NAME, MDATA, or TABLE definition statement. The label is separated from the statement by one or more blanks.

Statement

The RECODE statement is entered starting in column 2 and may be spanned over multiple lines as noted below. Blanks may appear anywhere.

Summary of Available Statements and Functions

Assignment Statements

Statements that assign a value to a variable.

Statement	Purpose	Example
<u>ASSIGNMENT</u>	Assign a value to a variable.	R10=V9*V10
<u>DUMMY¹</u>	Assign a value of 0 or 1, depending on the value of a specified variable, to produce a series of "dummy" variables.	DUMMY R11-R13 USING V8(1-4)(5) -(0, 8) ELSE 99
<u>SELECT</u>	Assign a value to a variable selected from a list of variables by the value of a specified variable.	SELECT(FROR1-R5, BY=R99)=1

¹ May not appear in the new or ELSE clause of an IF statement.

Control Statements

Statements that control the order in which other recode statements are processed.

Statement	Purpose	Example
<u>BRANCH</u>	Jump to one of several specified labeled statements, depending on the value of a specified variable.	BRANCH(V2, A, B, C)

<u>CONTINUE</u> ¹	Continue with the next statement.	A CONTINUE
<u>END</u> ²	Designate the end of a recode definition.	END
<u>ENDFILE</u>	Close the input dataset as if an end-of-file was reached.	IF V1 GE 100 THEN ENDFILE
<u>ERROR</u>	Terminate the command using the recode definition a message telling where the "error" occurred.	IF V2 NOT IN(1-3) THEN ERROR
<u>GO TO</u>	Continue with the specified labeled statement.	IF V3 EQ 5 THEN GO TO B
<u>IF</u> ¹	Cause one or more statements to be executed if specified conditions are true or, optionally, other statements to be executed if they are not.	IF MDATA (V10,V11) THEN R5=99 ELSE R5=V10/V11
<u>RECODE</u> ³	Designate the beginning of and to assign a number to a recode definition.	RECODE 5
<u>REJECT</u>	Reject the current case (i.e., do not pass it to the command using the recode definition) and continue processing with the next case at the beginning of the recode definition.	IF R1 GT 5 THEN REJECT
<u>RELEASE</u>	"Release" the current case (i.e., pass it to the command using the recode definition) and continue processing with the same case at the beginning of the recode definition.	IF R2 LT 2 THEN RELEASE
<u>RETURN</u>	Return the current case (pass it immediately to the command using the recode definition without processing the remainder of the recode statements) and continue processing with the next case at the beginning of the recode definition.	IF R3 IN(1-3) THEN RETURN

1 May not appear in the THEN or ELSE clause of an IF statement

2 May only be the very last statement in a set of recode statements.

3 May only be the first statement in a set of recode statements.

Definition Statements'

Statements that set variable characteristics or provide definitions used by other recode statements.

Statement	Purpose	Example
<u>CARRY</u>	Define one or more R-type variables as "carry " variables, which retain their values from case to case.	CARRY (R1-R7, R10)

<u>LABEL</u>	Assigns or changes variable value labels for variables used in RECODE	LABEL R21'0=no,2=yes'
<u>NAME</u>	Assign names to R-type and/or V-type variables.	NAME R1 'BKT AGE OF SPOUSE' NAME V1 'YEARS OF EDUCATION'
<u>TABLE²</u>	Define a table use with the TABLE function for bivariate or univariate recoding.	TABLE ABC, PAD=S , COLS 1,2,3, ROWS 1(3,5,1), 2(1,8,2) ENDTAB

1 May not appear in the THEN or ELSE clause of an IF statement

2 Must appear before the TABLE function reference.

Arithmetic and Alphanumeric Functions

Functions that yield a numeric value or an alphabetic or numeric value.

Function	Purpose	Example
<u>ABS</u>	Obtain the absolute value of a variable or arithmetic expression.	R21=ABS(V2-V3)
<u>BRAC'</u>	Perform complete univariate recoding by obtaining a value based on the value of a single variable according to a set of rules.	R22=BRAC(V5,TAB=1,ELSE=9,1-10=1,11-20=2) R23=BRAC(V5,TAB=2,ELSE='X',1-10='A',11-20='B')
<u>LOG</u>	Obtain the logarithm to base 10 of a variable or arithmetic expression.	R3=LOG(V2)+LOG(R5)
<u>MD1</u> <u>MD2</u>	Obtain the first or second missing-data value of a variable.	IF V3 EQ MD1(V3) THEN R2=MD2(R2) ELSE R2=ABS(V3)
<u>RAND</u>	Generate an integer random number in a specified range.	R2=RAND(0)
<u>RECODE</u>¹²	Perform partial or complete univariate, bivariate, or multivariate recoding by obtaining a value based on the values of one or more variables according to a set of rules.	R28=RECODE V7, (3)=1 ,6-9)=2,ELSE=0 R29=RECODE V7, V8, (1/1)=1, (2/2)=2, - (1/2)(2/1)=3, ELSE=0
<u>ROUND</u>	Obtain the rounded integer value of a variable or arithmetic expression.	R2=ROUIND(V1/12)
<u>SELECT</u>¹	Obtain the value of a variable or constant selected from a list of variables and constants, based on the value of a specified variable.	R23=SELECT(FROM R1-R5,9 BY V10) R24= SELECT(FROM 'M','F','DK' BY V10)
<u>SQRT</u>	Obtain the square root of a variable or arithmetic expression.	R2=SQRT(V2)

<u>TABLE</u>	Perform bivariate or univariate recoding by obtaining a value from a table defined by a TABLE statement, based on the values of two variables or constants.	R2=TABLE(5,V3,TAB=2,ELSE=9)
<u>TRUNC</u>	Obtain the truncated integer value of a variable or arithmetic expression.	R10=TRUNC(V26)

1 Alphanumeric--may return either a numeric or alphabetic value.

2 Must be used only on the right of a equal sign and only used in an assignment statement or the THEN or ELSE clause of an IF statement.

Logical Functions

Functions that yield a value of "true" or "false".

Function	Purpose	Example
<u>EOF</u>	Test whether the end-of-file is reached and, when it has, to allow an additional pass through the recode definition.	IF R1 NE V1 OR EOF THEN GO TO L1
<u>INLIST</u>	Test whether a variable or expression is equal to any of the values in a list.	IF V5 IN(2,4,6) THEN R1=1 ELSE R1=0
<u>MDATA</u>	Test whether any of the variables in a list have missing data.	IF MDATA(V3) THEN R8=MD1(R8) ELSE R8=LOG(V3)

Statement and Function Descriptions

ABS Function

This function returns the absolute value of a variable or arithmetic expression. The absolute value of a number is its numerical value without regard to sign; the number 2 is the absolute value of both +2 and -2.

Prototype: ABS(arg)

Arg is any arithmetic expression for which the absolute value is desired.

Example:

R5=ABS(V5)

R5 is assigned the absolute value of the input variable V5. Thus if variable 5 has the value -7 or +7, the value of R5 will be 7.

ASSIGNMENT Statement

The assignment statement is the main computational statement. It assigns a new value to a result or input variable.

Prototype: variable=expression

Variable is any input (Vn) or result (Rn) variable. The value of the variable is changed by the execution of RECODE and the associated command so that it no longer has its original value.

Expression is any arithmetic expression.

Missing-data values in the expression are not automatically detected, so the user must handle missing-data separately from the assignment statement.

Examples:

R10=5	!R10 is assigned the constant 5 as its value.
R5=2*V10	!R5 is assigned two times the value of V10.
R1=SQRT(V20)	!R1 is assigned the square root of the value of V20.
R10=MD1(V10)	R10 is assigned the first missing-data code of V10.

BRAC Function

BRAC returns a value derived from performing specified operations on a single variable.

Prototype: BRAC(var,TAB=i,ELSE=value,rule1,...,rule n)

Var Any V- or R-variable.

TAB=i Numbers the set of rules and the associated ELSE so that the same rules may be used again in another BRAC statement without being re-specified (optional).

ELSE=value Use value when the value of var cannot be found in the rules. If ELSE=value is not specified, BRAC returns 99 when the value can't be found.

rule 1,...,rule n

Rules defining the values to return depending on the value of var. The rules are expressed in the form: x=c, where x defines one code or a range of codes and c is the value to return when the value of var is in the range of the code(s) defined by x. The "x" may be ">m", "<m", "m", or "m1-m2" where m is any integer or floating-point constant. Commas are not permitted in a rule. Thus, the possible rules are:

>m=c	If the value of var is greater than m, return c;
<m=c	If the value of var is less than m, return c;
m=c	If the value of var is equal to m, return c;
m1-m2=c	If the value of var is in the range m1 to m2, i.e., $m1 \leq \text{var} \leq m2$, return c.

As many rules may be given as necessary. They are evaluated from left to right, and the first one that is satisfied is used. **Note that ">" and "<" are used, rather than GT and LT logical operators.** This is the only place in RECODE these operands are used.

ELSE, TAB, and the rules may be specified in any order.

Example:

```
R1=BRAC(V10,TAB=1,ELSE=9,1-10=1,11-20=2)
R2=V1+BRAC(V2,TAB=1)*3
```

The value of R1 will be 1 if V10 is 1 to 10 and 2 if V10 is greater than or equal to 11 but not greater than 20. If V10 has any other value the ELSE clause applies and R1 becomes 9. V2 is bracketed by the same rules and R2 is set to V1 + (the bracket result times 3).

BRANCH Function

To cause processing to continue with one of several specified labeled statements, depending on the value of a specified variable.

Prototype: BRANCH(v-num,label-1,label-2, ..., label-n)

v-num A variable. The value of this variable determines which of the labeled statements specified will be executed next. It must be in the range of 1 to the number of labels specified.

label-1,label-2,...label-n
 A list of one or more 1-4-character statement labels. Assignment or control statements with these labels must appear in the set of recode statements.

Normally, a set of recode statements is executed in order from the first to the last. The BRANCH statement may be used to change this. This is useful if certain sections of the recode definition are to be applied for only certain cases. For example, depending on the number of wage earners in the household, different calculations might need to be done to calculate total household income. These different calculations could be contained in separate sections of the set of recode statements. Which section was appropriate for the case being processed could be determined by the value of a variable indicating the number of wage earners. Control could then be transferred to the appropriate section with the BRANCH statement.

When the BRANCH statement is encountered, control is transferred to one of the specified labeled statements (label-1. . .label-n). The value of the variable v-num is used to determine which of the specified labeled statements is to be executed next. If the value of v-num is 1, the first statement will be executed next, if the value is 2, the second statement will be executed

next, etc. If the value of v-num is less than 1 or greater than the number of labels specified, an error will result and the command using the recode definition will terminate abnormally with an error message containing the sequential number of the case where processing stopped and the sequential number of the recode statement where the error occurred.

BRANCH may be used to transfer control to any labeled executable statement. Assignment statements and control statements, other than RECODE and END, are executable. Definition statements (CARRY, MDATA, NAME, LABEL, and TABLE) and the RECODE and END control statements are not.

WARNING: Take care that an infinite loop is not formed if control is transferred to a statement before the BRANCH statement.

Example:

```
          BRANCH (R999,YR1,YR1,YR3)
YR1      R1=V12*.203
          GO TO LAST
YR2      R1=V24*.448
          GOTO LAST
YR3      R1=V36*.808
LAST     CONTINUE
```

If R999 has the value 1, processing continues with the statement labeled YR1. If R999 has the value 2, processing continues with the statement labeled YR2. If R999 has the value 3, processing continues with the statement labeled YR3. If R999 has a value of less than 1 or greater than 3, an error results; and the command using the recode definition terminates abnormally.

CARRY Statement

The CARRY statement tells RECODE to retain the values from case to case for the listed variables. CARRY variables are set to zero before the first case is read and change only when explicitly set to something else. The CARRY variable can be used as counters or accumulators for aggregation.

Prototype: CARRY(var-1,var-2,var-3,...,var-n)

Var-1,var-2,var3,...,var-n The variables to initialize to zero and carry

CARRY cannot have a statement label

V-variables may not be used.

Example 1:

Select the first record of a group of records with the same ID (V1)

```
CARRY (R9)
IF R9 EQ V1 AND R9 NE 0 THEN REJECT
```

R9=V1

Example 2:

Select the last record of a group of records with the same ID (V1)

```
CARRY (R9,R1-R5,R11-R15)  ! R1-R5 and R11-R15 temporarily holds variables V1-V5
&    Save current record
    R1000=1
A0    SELECT(BY R1000 FROM R11-R15)=SELECT(BY R1000 FROM V1-V5)
    IF R1000 LT 5 THEN R1000=R1000+1 AND GO TO A0
L0    IF R9 NE V1 OR EOF THEN GO TO L1      !Check for new group
&    Save this group record
    R1000=1
A2    SELECT(BY R1000 FROM R1-R5)=SELECT(BY R1000 FROM V1-V5)
    IF R1000 LT 5 THEN R1000=R1000+1 AND GO TO A2
    REJECT
&    New group found. Reset and restore last record of group just finished
L1    R100=R9          ! R100 is used to check for initial (phantom), group
    R9=V1
    R1000=1
A3    SELECT(BY R1000 FROM V1-V5)=SELECT(BY R1000 FROM R1-R5)
    IF R1000 LT 5 THEN R1000=R1000+1 AND GO TO A3
&    Save new record
    R1000=1
A4    SELECT(BY R1000 FROM R1-R5)=SELECT(BY R1000 FROM R11-R15)
    IF R1000 LT 5 THEN R1000=R1000+1 AND GO TO A4
    IF R100 EQ 0 THEN REJECT      ! Reject the phantom initial state
```

COMBINE Function

Obtain a unique value for each combination of values of two or more variables.

Prototype: COMBINE v-num1(n1), v-num2(n2), ..., v-numn(nm)

v -num1 The first variable.

v-num2 The second variable.

v-numn	The last variable.
n1	The maximum code value of the first variable plus 1.
n2	The maximum code value of the second variable plus 1.
nm	The maximum code value of the last variable plus 1.

Up to 13 variables may be specified. The value of these variables in combination with others is combined to produce a unique value for each combination of values. Each variable must have positive integer values.

The COMBINE function returns a unique value for each combination of values of the variables specified. It is used to create a pattern (or combination) variable.

Such variables are useful in handling predictor variable interactions in multivariate analyses that assume additive models, such as the multiple classification analysis technique (see MCA), where interactions between variables often produce distorted or misleading results. If you discover that such interactions exist between two or more variables, you may create a new "combination" or "pattern" variable whose values reflect all possible combinations of the original variables. For example, in situations where the predictors "gender" and "age" might interact in their effects upon a dependent variable, a single combination variable could be created with the values "young female", "young male", "middle-aged female", "middle-aged male", "older female", and "older male" (see example 1).

You may determine the results of the function by inspecting the table of all possible combinations of values for the specified variables, which is printed when the recode statements are compiled. (See example 3) Alternatively, the values returned by the COMBINE function may be determined by the following formula:

$$val-vnum1 + (n1 \times val-vnum2) + (n1 \times n2 \times val-vnum3) + (n1 \times n2 \times n3 \times val-vnum(4), \text{ etc.})$$

where val-vnum1 is the value of the first variable, val-vnum2 is the value of the second variable, and so forth.

In order for the COMBINE function to return meaningful codes, each variable specified (v-num1, v-num2, . . . v-numn) must have positive integer values. Negative values or values with decimals will not cause an error but will cause the COMBINE function to return meaningless, possibly non-unique, values. Also, care must be taken to specify the maximum codes accurately. Otherwise, non-unique values will be generated. For example, if

COMBINE V11(3),V12(4),V13(2)

were specified, the function would return a value of 22 for the "legal" set of values if V11 had a value of 1, V12 had a value of 3, and V13 had a value of 1:

$$\begin{array}{rclclcl} V11 & + & (n1 \times V12) & + & (n1 \times n2 \times v13) & = \\ 1 & + & (3 \times 3) & + & (3 \times 4 \times 1) & = 22 \end{array}$$

However, it would also return a value of 22 for the "illegal" set of values if V11 had a value of 4 (over the maximum), V12 had a value of 2, and V13 had a value of 1:

$$V11 + (n1 \times V12) + (n1 \times n2 \times V13) =$$

$$4 + (3 \times 2) + (3 \times 4 \times 1) = 22$$

The COMBINE function may only be used in an ASSIGNMENT statement or in the THEN/ELSE clause statement. It must be the only expression on the right side of the equal sign--i.e., it may not be combined with other expressions by arithmetic operators.

Example 1:

R1=COMBINE V85(2), V52(3)

If V85 had two codes, 0 for female and 1 for male, and V52 had three codes, 0 for young, 1 for middle-age for older, R1 would be assigned one of six values based on the values of V85 and V52 as follows:

CODE	V85 gender	V52 Age	Name
0	0	0	young female
1	1	0	young male
2	0	1	middle-aged female
3	1	1	middle-aged male
4	0	2	older female
5	1	2	older male

Alternatively, the formula

$$\text{Val-vnum1} + (\text{n1} \times \text{val-vnum2})$$

may be used to determine the function return. Thus, if V85 had a value of 1 and V52 a value of 1, R1 would have a value of 3:

$$1 + (2 \times 1) = 3$$

Example 2:

R1=COMBINE V11(4), V12(3)

If V11 (v-num1) had values of 2 and 3, the value of n1 would need to be stated as 4 (one greater than maximum code value). This would allow for the values of 0, 1, 2, and 3, even though the values of 0 and 1 do not appear in the data. The "extra" codes may be avoided by first recoding the variables as needed by the BRAC function.

$$\begin{array}{rclcl} \text{V11} & + & (\text{n1} \times \text{V12}) & + & (\text{n1} \times \text{n2} \times \text{V13}) & = \\ 1 & + & (3 \times 3) & + & (3 \times 4 \times 1) & = 22 \end{array}$$

However, it would also return a value of 22 for the "illegal" set of values if V11 had a value of 4 (over the "legal" maximum), V12 had a value of 2, and V13 had a value of 1:

$$\begin{array}{rclcl} \text{V11} & + & (\text{n1} \times \text{V12}) & + & (\text{n1} \times \text{n2} \times \text{V13}) & = \\ 4 & + & (3 \times 2) & + & (3 \times 4 \times 1) & = 22 \end{array}$$

The COMBINE function may only be used in an ASSIGNMENT statement or in the THEN/ELSE clause of an IF statement. It must be the only expression on the right side of the equal sign--i.e., it may not be combined with other expressions by arithmetic operators.

Example 3:

R1=COMBINE V11(3), V12(4), V13(2)

V11 has three codes (0, 1, 2); V12, four codes (0, 1, 2, 3); and V13, two codes (0, 1). Thus, R1 would have 24 codes. The values of R1 can be determined by the formula

$$\text{val-vnum1} + (\text{n1} \times \text{val-vnum2}) + (\text{n1} \times \text{n2} \times \text{val-vnum3})$$

Or, in this case, substituting the specified combine values and variables:

$$\text{V11} + (3 \times \text{V12}) + (3 \times 4 \times \text{V13})$$

Thus, if V11 is 2, V12 is 3, and V13 is 1, the value of R1 will be 23:

$$2 + (3 \times 3) + (3 \times 4 \times 1) = 23$$

And if V11 is 1, V12 is 2, and V13 is 0, the value of R1 becomes 7:

$$1 + (3 \times 2) + (3 \times 4 \times 0) = 7$$

Alternatively, you can use the table printed by &RECODE to determine the meaning of the value returned. A single column version of the following table would be printed for the recode statement in this example:

CODE	V11	V12	V13	CODE	V11	V12	V13
0	0	0	0	12	0	0	1
1	1	0	0	13	1	0	1
2	2	0	0	14	2	0	1
3	0	1	0	15	0	1	1
4	1	1	0	16	1	1	1
5	2	1	0	17	2	1	1
6	0	2	0	18	0	2	1
7	1	2	0	19	1	2	1
8	2	2	0	20	2	2	1
9	0	3	0	21	0	3	1
10	1	3	0	22	1	3	1
11	2	3	0	23	2	3	1

Example 4:

R1=COMBINE V11(10), V12(10), V13(10)

When the variables to be combined have integer values 0-9, using the COMBINE function is equivalent to creating a variable by multiplying each variable by a consecutively ascending power of 10. Calculating values according to the formula

$$\text{val-vnum1} + (\text{n1} \times \text{val-vnum2}) + (\text{n1} \times \text{n2} \times \text{val-vnum3})$$

the value of R1 would be

$$\text{V11} + (10 \times \text{V12}) + (10 \times 10 \times \text{V13})$$

Thus, if V11 is 8, V12 is 3, and V13 is 2, R1 would have a value of 238:

$$8+(10 \times 3)+(10 \times 10 \times 2)$$

The same results would be obtained by the recode statement:

```
R1=V11+V12*10+V13*100
```

CONTINUE Statement

The CONTINUE statement performs no operation. It is used as a convenient transfer point.

Prototype: CONTINUE

Example:

```
IF V17 EQ 10 GO TO AT
R20=V11
GO TO THAT
AT R20=V11*100
THAT CONTINUE
```

CONTINUE may not be used in an IF statement.

DUMMY Statement

Assigns a value of 0 or 1, depending on the value of a specified variable, to a series of dummy variables.

Prototype: DUMMY v-list USING v-num (vals-1)(vals-2) . . . (vals-n) ELSE expression

v-list A list of dummy variables whose values are being assigned by the statement. The list may contain R- or V-type variables listed singly or in ranges, separated by commas (e.g., R1-R12 or V20,V10,V31-V35,R50). The order specified is preserved. Double references (e.g., R1,R3,R1) are valid. The maximum number of dummy variables that may be created in a single DUMMY statement is 100.

v-num A variable. The value of this variable is tested against the value lists(vals-1)(c-2) . . . (vals-n)----to set the values for the dummy variables.

(vals-1)(vals-2) . . . (vals-n) Lists of values. The values for each list are enclosed in parentheses. There must be the same number of lists as dummy variables specified in v-list. The values may be specified singly or in ranges, separated by commas--e.g., (1,3,5) or (1-5,12,18-25). No value should be in more than one list. These lists are used to set the values of the dummy variables.

ELSE expression

Specifies the value to use as the value for all dummy variables specified in v-list if the value of the variable v-num is not in one of the value lists---(vals-1)(vals-2) ... (vals-n). The value is specified in expression, which may be any arithmetic expression. *Default: 0.*

The DUMMY statement creates a set of variables from variable v-num. These variables are assigned a value of 0 or 1, based on the value of v-num. The value of the variable v-num is compared to the values specified in the lists of values (vals-1)(vals-2) ... (vals-n). If the variable v-num has a value in the first list of values, the first dummy variable is set to 1, the others to 0, and no further checking is done; if the variable v-num has a value in the second list of values, the second dummy variable is set to 1, the others to 0, and no further checking is done, etc. If the variable v-num has a value that does not occur in any of the lists specified, all dummy variables are set to the value specified by the ELSE clause, which, if not specified, defaults to 0.

The DUMMY statement may not be used in an IF statement

Example 1:

```
DUMMY R1-R3 USING V8(1-4)(5,7,9)(0,8) ELSE 99
```

The following chart shows how the values R1-R3 would be assigned in the above example:

V8	R1	R2	R3
0	0	0	1
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	0	1	0
6	99	99	99
7	0	1	0
8	0	0	1
9	0	1	0
other values	99	99	99

Thus, if V8 has a value of 1, R1 will be assigned a value of 1 and R2 and R3 will be assigned values of 0.

Example 2:

```
DUMMY R1,R1,R2 USING V3(1)(2)(3)
```

The following chart shows how the values of R1-R2 would be assigned in the above example:

V3	R1	R2
1	1	0
2	1	0
3	0	1
4	0	0

The following statement would produce exactly the same results:

DUMMY R1-R2 USING V3(1,2)(3)

END Statement

Each set of RECODE statements except the last must be terminated by an END statement beginning in column 2 or beyond. The last or only RECODE statement in a series is terminated in the conventional manner with an END statement beginning in column 1.

Prototype: END

Example:

```
RECODE
RECODE 1
R1=1
R2=2
END
&
RECODE 2
R22=V1+V2 R23=V3*10.
END
```

ENDFILE Statement

The ENDFILE statement causes RECODE to close the input dataset exactly as if an end-of-file had been reached.

Prototype: ENDFILE

Example:

```
IF V1 EQ 100 THEN ENDFILE
```

In this example, end-of-file is sent the first time V1 equals 100.

EOF Function

The EOF function is used for aggregation. The presence of the EOF function causes the recode statements to be passed (i.e., executed) once more after the end-of-file is encountered. The value of the EOF function is true during this after-end-of file pass of the recode statements and is false at all other times.

For the final pass through the recode statements, V-variables have the value they had after the last case was fully processed. Regular R-variables will be reinitialized to 1,500,000,000. CARRY

variables are not reinitialized, as usual. You must be careful to establish a correct path through the recode statements when end-of-file is reached.

Prototype: EOF

Example:

```
CARRY (R1,R2)
IF EOF THEN GO TO MEAN
IF MDATA(V1) THEN GO TO NEXT
R1=R1+1
R2=R2+V1
NEXT REJECT
MEAN R2=R2/R1
```

This example shows how to compute the mean of V1. CARRY is used to initialize R1 and R2 to zero. R1 is used to count the cases with valid data for V1 and R2 becomes the sum of V1 for all valid V1. REJECT causes a new case to be read and processed from the beginning of the RECODE statements (not including the CARRY statement) without returning to the command using the data. Missing-data are excluded with the MDATA function. When all cases are read, EOF becomes true. Only then is the last statement, labeled MEAN, executed to convert R2 into the mean and send it to the command using the data. See the RECODE section labeled [Aggregation](#) for a more complete example.

ERROR Statement

The ERROR statement directs RECODE to terminate the command reading the data with an error message indicating the case number and the number of the RECODE statement at which the error occurred.

Prototype: ERROR

Example:

```
IF R6 EQ 2 THEN ERROR
```

GO TO Statement

The GO TO statement is used to change the sequence in which the statements are executed. In the absence of a GO TO statement, statements are executed sequentially according to their physical order.

Prototype: GO TO label

Label is a 1- to 4-character label. The statement identified by the label may be physically before or after the GO TO statement.

Warning: Be careful of referencing a statement entered before the GO TO, as infinite loops can thus be formed.

(LABEL, MDATA, NAME, and TABLE statements may not have labels.)

Example:

```
IF V1 LT 9 GO TO TOWN
R10=999
GO TO MORE
TOWN R10=R5+V11
MORE ...
```

IF Statement

The IF statement allows conditional assignment and transfer. The keyword IF is the first element of the statement text.

Prototype: IF test THEN stmt1 AND stmt2 AND ... Stmtn ELSE stmtn+1 AND stmtn+2 AND ... Stmtm

Test Any combination of logical expressions (including logical functions) separated by AND or OR and optionally preceded by NOT. Test may be enclosed in parentheses.

Stmt1,...,stmtn Any assignment or control statement except CONTINUE, including those using the BRAC or TABLE functions.

The statements between THEN and ELSE are executed if the test is true.

The statements after the ELSE are executed if the test is false. If no ELSE clause is present, the next statement is executed.

The THEN and ELSE may each be followed by any number of statements, each separated by AND.

Examples:

```
IF V5 EQ V6 THEN R1=1 ELSE R1=2
```

Set R1 to 1 if the value of V5 equals the value of V6; otherwise set R1 to 2.

```
IF MDATA(V7,V10-V12) THEN R6=MD1(V7) AND R10=99 ELSE R6=V7+V10+V11 AND -
R10=V12*V7
```

Set R6 to V7's first missing-data code and R10 to 99 if any of the variables V7, V10, V11, V12 are equal to their missing-data codes. Otherwise set R6 equal to the sum of V7, V10 and V11, and set R10 equal to the product of V12 and V7.

```
IF (V5 NE 7 AND R8 EQ 9) THEN V3=1 ELSE V3=0
```

Set V3 to 1 if both V5 is not equal to 7 and R8 is equal to 9. (Parentheses are not required.)

INLIST Function

The INLIST function returns a value of "true" if an expression evaluates to any of a set of values passed to the subroutine. If the expression equals a value outside the set of values, the function returns a value of "false."

Prototype: expression INLIST(values)

Expression Any arithmetic expression.

Values A list of values or ranges of values.

IN is a valid abbreviation for INLIST.

Examples:

```
IF R12 INLIST(1-5,9,10) THEN V5=0
```

If R12 has a value of 1, 2, 3, 4, 5, 9, or 10, the INLIST function returns a value of "true," and V5 is set to 0.

```
IF V3+V7 IN(2, 4, 5, 6) THEN R1=1
```

If the sum of input variables V3 and V7 results in the value 2, 4, 5, or 6, then INLIST returns a value of "true" and result variable R1 is set to 1. Otherwise, INLIST returns a value of "false," and R1 retains its former value.

LABEL Statement

A LABEL statement assigns or changes variable value labels for variables used in RECODE.

Prototype: LABEL v1'code 1,label 1,...label n',V2'label'code 1,label 1,...label n',...

v1, v2,...,vn R- or V-type variables or lists of variables.

Labe 1, label 2, ...,label n

Value labels to assign code 1,...code n.

The labels can be at most 16 characters and can include blanks; if longer, they are truncated. The total length of all labels for a given variable can be no longer than 100 characters. If the R-type variables in a list do not appear elsewhere in the RECODE, they are ignored.

To include an apostrophe in a label, use double primes (e.g., 'Person"s').

A LABEL statement can't have a label.

Example:


```
LABEL R21'0=no,2=yes,3=maybe',V2'1=Female,2=Male,9=Unknown'
```

LOG Function

The LOG function returns the logarithm to the base 10 of the argument passed to the function.

Prototype: LOG(arg)

Arg is any arithmetic expression for which the log is desired.

Example:

```
R10=LOG(V30)
```

R10 is assigned the log of V30.

Application:

The logarithm of any number X to any other base B can be found by the following simple transformation:

$$R1 = \text{LOG}(x) / \text{LOG}(b)$$

For the natural logarithm (base e) this becomes simply

$$R1 = 2.302585093 * \text{LOG}(x)$$

Thus $R1 = 2.302585093 * \text{LOG}(V30)$ assigns R1 the natural logarithm of V30.

MDATA Function

Returns a logical value of "true" if any of the variables specified have a missing-data value; otherwise, the function returns a value of "false." The MDATA function provides a convenient way to check for missing-data, since this is not automatically done in the computation phases of RECODE.

Prototype: MDATA(vars)

Vars is a list of variables and variable ranges.

Example:

```
IF MDATA(V1,V5-V7) THEN R1=MD1(R1) ELSE R1=V1+V5+V6+V7
```

If any variable in the list V1, V5, V6, V7 has a value equal to its first missing-data code or in the range specified by its second missing-data code, the MDATA function returns a value of "true," and R1 is set to its first missing-data code. Otherwise, the MDATA function returns a value of "false" and R1 is set to equal the sum of V1, V5, V6 and V7.

MDATA Statement

The MDATA statement assigns missing-data codes to R-type and V-type variables. They are applied at the beginning of the recode regardless of where they appear in the set of recode statements. If a variable appears in more than one MDATA statement, the last occurrence to appear in the recode statements is used. You may also use it to assign new missing-data codes to V-type variables. . If the R-type variables in a list do not appear elsewhere in the RECODE, they are ignored.

Prototype: MDATA v-list1 (mdl,md2), v-list2 (mdl,md2), , v-listn (mdl,md2)

v-list1,v-list2...v-listn

Variable lists. Both V-type and R-type variables may be specified. The list may be a single variable (R50), a range of variables (R50-R59), or a combination of single variables and variable ranges (R50-R59, V35, V40-V45). The variables specified are assigned the designated missing-data codes.

mdl A value to use as the first missing-data code for the specified variables. This value will be treated as missing data.

md2 A value use as the second missing-data code for the specified variables. This value and any larger values (or any smaller values if md2 is negative) will be treated as missing data.

Each set of variables must be followed by the mdl and md2 values to use for that set of variables. Either mdl or md2 may be omitted: if mdl is omitted, a comma must precede the md2 value.

If mdl is omitted, the variables will retain their first missing-data code as specified in the dictionary for V-type variables and 1,500,000,000 for R-type variables.

If md2 is omitted, the variables will retain their second missing-data code as specified in the dictionary for V-type variables and 1,600,000,000 for R-type variables.

An MDATA statement cannot have a statement label.

Missing-data codes may be set only to integral values--fractional values are truncated.

Missing-data values assigned by the MDATA statement are treated as missing data by any commands that use the recode definition. They are not stored permanently in an Microsiris dictionary unless the recode definition is used with a command that creates an output dataset, such as TRANSFORM.

Examples:

```
MDATA V2(0,9)
```

The first missing-data code for V2 will be 0; the second will be 9.

```
MDATA R10-R15(8)
```

The first missing-data code for R10 through R15 will be 8; the second will be the default second missing-data code of 1,600,000,000.

```
MDATA R16(,99), V6(9)
```

The first missing-data code for R16 will be the default first missing-data code of 1,600,000,000; the second will be 99. The first missing-data code for V6 will be 9; the second will be the value specified in the dictionary.

MD1, MD2 Function

The MD1 (or MD2) function returns a value that is the first (or second) missing-data code of the variable passed to the function.

Prototype: MD1(var)

MD2(var)

Var is any V- or R-variable.

Example:

```
R12=MD2(V20)
```

R12 is assigned the second missing-data code for V20.

NAME Statement

A NAME statement assigns or changes names of variables used in RECODE. They are applied at the beginning of the recode regardless of where they appear in the set of recode statements. If a variable appears in more than one NAME statement, the last occurrence to appear in the recode statements is used.

Prototype: NAME v1'name-1', v2'name-2', ..., vn'name-n'

v1, v2,...,vn R- or V-type variables.

name-1, name-2, ..., name-n

Names to assign these variables. The names can be at most 24 characters; if longer, they will be truncated. The default name for an R-type variable, if not specified in a NAME statement, is "Recoded Variable".

To include an apostrophe in a name, use double primes (e.g., 'Person"s').

A NAME statement can't have a label.

Example:

```
NAME R1'Combined SAT Scores', V2'Age Bracket'
```

NDEC Statement

NDEC statements assign the number of display decimal places to V- or R-type variables. They are applied at the beginning of the recode regardless of where they appear in the set of recode statements. If a variable appears in more than one NDEC statement, the last occurrence to appear in the recode statements is used.

Prototype: NDEC v-list1 (n1), v-list2 (n2), , v-listn (n3). DEFAULT(ndef)

v-list1,v-list2,...,v-listn

A variable list. The list may be a single variable (R50), a range of variables (R50-R59), or a combination of single variables and variable ranges (R50-R59, R45). The variables specified are assigned the designated number of decimal places. If the R-type variables in a list do not appear elsewhere in the RECODE, they are ignored.

ndef

The value to use for all R variables not appearing in this or any other NDEC statement.

Decimal places can only be set in the range 0-15. A number larger than 15 is set to 15 and a number less than 0 is set to 0. You may use V-type variables in an NDEC statement, but note that if the variable is not stored in floating-point but as character or integer data, the number n becomes an implicit decimal point, which has the effect of scaling down the value.

NDEC values assigned by the NDEC statement are not stored permanently in a MicroSiris dictionary unless the recode definition is used with a command that creates an output dataset, such as TRANSFORM.

Example:

```
NDEC DEFAULT(5),V1(7), R1,R2(3)
```

RAND Function

The RAND function returns a value that is a uniformly distributed random number between 0 and 1, based upon the argument "seed."

Prototype: RAND(seed)

Seed is an integer constant that is used to initiate the random sequence. If seed is 0, then a value based on the time of day the RECODE command decodes the statement is used.

Example:

```
R1=RAND(0)
```

R1 is set equal to a random number, uniformly distributed between 0 and 1. The sequence is initialized from the time of day RECODE decodes the statement.

Application:

The RAND function can be used to select a random subset of data or to generate more complex distributions. To randomly select 5% of a dataset, use:

```
IF RAND(0) GT .05 THEN REJECT
```

If the same subset is to be used more than once, use the same non-zero seed value each time. If you want a different range, e.g., a random number uniformly distributed from 1 to 10, multiply and add constants as appropriate:

```
R1=RAND(0)*9 + 1
```

There are two relatively simple methods to generate normally distributed random numbers. The first generates a single random variable:

```
R1=RAND(0)
IF R1 LT .5 THEN R2=-1 ELSE R2=1 AND R1=1-R1
R3=SQRT(-2.302585*LOG(R1*R1))
R4=R3*R3
R1=R2*(R3-((2.515517+.802853*R3+.010328*R4)/1+R3*1.432788+.189269*R4 -
+.001308*R3*R4)))
```

In this method, a uniformly distributed random number between 0 and 1 is transformed via Taylor series expansion to a normally distributed random variable with mean 0 and standard deviation 1.

The second method is called the *polar method* (Knuth, Vol. 2, p. 104) and is used when two independent normally distributed random variables are required. The method requires two independent random variables uniformly distributed between -1 and +1 and is as follows:

```
ONE  R1=RAND(0)*2 - 1
      R2=RAND(1234567)*2 - 1
      R3=R1*R1+R2*R2
      IF R3 GE 1 OR R3 EQ 0 THEN GO TO ONE
      R3=SQRT(-4.60517*LOG(R3)/R3)
      R1=R1*R3
      R2=R2*R3
```

In the example, the second call to RAND uses 1234567 as the seed; this was arbitrarily chosen. R1 and R2 are the two independent, normally distributed random variables.

RECODE Function

The recode function performs partial or complete univariate, bivariate, or multivariate recoding by obtaining a value based on the values of one or more variables according to a set of rules.

Prototype: RECODE v-list, TAB =label ELSE=expression, (rule-1)=value-1, (rule-2)=value-2, ... (rule-n)=value-n

```
RECODE v-list TAB =label
```

v-list	A list of up to twelve variables. The values of these variables will be used to determine the value the function returns.
TAB=label	Optional. Specifies a 1-4-character string that is used as a label for a set of RECODE rules so that the same set of rules may be used more than once without being respecified.
ELSE=expression	Optional. Specifies the value to return when the values of v-list are not found in the rules specified. While expression is typically a constant (e.g., ELSE= 9), it may be any arithmetic (or alphabetic) expression (e.g., ELSE= V5*V6 + 10) except that it may not include a reference to a RECODE or COMBINE function. If expression is a value for an alphabetic variable, it must be enclosed in primes (e.g., ELSE= 'NA').
Rules	Optional. A set of rules specifying values to compare with the variables in v-list and corresponding values the function returns. You may specify as many rules as necessary.

The rules are expressed in the form (list-1)(list-2) . . . (list-n)=value

where (list-1)(list-2) . . . (list-n) specify lists of values to compare with the values of the variables in v-list and value is the value the function will return. Alphabetic values must be enclosed in primes--e.g., ('M') = 25.

Specify as many lists as you wish in a single rule. Each list of values must be in the form:

(values-1/values-2/ ./values-n)

where values-1 is the set of values to compare with the value of the first variable in v-list. values-2 is the set of values to compare with the value of the second variable in v-list etc.

The number of sets of values specified must equal the number of variables in v-list. For instance, if two variables are specified in v-list two sets of values must be specified--e.g., (1-3/2) = 1.

A set of values may contain single values and/or a list of single values and/or ranges of values--e.g., (52/1,3,5/1-3,7). If v-list contains more than one variable, the sets of values are separated by a slash.

Rules may not be specified if TAB=label references a previously labeled set of rules.

The RECODE function is used for partial or complete univariate, bivariate, or multivariate recoding. It returns a value based on the values of the variable or variables and the set of rules specified. Up to twelve variables may be specified. The value returned is determined as follows:

If the values of the variables in v-list match the values or fall within the range of the values specified in one of the rules, the value specified by the rule is returned. The rules are evaluated from left to right and the first one that is satisfied is used. Therefore, they should be specified in the order that you want them applied.

If the values of the variables in v-list do not match the values or fall within the range of the values specified in any of the rules, the value specified by ELSE is returned. If no ELSE value is specified the first missing-data code value is returned.

You may use a set of RECODE rules more than once by labeling them with TAB=label when they are initially specified. These rules will then be "remembered" and may be referenced by the specification of TAB=label. Note that ELSE=expression will not be "remembered" and must be explicitly specified in subsequent references (see example 3). ELSE, TAB, and the rules may be specified in any order.

The RECODE function may only be used in an ASSIGNMENT statement or in a THEN/ELSE clause in an IF statement. It must be the only expression on the right-hand side of the equal sign; i.e., it may not be combined with other expressions by arithmetic operators.

Examples:

```
V76=RECODE V76, (97-99)=0
```

If V76 has a value in the range of 97 to 99, the value of V76 will be changed to 0. If V76 has any other value, the value of V76 will be unchanged.

```
R1=RECODE V21,V22 (3/5)(7/8)=1, (6-8/1-6)=99
```

V20 will be assigned a value based on the values of V21 and V22:

If V21 is 3 and V22 is 5 or if V21 is 7 and V22 is 8, V20 is assigned a value of 1. If V21 is in the range of 6-9 and V22 is in the range of 1-6, V21 is assigned a value of 2.

Otherwise, V20 retains its original value.

```
V20=RECODE V21,V22,TAB=ABC,ELSE=MD1(V20), (3/5)(7/8)=1, (6-9/1-6)=2  
V30=RECODE V31,V32,TAB=AC,ELSE=MD1(V30)  
MDATA V20,V30 (9)
```

V20 is assigned a value in the same manner as in the preceding example, except that V20 will be set equal to its MD1 value when the rules are not met. The specification of TAB=ABC allows the same RECODE rules to be referenced subsequently. V30 is assigned a value based on the values of V31 and V32 according to the rules defined in the previous statement. If the rules are not met, V30 is set equal to its MD1 value.

RECODE Statement

The RECODE statement assigns a number to the set of recode statements that follow, up to an END statement. Multiple recodes may be performed by assigning each one a new number. If no number is assigned, RECODE defaults to number one. When any command is to use a recode, you specify which one via the RECODE keyword in the command setup.

Prototype: RECODE n

Examples:

```
RECODE 10  
R1=V1*100  
END
```

Each set of RECODE statements except the last must be terminated by an END statement beginning in column 2 or beyond. The last or only RECODE statement in a series is terminated in the conventional manner with an END statement beginning in column 1.

```
RECODE 1  
R1=1  
R2=2  
END  
RECODE 2  
R22=V1+V2 R23=V3*10.  
END
```

REJECT Statement

The REJECT statement directs RECODE to reject the present case and obtain another case. The new case is then processed from the beginning of recode statements. Thus, REJECT can be used as a filter with recoded variables.

Prototype: REJECT

Example:

```
IF MDATA(V8,V12-V13) THEN REJECT
```

RELEASE Statement

The RELEASE statement directs RECODE to release the present case for processing by the command using the recode and to regain control after processing without reading another case. The same case is then processed from the beginning of recode statements. RELEASE can be used to break up a single case into several cases for the analysis-- a way of "disaggregation."

Take care when using the RELEASE statement that processing does not continue indefinitely.

Prototype: RELEASE

Example:

```
CARRY (R1)  
R1=R1+1  
IF R1 LT V1 THEN RELEASE ELSE R1=0 AND REJECT
```

If the value of V1 is a positive number, the case will be passed to the command using the recode definition V1 times. If the value of V1 is zero or a negative number, the case will be

rejected--not passed to the command. R1, a carry variable is used to count the number of times the case has been processed and released. CARRY variables have a value of zero when the first case is processed for the first time. When the value of R1 exceeds that of V1, R1 is reset to 0, the case is rejected and processing continues with the next case.

RETURN Statement

The RETURN statement directs RECODE to return control to the command using the recode. The next call to RECODE will process the next case. This is equivalent to using a GO TO statement to transfer control to a CONTINUE statement which is the last statement in the RECODE set.

Prototype: RETURN

Example:

```
IF V8 ge 12 THEN RETURN
```

ROUND Function

Obtains the rounded integer value of a variable or arithmetic expression by adding .5 to the value and truncating the result to the next lowest integer. Due to the limitations of machine precision, the results of division may be very near to, but not exactly, a value. Any value whose fractional or decimal part cannot be simply expressed as a fraction whose denominator is a power of 2 (e.g., halves, quarters, eighths, etc.) cannot be accurately stored. In some circumstances this lack of precision may become apparent--for instance is a value that is a result of a division is checked for equality to a value not derived in the same manner. The ROUND function only needs to be used in these rare circumstances.

Prototype: ROUND(expression)

expression is any arithmetic expression for rounding is desired..

Example:

```
IF (ROUND(V5/12*12) EQ V5 THEN R1=1 ELSE R1=0
```

If V5 has an integer value, using the ROUND function, R1 will have a value of 1. If ROUND were not used, and V5 had integer values in the range 1 to 6, R1 would have a value of 1 when V5 had a value of 3 or 6, would have a value of 0 when v5 had a value of 1,2,4, or 5.

SELECT Function

To obtain the value of a variable or constant selected from a list of variables and/or constants based on the value of a specified variable. See also SELECT statement.

Prototype: SELECT (FROM list BY v-num)

FROM list Specifies a list of variables and/or constants. One of these values will be the value returned by the function. Alphabetic constants must be enclosed in primes (e.g., FROM 'MI','WI','NY','CA').

BY Vnum Specifies the variable whose value will be used to select the value to return from the FROM list. The value of the variable v-num should be an integer in the range of 1 to the number of variables/constants in the FROM list.

The SELECT function returns the value of the variable or constant in the FROM list that is in the position designated by the value of the BY variable. For example, if the value of the BY variable is 2, the value of the second variable or constant in the FROM list will be returned. FROM and BY may be specified in any order.

If the value of the BY variable is less than 1 or greater than the number of variables or constants in the FROM list, an error will result; and the command using the recode definition will terminate abnormally with a message containing the sequential number of the case where processing stopped and the sequential number of the recode statement where the error occurred. If the value of the BY variable is not an integer, only the integer portion of the value will be used--e.g., if the BY variable had a value of 2.7, the value of the second variable or constant in the FROM list would be returned.

Examples:

```
R100=SELECT(FROM R1-R5,9 BY V2)
```

If the value of V2 is 1, R100 will be assigned the value of R1. If the value of V2 is 2, R100 will be assigned the value of R2, etc. If the value of V2 is 6, R100 will be assigned a value of 9. If the value of V2 is less than 1 or greater than 6 (the number of variables/values in the FROM list), an error will result and the command using the recode definition will terminate.

```
V50=SELECT(BY R1 FROM 1,3,5,R9)
```

If the value of R1 is 1, V50 will be assigned a value of 1. If the value of R1 is 2, V50 will be assigned a value of 3. If the value of R1 is 3, V50 will be assigned a value of 5. If the value of R1 is 4, V50 will be assigned the value of R9. If the value of R1 is less than 1 or greater than 4, an error will result and the command using the recode definition will terminate.

```
R999=1
A      SELECT(FROM V1-V10 BY R999)=SELECT(FROM R1-R10 BY R999)
       IF R999 LT 10 THEN R999=R999+1 AND GO TO A
```

In this example the SELECT function is used in a SELECT statement. As R999 is incremented from 1 to 10, V1 is assigned the value of R1, V2 is assigned the value of R2, etc., until the value of R999 becomes 10 when V10 is assigned the value of R10.

```
NAME R1 'Duplicates 0=No 1=Yes'
LABEL R1'0=no,1=yes'
```

	R1=0
	R999=1
	R998=2
A	IF SELECT (BY=R999, FROM=V1-V5) EQ SELECT(BY=R998, FROM=V1-V5) THEN - R1=1 AND GO TO B
	IF (R999 LT 4 AND R998 LT 5 THEN R998=R998+1 AND GO TO A
	IF (R999 LT 4 AND R998 EQ 5 THEN R999=R999+1 AND R998=R998+1 AND - GO TO A
B	CONTINUE

R1 is set equal to 1 if any code value appears more than once in V1-V5. To make all comparisons among the five variables, 10 comparisons, as diagrammed below, need to be made. R999 will be incremented from 1 to 4 while R998 will be incremented from 2 to 5:

		R999				
		1	2	3	4	5
R998	1					
	2	1				
	3	2	5			
	4	3	6	8		
	5	4	7	9	10	

The first time through the loop R999 has a value of 1 and R998 has a value of 2, V1 is compared to V2. If V1 equals V2, R1 is assigned a value of 1 and the loop is exited.

If V1 is not equal to V2, since R999 is less than 4 and R998 is less than 5, R998 will be incremented. R999 will still have a value of 1 and R998 will now have a value of 3. Thus V1 will be compared to V3.

If V1 equals V3, R1 is assigned a value of 1 and the loop is exited. If V1 is not equal to V3, R998 is incremented, so V1 will be checked against V4.

If V1 is not equal to V4, R998 will again be incremented and V1 will be checked against V5. If V1 is not equal to V5, R999 will be incremented to 2 and R998 will be assigned a value of 3. Then V2 will be checked against V3.

Checking will continue until two variables with the same values are found or until R999 has a value of 4 and R998 has a value of 5 and V4 is checked against V5.

SELECT Statement

To assign a value to a variable selected from a list of variables by the value of a specified variable.

Prototype: SELECT (FROM V-list BY v-num)=expression

FROM Specifies a list of variables from which one variable will be selected. The selected variable will be assigned the value of expression.

BY Vnum	Specifies the variable whose value will be used to select a variable from v-list. The value of the variable v-mcm should be an integer in the range of 1 to the number of the variables in the FROM list
expression	An alphabetic or arithmetic expression. The value of the expression will be assigned to the variable in v-list specified by the value of the variable v-num.

The SELECT statement assigns the value of the expression to the variable in the position in the FROM list designated by the value of the BY variable. For example, if the value of the BY variable is 3, the third variable in the FROM list will be assigned the value of the expression. Only the one designated variable in the FROM v-list is assigned a value; the other variables in the FROM list are unaltered. If the expression is an alphabetic expression, the variable selected from v-list should be an alphabetic variable. FROM and BY may be specified in any order.

If the value of the BY variable is less than 1 or greater than the number of variables in the FROM list, an error will result and the command using the recode definition will terminate abnormally with an error message containing the sequential number of the case where processing stopped and the sequential number of the recode statement where the error occurred. If the value of the BY variable is not an integer, only the integer portion of the value will be used--e.g., if the BY variable had a value of 2.7, the second variable in the FROM list would be assigned the value of the expression

Examples:

```
SELECT (BY V10 FROM V24,V12,V36)=R1*12
```

If V10 has a value of 1, V24 will be assigned the value of R1 x If V10 has a value of 2, v12 will be assigned the value of R1 x 12. If V10 has a value of 3, V36 will be assigned the value of R1 X 12. If V10 has a value of less than 1 or greater than 3, an error will result and the command using the recode definition will terminate abnormally.

```
SELECT(FROM R1-R12 BY R999
```

If R999 has the value of 1, R1 will be assigned the value of 0. If R999 has the value of 2, R2 will be assigned the value of 0, etc.

```
R999=1
A    SELECT(FROM R1-R12 BY R999)=0
      IF (R999 LT 12 THEN R999=R999+1 AND GO TO A
```

The 12 variables R1-R12 will each be set to 0 as R999 is incremented from 1 to 12. When R999 has the value of 1, R1 will be assigned the value of 0. When 8999 has the value of 12, R12 will be assigned the value of 0.

```
R1000=1
A    SELECT(BY R1000 FROM V101-V125)=SELECT(BY R1000 FROM V101-V125)*R1
      IF R1000 LT 25 THEN R1000=R1000+1 AND GO TO A
```

Using a SELECT function as part of the expression in the SELECT statement, as R9 is incremented from 1 to 25, the twenty-five variables V101-V125 are multiplied by the value of R1.

SQRT Function

The SQRT function returns a value that is the square root of the argument passed to the function. When using the SQRT function, be sure and check for negative values first, if necessary.

Prototype: SQRT(arg)

Arg any arithmetic expression.

Example:

```
IF V5 GT 0 THEN R5=SQRT(V5) ELSE R5=MD1(R5)
```

R5 is assigned the square root of V5 if it is positive, missing-data otherwise.

TABLE Function

The TABLE function returns a value based on the concurrent values of two variables, using a previously defined table (See [TABLE statement](#)).

Prototype: TABLE(r,c,TAB=name, ELSE=expression)

r a variable or constant to use as a "row index" in a previously defined table.

c a variable or constant to use as a "column index" in a previously defined table.

TAB=name specifies the 1-4 character name of a table previously defined by a TABLE statement.

ELSE=expression
 gives a value to use for pairs of values that are not defined in the table. The value of ELSE defaults to 99.

The TAB and ELSE specifications may be in any order.

Examples:

Assume that the following table was defined by a TABLE statement and named "1":

		Column					
		1	2	3	4	5	6
Row	2	1	1	2	2	3	4
	3	1	2	2	2	3	4
	5	2	2	2	2	3	4
	6	3	3	3	3	3	4
	8	9	9	9	9	9	9

```
R1=TABLE(V6,V4,TAB=1,ELSE=0)
```

If V6 equals 5 and V4 equals 3, then R1 is assigned the value 2 (the intersection of row 5 and column 3). If V6 equals 2 and V4 equals 6, then R1 is assigned the value 4 (the intersection of row 2 and column 6). If V6 equals 4 and V4 equals 2, then R1 is assigned the value 0 (row 4 is not defined; the ELSE value is used).

```
R1=TABLE(V1,V7,TAB=A,ELSE=-1)
```

A table previously defined in a TABLE statement and named A is used. The row index is V1 and the column index is V7. Values of V1 or V7 not included in the table will result in the TABLE function returning a value of -1.

```
R5=TABLE(3,V8,TAB=7,ELSE=TABLE(TAB=1,V1,V6))
```

This uses the table named "7" with 3 as the row index and V8 as the column index. Any value of V8 not in table 7 causes the result from table 1 being used, with row variable V1 and column variable V6 to compute the value of R5.

TABLE Statement

TABLE statements define tables for use by the [TABLE function](#).

Prototype: TABLE name,PRINT,PAD=n, COLS col-1, col-2, ..., col-n, ROWS row-r1(vals-1), row-2(vals-2), ... , row-n(vals-n) ENDTAB

name 1-4 characters used to label the table created.

PRINT (optional)

Prints up to 50 rows of the table for reference.

PAD=n (optional)

A value to insert into any cell defined by the COLS specification, but not by the ROWS specification.

col-1,col-2,...,col-n

The columns of the table to create by this statement. Ranges may be used in the column definitions.

row-1,row-2,...,row-n

The rows of the table to create by this statement. The total size of the table is m by n, where m is the number of columns and n is the number of rows.

(vals-1),(vals-2),...(vals-n)

The values to place in the table (row-r1 values are inserted into r1, etc.). The values are given in the same order as the column specifications; the first value is placed in col-1, the second in col-2, etc. Ranges may be used in the row value definitions. If a PAD value is not supplied, then the number of row values must equal the number of columns.

ENDTAB is optional and indicates the end of the table statements.

A TABLE statement cannot have a label.

Example:

```
TABLE A,PAD=9,COLS 1-3,7,ROWS 3(3,5,1,0), 4(6,8),1(1-4),8(3) ENDTAB
```

The table produced by this definition is:

	1	2	3	7
3	3	5	1	0
4	6	8	9	9
1	1	2	3	4
8	3	9	9	9

TRUNC Function

The TRUNC function returns the integer value of the argument passed to the function.

Prototype: TRUNC(arg)

Arg is any arithmetic expression for which the integer value is desired.

Examples:

```
R5=TRUNC(V5)
```

R5 is the integer value of V5. Thus, if V5 is 5.9, then R5 will be 5.

```
V15=TRUNC(V24/100)
```

V15 is assigned the integer value of the original value of V24 divided by 100. Thus, if V24 originally had a value of 3455 it would be reassigned a value of 34.

```
R100=V24-TRUNC(V24/100)*100
```

This shows how to get the last two digits of a number. If V24 has the value 1234, R100 is assigned the value 34.

EXAMPLE

The following brief example uses RECODE in a Runfile to:

Compute the average speed at which a respondent travels to the nearest city (V25 is the distance; V26 is the time of travel).

Collapse V8 (education) into three categories: no high school (V8=1-6), high school (V8=7-9), and college (V8>9). V8=0 is missing data.

Collapse V9 (occupation) into three categories: high status, medium status, and low status.

Create an inconsistency index from the collapsed education and occupation codes where code 1 is low on both, medium on both, or high on both, code 2 is medium on one only, and code 3 is low on one, high on the other. Code 0 is missing data on either or both.

Compute the mean of four variables V31-V34, allowing up to 1 missing-data value. If more than one missing-data value, then set the result to missing-data.

Each data record is recoded as it is passed from the input data file to the command using the recode.

```
RECODE
  RECODE 1

& COMPUTE AVERAGE SPEED

  IF MDATA (V25,V26) OR V26 EQ 0 THEN R1=MD1(R1) ELSE R1=V25/V26

& COLLAPSE EDUCATION

  R2=BRAC(V8,0=0,1-6=1,7-9=2,>9=3)

& COLLAPSE OCCUPATION

  R3=BRAC(V9,1-2=3,10=3,3-5=2,9=2,6-8=1,ELSE=0)

& FORM INCONSISTENCY INDEX

  TABLE 1, COLS=(1-3), ROWS 1(1-3), 2(2,1,2), 3(3,2,1)
  R4=TABLE(R2,R3,TAB=1,ELSE=0)

& COMPUTE MEAN OF V31-V34

  R100=0
  R101=0
  IF NOT MDATA(V31) THEN R100=R100+1 AND R101=R101+V31
  IF NOT MDATA(V32) THEN R100=R100+1 AND R101=R101+V32
  IF NOT MDATA(V33) THEN R100=R100+1 AND R101=R101+V33
  IF NOT MDATA(V34) THEN R100=R100+1 AND R101=R101+V34
  IF R100 LE 2 THEN R101=MD1(R101) ELSE R101=R101/R100

END
```


REGRESSION

Multiple Regression Analysis

File Assignments:	DATASET	Input data
	MATIN	Input correlation matrix (conditional)
	DATAOUT	Residuals output dataset(optional)

GENERAL DESCRIPTION

Computes standard or step-wise multiple regressions. It accepts categorical (dummy) predictors. With the step-wise option, predictors may be forced into the regression before the step process begins.

See [LOGIT LINEAR](#) for linear and logistic alternatives and IVEware [REGRESS](#) for polytomous, Poisson, Tobit and proportional hazard regression models and for data resulting from a complex sample design.

COMMAND FEATURES

Categorical predictor variables.

There are times you may want to include a categorical variable in the model such as gender or education level. You cannot enter them directly because they are not continuously measured variables, but they can be represented by dummy variables. For example, if variable V32 is "Education of Head" with categories 1=0-11th Grade, 2=Completed HS, 3=Some College, 4=College Degree, 5=Graduate Degree you can create four dummy variables with the dummy statement

V32(2-5)

0-11th Grade will not have a dummy variable but instead is represented by the other four dummy variables all being equal to zero. For each of these, the regression compares the category in question to the base case 0-11th Grade. See Draper and Smith (1981, p. 134) for a discussion of dummy predictors.

PRINTED OUTPUT

Means and Standard Deviations: (Optional: see option PRINT=USTATS.)

Correlation Matrix: (Optional: see option CORR.) The correlation matrix with variable names and numbers labeling the rows and columns.

Marginal R-squares for Potential Predictors: (Step-wise regression only.) The marginal R-squares for all potential predictors are printed for each step. For categorical (dummy) predictors a marginal r-square for all the categories of the predictor taken together is also printed, since

this will be used to determine whether to enter or remove all codes of the categorical (dummy) predictor simultaneously. T-ratios and probabilities also are printed for the categorical (dummy) predictors.

Multiple Regression Statistics: For each step of a step-wise regression and once for a standard regression, the following are printed:

- Standard error F-ratio and probability
- Multiple correlation coefficient (unadjusted and adjusted)
- R-squared, unadjusted and adjusted
- Determinant of the correlation matrix
- Residual degrees of freedom (n-k-1)
- Constant term
- Durbin-Watson statistic (option DW)

Predictor Statistics: For each step of a step-wise regression and once for a standard regression, the following statistics are printed for each predictor:

B-coefficient	Standard error of B
Beta-coefficient	Standard error of beta
Partial r	Part correlation (root of Marginal RSQD)
Marginal r-squared	T-ratio and probability
Covariance ratio	

B-coefficient Variance-covariance Matrix: (Optional: see the option BCOVAR.) The lower left triangle of the B-coefficient variance-covariance matrix, including diagonal elements.

Plot(optional): A plot of the actual data and predicted values using Excel. Because Excel uses up to 255 points for a chart, a random selection of approximately 255 data points is used.

INPUT DATA

Raw Data Input: Raw data input (MATRIX=n option not used) consists of a MicroSiris dataset. A case is deleted for all analyses if it contains missing-data codes for any variable used in any analysis. Categorical variables may be treated as a set of dummy variables through the CATL option. Use **IVEWARE** to impute missing-data first if there is a lot of missing data.

Matrix Input: Matrix input (MATRIX=n option used) consists of a symmetric MicroSiris correlation matrix with means and standard deviations included as produced by CORRELATIONS.

OPTIONS

Choose REGRESSION from the command screen and make selections.

For a Runfile use:	REGRESSION
	Filter statement (optional)
	Job Title
	Keyword choices from below

MATRIX=n n is the number of the correlation matrix to use.

PRINT=(BCOVAR,CORR, USTATS)

BCOVAR Print the B-coefficient variance-covariance matrix.

CORREL Print the correlations.

USTATS Print the means and standard deviations.

RECODE=n Use RECODE n, previously entered via the RECODE command.

WT=n Use variable n as a weight variable.

CAT='stmt' Stmt is a list of categorical variables and valid codes to transform into dummy predictors. For each variable every code specified is transformed into a dummy predictor, e.g., V100(5,6,1,2),V101(1-6,7)

VARS=variable numbers Use the variables specified in the list.

DEPV=variable number Identifies the dependent variable.

ZERO Force a zero intercept (i.e., no constant term). ZERO may not be used with matrix input.

STEP=PROB|FLEVEL Indicates a step-wise regression is desired. *Default:* none.

FORCE=(variable list) Predictor variables to force into a step-wise regression first.

FIN=n The F-ratio below which a variable will not be entered in a step-wise regression when STEP=FLEVEL. This is the F-level to enter. Default: FIN=.001.

FOUT=n The F-ratio above which a variable will remain in a step-wise regression when STEP=FLEVEL. This is the F-level to remove. Default: FOUT=0.0.

AIN=n The significance level below which a variable will be entered in the model when STEP=PROB. Default: ALPHA=.025.

AOUT=n The significance level below which a variable will remain in the model when STEP=PROB. This is the alpha level to remove. Default: ALPHA=.025.

DW Print Durbin-Watson statistic (requires extra pass of the data). Ignored if matrix input (MATRIX=n keyword not used) or if categorical variable statement used.

PLOT Plot regression line with CHART.

RESIDUALS=DATASET|RECODE

Create recode to compute residuals with predicted value variable number 10000 and residual variable number 10001.

DATASET Write a dataset using the recode.

RECODE Create recode only for use in subsequent commands.

EXAMPLE

Regression using matrix input from a previous command (CORRELATIONS).

*** REGRESSION ANALYSIS ***

PREDICTING SIZE OF CAR

Using matrix file 1

Matrix 1 read in successfully.

Matrix N: 327

STANDARD REGRESSION

THE DEPENDENT VARIABLE IS V193: SIZE OF CAR

STANDARD ERROR OF ESTIMATE 1.76

F-RATIO FOR THE REGRESSION 22.364 PROBABILITY 0.00

MULTIPLE CORRELATION COEFFICIENT 0.4663 ADJUSTED 0.4557

FRACTION OF EXPLAINED VARIANCE 0.2174 ADJUSTED 0.2077

DETERMINANT OF THE CORRELATION MATRIX 0.40792

RESIDUAL DEGREES OF FREEDOM (N-K-1) 322

CONSTANT TERM 2.2746 STD. ERROR 0.278041

VARIABLE	NAME	B	SIGMA(B)	BETA	SIGMA(BETA)	PARTIAL R	PART R	MARGINAL RSQD	T-RATIO(PROB)	COVARIANCE RATIO
V24	NO. OF FU ADULTS	-9.49622E-02	0.16343	-3.62700E-02	6.24194E-02	-0.032	0.029	0.0008	0.5811(0.569)	0.3762
V26	NO. OF CHILDREN	5.71664E-02	6.14398E-02	4.67478E-02	5.02423E-02	0.052	0.046	0.0021	0.9304(0.356)	0.0372
V189	NO OF CARS OWNED	1.1728	0.17005	0.46381	6.72497E-02	0.359	0.340	0.1156	6.8968(0.000)	0.4626
V268	TOTAL FAMILY INC	7.29110E-06	1.62451E-05	2.76663E-02	6.16427E-02	0.025	0.022	0.0005	0.4488(0.659)	0.3604

REGRESS

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

REGRESS fits linear, logistic, polytomous, Poisson, Tobit and proportional hazard regression models for data resulting from a complex sample design. Jackknife repeated replication is used to estimate the sampling variances (Kish and Frankel 1974).

REGRESS invokes the SrcWare version of IVEware (installed with MicrOsiris) to perform the analysis.

COMMAND FEATURES

A simple random sample analysis is performed if STRATUM, CLUSTER and WT variables are not specified. Consider using [REGRESSION](#) and [LOGIT-LINEAR](#) for simple random samples. Note: LOGIT LINEAR models probability ($y=1$); The REGRESS logistic option models probability ($y=1$) by default, but optionally models prob ($y=0$), which is mathematically equivalent but leads to a sign difference in the coefficients (PZERO option.)

If a design based analysis involves only a WT variable and no STRATUM or CLUSTER variable, then a pseudo stratification variable and a pseudo cluster variable should be used. When using pseudo variables, all observations in the data set should have the same value for the pseudo STRATUM variable (e.g., 1), while each observation should have a unique value on the pseudo-CLUSTER variable (e.g., observation ID. The pseudo variables can be created with RECODE prior to performing the analysis.

IVEWARE creates name.set and name.data, where name is the name of the input dataset, which are submitted to SrcWare. You can save these (SAVE option) for later modification and refinement and use them directly with Srcware. See [Srcware User Guide](#) for details.

PRINTED OUTPUT

Sum of squares:

Model
Error
Total
R-square
F-value
P-value

For each Variable:

Estimate

Std Error
T Test
Prob > |T|
95% Confidence Interval

OUTPUT DATA

If the WRITE=(PREOUT,ESTIMATES,REPOUT,ID=vn) options are specified, MicrOsiris DATASETS are created for each one specified. The names of the files are in the form name_regress.xxx.DIC where name is the input dataset name and xxx is PRE, REP, EST according to options used.

RESTRICTIONS

Alphabetic variables may not be used.

REFERENCES

[**IVEware**](#) was developed by the Survey Methodology Program at The University of Michigan's Survey Research Center, Institute for Social Research.

OPTIONS

Choose IVEWARE from the command screen and make selections.

(The descriptions for most of these keywords were adapted from the [**IVEware User Guide.**](#))

SAVE Save the IVEWARE setup and data file for later modification and use with the SRCLIB version of IVEWARE.

MAXPRED=n Maximum number of predictors (stepwise regression performed).

MINRSQD=decimal number

Minimum marginal r-squared for a stepwise regression. (Minimum initial marginal r-squared for a logistic regression, and minimum initial r-squares for any code being predicted for a polytomous regression.) A small decimal number like 0.005 builds very large, time-consuming, regression models while 0.25 will include a smaller number of predictors in the regression models. If neither MAXPRED nor MINRSQD is set then no stepwise regression is performed.

MAXLOGI=n Maximum number of iterative algorithms to be performed in a logistic or multilogit regression model. Default: 50.

MINCODI=n Minimum proportional change in any regression coefficient to continue the logistic regression iteration process. This applies to the convergence criterion for the logistic, polytomous and Poisson regression models.

PZERO For logistic regression, model prob (y=0) instead of prob (y=1). The models are mathematically equivalent but produce coefficients with opposite signs.
Default: model prob (y=1).

STRATUM=Vn Use variable n as the stratum variable.

CLUSTER=Vn Use variable n as the cluster variable.

WT=Vn Use variable n as a weight variable.

DEPV=n Use variable n as the dependent variable.

PREDICTORS= variable numbers
 Predictor variables are assumed continuous unless defined as CATEGORICAL (see below). Interaction terms are specified using the "*" notation, e.g.,
 PREDICTORS=(V1,V2,V1*V2).

CATP=variable numbers
 List of categorical predictors.

BYVARS=variable numbers
 The regression analysis is performed for each level of the variable(s) specified in the BYVARS statement.

NOINTER Fit the regression models without the intercept term. (This is like the ZERO (intercept) option in MicrOsiris REGRESSION.)

LINK=(LINEAR,TOBIT,LOG,PHREG,LOGISTIC)
 LINEAR: Fit a multiple linear regression model.
 LOG: Fit a Poisson regression model for a count variable.
 TOBIT: Fit a Tobit model.
 PHREG: Fit a Proportional Hazards model (Cox model).
 LOGISTIC: Fit a logistic (binary) or generalized logistic (polytomous) regression model.

OFFSETS=(count variable, code variable)
 Use to specify an offsets variable when fitting a Poisson regression model (LOG). For example, if V1 is Injuries and V2 is years, OFFSETS=(V1,V2) will fit a model predicting the number for injuries occurring per year.

MAXPRED=n Maximum number of predictors (stepwise regression performed).

MINCODI=n Minimum proportional change in any regression coefficient to continue the logistic regression iteration process. This applies to the convergence criterion for the logistic, polytomous and Poisson regression models.

CENSOR=(Vn, number)
 Vn is a censoring variable, and number is the code indicating censoring. Censor is required if LINK=PHREG. For example, if V1 is survival time and v2=died, then LINK=PHREG DEPV=V1 CENSOR=(V2,0) predicts survival time censoring on whether or not the respondent died.

ESTIMATES= label: specification

Use for estimating values of the dependent variable for a specific set of covariates or testing hypotheses involving the estimated regression coefficients. For example, given the regression model:

$$Y = b + b_1 x_1 + b_2 x_2 + b_3 x_3$$

If you are interested in predicting Y when $x_1=1$, $x_2=2$ and $x_3=0$, then you can obtain the predicted value and the 95% confidence interval by using:

ESTIMATES=Mylabel: Intercept (1) X1(1) X2(2)

Several estimates can be requested by separating them with '/':

ESTIMATES=Mylabel1: Intercept (1) X1(1) X2(2) /Mylabel2: Intercept (1) X1(1) X3(1) /Mylabel3: Intercept (1) X2(1) X3(1)

IVEWARE does not check the specification for syntax within the parentheses; this is done by the IVEware software.

WRITE=(PREDOU,REPOU,ESTIMATES,ID=Vn)

PREDOU: Write a dataset of predicted values, residuals and leverage information.

REPOU: Write a dataset of estimates and their variances-covariances.

ESTIMATES: Write a file of estimated regression coefficients for each combination of STRATUM, CLUSTER and BY.

ID=n: ID variable to include in the PREDOU file.

EXAMPLE

Logistic regression.

REGRESS PROCEDURE

Setup listing:

```
DATAIN DREG3_data;  
DEPENDENT V1;  
PREDICTOR V2 V3;  
LINK LOGISTIC;
```

OUTPUT FOR REGRESS

Regression type:	Logistic
Dependent variable:	V1 Better or worse
Predictors:	V2 Income (000) V3 Children
Cat. var. ref. codes:	V1: Better or worse 1
Valid cases	22
Degr freedom	19
-2 LogLike	29.5166129

Variable	Estimate	Std Error	Wald test	Prob > Chi
Intercept	-0.6616183	0.8699280	0.57843	0.44693
V2: Income (000)	0.0143210	0.0335839	0.18184	0.66980
V3: Children	-0.0080501	0.3012823	0.00071	0.97868

Variable	Odds Ratio	95% Confidence Interval	
		Lower	Upper
Intercept			
V2: Income (000)	1.0144241	0.9455664	1.0882961
V3: Children	0.9919822	0.5280085	1.8636606

SEARCH - SEARCHING FOR STRUCTURE

File Assignments:	DATASET	Input data
	DATAOUT	Residual dataset (optional)

GENERAL DESCRIPTION

The purpose of SEARCH is to allow an evaluation of many competing and probably mis-specified models. It uses a binary segmentation procedure to develop a predictive model for a dependent variable. It relies on the fact that the explanatory power of any one predictor is rapidly exhausted by a few binary splits using it, so that a sequence of binary splits allowing competing predictors at each split, can search data for structure without restrictive assumptions of linearity or additivity of effects. The approach is closer to analysis of variance components than to sequential regression.

SEARCH examines set of predictor variables for those predictors which most increase the researcher's ability to account for the variance or distribution of a dependent variable. The question, "what dichotomous split on which single predictor variable will give us a maximum improvement in our ability to predict values of the dependent variable?" embedded in an iterative scheme, is the basis for the algorithm used in this command.

SEARCH divides the sample, through a series of binary splits, into a mutually exclusive series of subgroups. They are chosen so that, at each step in the procedure, the split into the two new subgroups accounts for more of the variance or distribution (reduces the predictive error more) than a split into any other pair of subgroups. The predictor may be ordinal or nominal-scaled variables. The dependent variable may be continuous or categorical.

SEARCH is an elaboration of the Osiris III AID and THAID programs.

COMMAND FEATURES

Research questions are often of the type "What is the effect of X on Y?" But the answer requires answering a larger question "What set of variables and their combinations seems to affect Y?" With SEARCH a variable X that seems to have an overall effect may have its apparent influence disappear after a few splits, with the final groups, while varying greatly as to their levels of Y, showing no effect of X. The implication is that, given other things, X does not really affect Y.

Conversely, while X may seem to have no overall effect on Y, after splitting the sample into groups that take account of other powerful factors, there may be some groups in which X has a substantial effect. Think of economists' notion of the actor at the margin. A motivating factor might affect those not constrained or compelled by other forces. Those who, other things considered, have a 40-60 percent probability of acting, might show substantial response to some motivator. Or a group with very high or very low likelihood of acting might be discouraged or encouraged by some motivator. But if X has no effect on any of the subgroups generated by Search, one has pretty good evidence that it does not matter, even in an interactive way.

SEARCH makes a sequence of binary divisions of a dataset in such a way that each split maximally reduces the error variance or increases the information (chi-square or rank correlation). It finds the best split on each predictor and takes the best of the best.

The process stops when additional splits are not likely to improve predictions to a fresh sample or to the population, i.e., when the null probability from that split rises above some selected level (e.g., .05, .025, .01 or .005). Of course, having tried several possibilities for each of several predictors, the null probability is clearly understated. Alternative stopping rules can be used in any combination: minimum group size, maximum number of splits, minimum reduction in explained variance relative to the original total, or maximum null probability.

Splitting criteria

There can be four splitting criteria, based on the dependent variable type:

- Means
- Regressions
- Classifications
- Ranks

The splitting criterion in each case is the reduction in ignorance (error variance, etc.) or increase in information. Terms like classification and regression trees should be replaced by binary segmentation or unrestricted analysis of variance components, or searching for structure. With rich bodies of data, many non-linearities and non-additivity possible, and many competing theories, the usual restrictions and assumptions that one is testing a single model are not appropriate. What does remain, however, is a systematic, pre-stated searching strategy that is reproducible, not a free ransacking.

Means: For means the splitting criterion is the reduction in error variance, that is, the sum of squares around the mean, using two subgroup means instead of one parent group mean.

Regressions: For regressions ($y=a+bx$) the splitting criterion is the reduction in error variance from using two regressions rather than one.

Classifications (Chi option): For classifications (categorical dependent variable), the splitting criterion is the likelihood-ratio chi-square for dividing the parent group into two subgroups.

Ranks (Tau option): For rankings (ordered dependent variable), the splitting criterion is Kendall's tau-b, a rank correlation measure.

Stopping Rules

There are four stopping rules, each with a default option:

1. Maximum number of splits. Default: 25.
2. Minimum number in any final group. Default: 25.
3. Minimum reduction in error, relative to the original total. Default: 0.8 percent.
4. Maximum null probability. Default: none, no significance test.

A combination of the minimum number in any final group and the minimum reduction in error is a primitive significance test, but a more formal test is possible. Assuming that the minimum in any final group is 15 or more, the degrees of freedom for any test will be over 30, large enough

to assure reasonable normality, and a Z-ratio (ratio of the gain from a split relative to its standard error) would be 2.33 for a maximum probability that there is nothing there (null hypothesis) of .01. The loss from trying several splits is small if predictor order is maintained, or each class is only tried against all the others (k-1 or k).

For the tau-b option, we cannot define a minimum reduction in error, relative to the original total, so we use a minimum tau-b value for each split. Even with means a minimum error reduction can cause difficulty if the first few splits account for a large fraction of the variance, and the "significance level" however fraudulent, is perhaps a better stopping rule.

For the means and ranks criteria, the maximum null probability stopping rule is based on Z, the ratio of the gain from a split to its standard error, using the normal distribution for the null probabilities. For the regression option, we use an f-test to get the null probabilities, and for the chi option, we use the chi-squared distribution.

The null probabilities are not multiplied by the number of alternatives tried (the Bonferroni correction), since for monotonic predictors or select predictors with fewer than 10 categories, the alternatives are few enough and not really independent. We suggest not using the "free" option with more than three or four categories.

Predictor Order

For each predictor one can maintain its **monotonic** order ("monotonic"), try each class against all the others ("**select**"), or reorder each time according to the criterion variable ("**free**"). The last should be used rarely, and only with predictors with few classes, for it involves implicitly trying many things, resulting in a bias in favor of that predictor. In addition, the combinations split off are difficult to interpret and probably idiosyncratic, as the parent groups become smaller.

With monotonic predictors one tries the first class against the rest, then the first two classes against the rest, etc., making k-1 tries. With select predictors one tries each class against all the others, making k tries, but since the splitting criterion combines difference between the two new groups with both their sizes, there is an offsetting bias against the select option. The alternatives are not really independent, so the bias in favor of predictors with more classes should be small. And with at least 50 cases, adjusting the degrees of freedom would make little difference.

Ranking

Predictors can also be hierarchically ranked as to when they are used. Rank 0 means compute the potential gain but do not split on that predictor. Ranks 1, 2, 3, etc., mean exhaust the rank 1 variables first, then try the rank 2 ones, then the rank 3 ones, etc. Since the program will produce recode statements to generate expected values or residuals, one can also hold aside some later-stage predictors for an analysis of the residuals.

Weighting

The use of weights to adjust for different sampling or response rates affects variances and tests, so the program calculates an estimate of that effect for the whole sample, based on the variance of the weights, and issues a warning. Weights should be used, because if they do not make a difference, nothing is lost, but if they do, the unweighted data are biased.

Missing Data.

Cases with missing-data in a continuous dependent variable or a covariate are deleted automatically. Cases with missing-data in a categorical dependent variable can be excluded by using a Filter or by specifying valid codes with the DEPV selection. Cases with missing-data in the predictor variables are not automatically excluded. However, the Filter and the CODES list may be used to exclude missing-data on predictor variables. Consider using [IVEWARE](#) or [USTATS](#) to impute for missing data before running SEARCH.

One might want to reassign missing information to some large class, or, better, use a multivariate assignment procedure, for example, SEARCH itself with the chi-square option.

PRINTED OUTPUT

The major components of the printed output are specified below. For details see *Searching for Structure*.

Trace Printout: (Optional: See options PRINT=TRACE and PRINT=FULLTRACE). Can be voluminous.

- The candidate groups for splitting
- The group selected for splitting
- All eligible splits for each predictor (optional)
- The best split for each predictor
- The split selected

Final Tables:

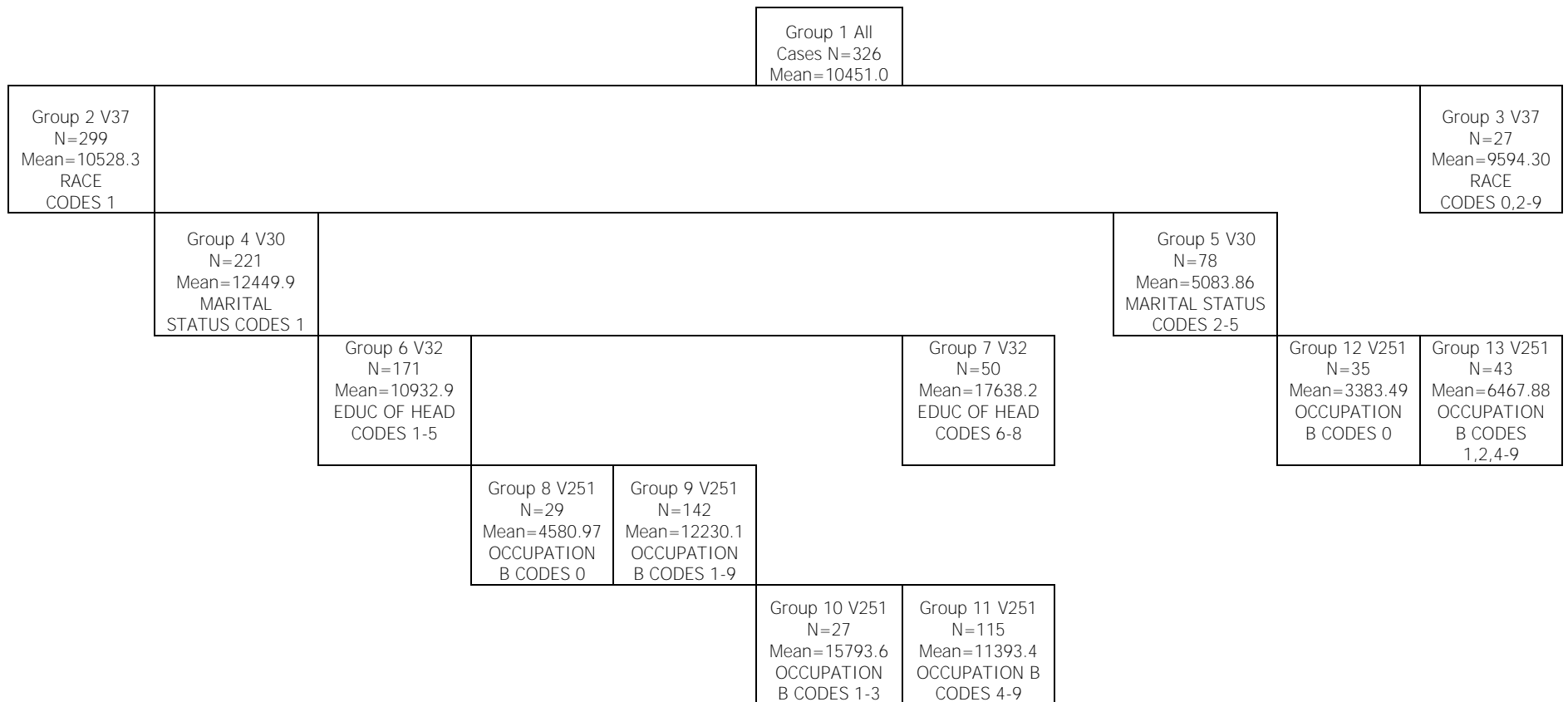
- The analysis of variance or distribution on final groups (except for "analysis=tau")
- The split summary
- The final group summary
- Summary table of best splits for each predictor for each group (except for "analysis= tau")
- The predictor summary table. You may request the first group (PRINT=FIRST), the final groups (PRINT=FINAL), or all groups (PRINT=TABLE). The tables are printed in reverse group order, i.e., last group first and first group last.

Group Tree Structure

A structure table with entries for each group, numbered in order and indented, so that one can easily see the pedigree of each final group and its detail. With relatively little word-processing one has a publishable table.

Tree Diagram

To get a clear tree diagram, save the output to a CSV file when execution finishes. Then, using Excel and following simple instructions you get a nice graphical tree:



INPUT DATA

The dependent variable may be continuous or categorical. Predictor variables may be ordinal or nominal scales.

RESTRICTIONS

Maximum number of predictors: 200.

Maximum number of dependent variables: 33 (includes variable number and codes for categorical dependent variable).

Maximum number of predictor codes: 100.

Maximum number of predefined splits: 49.

OPTIONS

Choose SEARCH from the command screen and make selections.

For a Runfile use:	SEARCH Filter statement (optional) Job Title Keyword choices from below
--------------------	----------------------------------------------------------------------------------

ANALYSIS=MEAN|REGRESSION|CHI|TAU Analysis type.
 MEAN Means analysis.
 REGR Regression analysis
 CHI Likelihood-ratio chi-square (categorical dependent variable)
 TAU Kendall's tau b (ranks)
 Default: ANALYSIS=MEAN.

COV=variable number
 A covariate variable number for REGRESSION analyses.

DEPV=variable number|Vn/list of codes|variable list
 The dependent variable or variables.

For means or regression analyses, a single dependent variable is specified.

For Chi and Tau analyses, a single variable, a variable number and list of codes or a variable list can be specified:

If a variable number and a list of codes is supplied, e.g., DEPV= V7/1,2,4-7, no missing data tests are made for the dependent variable and only the codes listed are used in analysis. Specifying a single dependent variable implies the default list of codes 0-9 and missing-data tests are made.

If a list of variables is given the analysis is done on the distribution of the variables--SEARCH does not perform multiple analyses on each dependent

variable in the list. Note that all cases where the sum of the dependent variables is 0 are deleted. *Default:* none, DEPV must be specified (see note under ANALYSIS option).

EXPL=x Minimum percentage increase in explanatory power required for a split.
Default: EXPL=0.8

ID=variable number
Identification variable to print with each case classified as an outlier.
Default: dependent variable.

MAX=n Maximum number of partitions.
Default: MAX=25.

MIN=n Minimum number of cases in one group.
Default: MIN=25.

NULL=n Maximum probability that there is really no gain from the split.
Default: No significance test.

OUTDISTANCE=n Number of standard deviations from the parent group mean defining an outlier. Outliers are reported if TRACE is specified but not excluded from the analysis. Outliers can be excluded in subsequent runs by filtering
Default: OUTD=5.0

PRINT=(TRACE|FULL, TABLE, FIRST, FINAL)

TRACE: Print trace of splits for each predictor for each split.
FULL: Print trace of splits for each predictor including suboptimal splits.
TABLE: Print all the predictor summary tables.
FIRST: Print the predictor summary tables for the first group.
FINAL: Print the predictor summary tables for the final groups.

RECODE=n Use RECODE n, previously entered via the RECODE command.

RESIDUALS=DATASET|RECODE

Create recode to compute residuals with group variable number 9999 and residual variable the residual variable number is the first of a set of consecutive variables representing the deviation of the case from the expected pattern. A two-stage analysis can be performed by using the residuals from one analysis as the dependent variable(s) for a subsequent analysis.

DATASET Write a dataset using the recode.
RECODE Create recode only for use in subsequent commands.

SYMMETRY=n
The amount of explanatory power one is willing to lose in order to have symmetry, expressed as a percentage.
Default: SYMMETRY=0.

WT=n Use variable n as a weight variable.

Predictor Statements

Supply one predictor statement for group of predictors which can be described with the same parameter values.

VARs=(variable numbers)

Use the variables specified in the list. If you want RECODE R-type variables you must list them explicitly.

Default: none, VARs must be supplied.

M|F|S

The predictor constraint.

M: Predictors are considered "monotonic," i.e., the codes of the predictors are kept adjacent during the partition scan.

F: Predictor codes are considered "free."

S: Predictor codes are "selected" and separated from the remaining codes in forming trial partitions.

Default: M.

CODES=list of codes

List of 2 to 100 acceptable codes. Specifying an exact list improves efficiency.

Default: CODES(0-9). Any case with any predictor value outside its list of acceptable codes is skipped.

RANK=n

Assigned rank. Rank 1 predictors are used before rank 2, rank 2 before rank 3, etc. A zero rank indicates that statistics are to be computed for the predictors, but they are not to be used in the partitioning.

Default: RANK=1.

Predefined Split Statements

If predefined splits are desired, supply one set of parameters for each predefined split.

GNUM=n

Number of the group split. Groups are specified in ascending order, where the entire original sample is group 1. Each set of parameters forms two new groups.

Default: none, GNUM must be supplied.

VAR=variable number

Predictor variable used to make the split.

Default: none, VAR must be supplied.

CODES=(list)

List of the predictor codes defining the first subgroup. All other codes will belong to the second subgroup.

Default: none, CODES must be specified.

REFERENCES

Agresti, Alan (1996), *Introduction to Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Dunn, Olive Jean, and Virginia A. Clark (1974), *Applied Statistics: Analysis of Variance and Regression*, New York: Holt, Rinehart and Winston.

Chow, G. (1960), "Test of Equality between Sets of Coefficients in Two Linear Regressions," *Econometrica*, 29:591-605.

Gibbons, Jean Dickinson (1997), *Nonparametric Methods for Quantitative Analysis*, 3rd edition, Syracuse: American Sciences Press.

Hays, William (1988), *Statistics*, 4th edition, New York: Holt, Rinehart, & Winston.

Klem, Laura (1974), "Formulas and Statistical References," in *Osiris III*, Volume 5, Ann Arbor: Institute for Social Research.

Sonquist, J. A., E. L. Baker and J. N. Morgan (1974), *Searching for Structure*, revised edition, Ann Arbor: Institute for Social Research, The University of Michigan.

EXAMPLES

Example 1: Investigates income (V268) using ANALYSIS=MEANS

predictor statements: v=v32 codes=(0-8)
v=v37,v251,v30
predefined split: gnum=1 var=v37 codes=1

```
*** SEARCH -- SEARCHING FOR STRUCTURE ***

SEARCH SAMPLE SETUP for Means Analysis, Predefined split

Dataset scf

ANALYSIS TYPE: MEANS

Using dictionary SCF.DIC
Using data file SCF.DAT

Dependent variables: V268

Predictor variables: V3 V32 V37 V251 V30

1 case rejected:

    1 for invalid code, predictor 1

326 cases accepted

The partitioning ends with 7 final groups

The variation explained is 37.6 percent.

One-way Analysis of Final Groups
```

Source	Variation	DF
Explained	6.905096E+09	6
Error	1.145044E+10	319
Total	1.835553E+10	325

Split Summary Table

Group 1, N=326

Mean(Y)=10451.0 Var(Y)=5.647856E+07 Variation=1.835553E+10
Split on37: Race Variance explained=2.160400E+07 Significance=.00000
Into Group 2, Codes 1
And Group 3, Codes 0,2-9

Group 2, N=299

Mean(Y)=10528.3 Var(Y)=5.705396E+07 Variation=1.700208E+10
Split on30: Marital status Variance explained=3.128121E+09 Significance=.00010
Into Group 4, Codes 1
And Group 5, Codes 2-5

Group 4, N=221

Mean(Y)=12449.9 Var(Y)=5.719987E+07 Variation=1.258397E+10
Split on32: Educ of head Variance explained=1.739443E+09 Significance=.00010
Into Group 6, Codes 1-5
And Group 7, Codes 6-8

Group 6, N=171

Mean(Y)=10932.9 Var(Y)=4.301275E+07 Variation=7.312168E+09
Split on251: Occupation b Variance explained=1.409004E+09 Significance=.00010
Into Group 8, Codes 0
And Group 9, Codes 1-9

Group 9, N=142

Mean(Y)=12230.1 Var(Y)=4.023026E+07 Variation=5.672467E+09
Split on251: Occupation b Variance explained=4.233616E+08 Significance=.00138
Into Group 10, Codes 1-3
And Group 11, Codes 4-9

Group 5, N=78

Mean(Y)=5083.86 Var(Y)=1.675309E+07 Variation=1.289988E+09
Split on251: Occupation b Variance explained=1.835620E+08 Significance=.00099
Into Group 12, Codes 0
And Group 13, Codes 1,2,4-9

Final Group Summary Table

Group 3, N=27

Mean(Y)=9594.30 Var(Y)=5.122491E+07 Variation=1.331848E+09

Group 7, N=50

Mean(Y)=17638.2 Var(Y)=7.208898E+07 Variation=3.532360E+09

Group 8, N=29

Mean(Y)=4580.97 Var(Y)=8.239151E+06 Variation=2.306962E+08

Group 10, N=27

Mean(Y)=15793.6 Var(Y)=9.242610E+07 Variation=2.403079E+09

Group 11, N=115

Mean(Y)=11393.4 Var(Y)=2.496515E+07 Variation=2.846027E+09

Group 12, N=35

Mean(Y)=3383.49 Var(Y)=5.159423E+06 Variation=1.754204E+08

Group 13, N=43

Mean(Y)=6467.88 Var(Y)=2.216680E+07 Variation=9.310056E+08

Percent Total Variation Explained by Best Split for Each Group (*=Final Groups)

	1	2	3*	4	5	6	7*	8*	9	10*	11*	12*	13*
V32	12.00	11.90	0.00	9.48	0.86	3.62	0.00	0.00	0.68	0.00	0.16	0.00	0.00
V37	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
V251	18.12	16.90	0.00	9.14	1.00	7.68	0.00	0.00	2.31	0.00	0.27	0.00	0.00
V30	17.92	17.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Group Tree Structure

Group 1 All Cases N=326 Mean=10451.0

Group 2 V37 N=299 (91.7%) Mean=10528.3 Race CODES 1

Group 4 V30 N=221 (67.8%) Mean=12449.9 Marital status CODES 1

Group 6 V32 N=171 (52.5%) Mean=10932.9 Educ of head CODES 1-5

Group 8 V251 N=29 (8.9%) Mean=4580.97 Occupation b CODES 0

Group 9 V251 N=142 (43.6%) Mean=12230.1 Occupation b CODES 1-9

Group 10 V251 N=27 (8.3%) Mean=15793.6 Occupation b CODES 1-3

Group 11 V251 N=115 (35.3%) Mean=11393.4 Occupation b CODES 4-9

Group 7 V32 N=50 (15.3%) Mean=17638.2 Educ of head CODES 6-8

Group 5 V30 N=78 (23.9%) Mean=5083.86 Marital status CODES 2-5

Group 12 V251 N=35 (10.7%) Mean=3383.49 Occupation b CODES 0

Group 13 V251 N=43 (13.2%) Mean=6467.88 Occupation b CODES 1,2,4-9

Group 3 V37 N=27 (8.3%) Mean=9594.30 Race CODES 0,2-9

Example 2: CHI analysis on variable V46.

Predictor statements: v=v32 codes=(0-8)

v=v37,v251,v30 f

*** SEARCH -- SEARCHING FOR STRUCTURE ***

Search sample setup, No predefined split

Dataset scf

CHI Analysis

ANALYSIS TYPE: CHI

Using dataset SCF

Dependent variables: V46

Predictor variables: V3 V32 V37 V251 V30

1 case rejected:

1 for invalid code, predictor 1

326 cases accepted

Split 1 Candidate Groups

Group	N	Variation
1	323	0.794292E+03

Attempt to split Group 1 Variation 794.292

Predictor V32: Educ of head, Rank 1, Type M, Codes 1-8

Best split after Code 3, Variance explained=19.1247, Significance=.00051

Predictor V37: Race, Rank 1, Type F, Codes 1-3,9

NO ELIGIBLE SPLIT

Predictor V251: Occupation b, Rank 1, Type F, Codes 0-9

Best split after Code 1, Variance explained=12.1421, Significance=.00738

Predictor V30: Marital status, Rank 1, Type F, Codes 1-5

NO ELIGIBLE SPLIT

Best split for Group 1 on Predictor V32: Educ of head

Variance explained=19.1247, Significance=.00051

Split Group 1 on V32: Educ of head Variance explained=19.1247 Significance=.00051

Into Group 2, Codes 1-3

And Group 3, Codes 4-8

Split 2 Candidate Groups

Group	N	Variation
2	143	0.362606E+03
3	183	0.412561E+03

Attempt to split Group 3 Variation 412.561

Predictor V32: Educ of head, Rank 1, Type M, Codes 4-8

NO ELIGIBLE SPLIT

Predictor V37: Race, Rank 1, Type F, Codes 1-3

NO ELIGIBLE SPLIT

Predictor V251: Occupation b, Rank 1, Type F, Codes 0-9

NO ELIGIBLE SPLIT

Predictor V30: Marital status, Rank 1, Type F, Codes 1-5

NO ELIGIBLE SPLIT

NO ELIGIBLE SPLIT FOR GROUP 3

Split 2 Candidate Groups

Group	N	Variation
2	143	0.362606E+03

Attempt to split Group 2 Variation 362.606

Predictor V32: Educ of head, Rank 1, Type M, Codes 1-3

NO ELIGIBLE SPLIT

Predictor V37: Race, Rank 1, Type F, Codes 1-3,9

NO ELIGIBLE SPLIT

Predictor V251: Occupation b, Rank 1, Type F, Codes 0-9

NO ELIGIBLE SPLIT

Predictor V30: Marital status, Rank 1, Type F, Codes 1-5

NO ELIGIBLE SPLIT

NO ELIGIBLE SPLIT FOR GROUP 2

No splits possible

The partitioning ends with 2 final groups

The variation explained is 2.4 percent.

One-way Analysis of Final Groups

Source	Variation	DF
Explained	1.912473E+01	3
Error	7.751668E+02	320
Total	7.942915E+02	323

Split Summary Table

Group 1, N=326, Variation=794.292

Split on32: Educ of head Variance explained=19.1247 Significance=.00051

Into Group 2, Codes 1-3

And Group 3, Codes 4-8

Final Group Summary Table

Group 2, N=146, Variation=362.606

Group 3, N=180, Variation=412.561

Percent Total Variation Explained by Best Split for Each Group (*=Final Groups)

	1	2*	3*
V32	2.41	0.18	0.56
V37	0.66	0.00	0.00
V251	1.53	0.45	0.52
V30	0.56	0.61	0.33

DEPENDENT VARIABLE PERCENT DISTRIBUTION FOR EACH GROUP (*=FINAL GROUPS)

	1	2*	3*
1	25.46	15.75	33.33
3	49.39	50.00	48.89
5	12.58	16.44	9.44
8	12.58	17.81	8.33

Group Tree Structure

Group 1 All Cases N=326 Dependent codes: 1=25.5%,3=49.4%,5=12.6%,8=12.6%

Group 2 V32 N=146 (44.8%) Dependent codes: 1=15.8%,3=50.0%,5=16.4%,8=17.8% Educ of head CODES 1-3

Group 3 V32 N=180 (55.2%) Dependent codes: 1=33.3%,3=48.9%,5=9.4%,8=8.3% Educ of head CODES 4-8

TABLES -- FREQUENCIES AND NON-PARAMETRIC STATISTICS

File Assignments:	DATASET	Input data
	KAPPA	Kappa weight matrix (optional)
	MPUTn	Output table n matrix (see MPUT option)

GENERAL DESCRIPTION

Produces univariate, bivariate, or three-way frequency tabulations and percentages. Each table can be written out as a [CSV](#) file for subsequent processing by spreadsheet and graphics programs such as Microsoft EXCEL (See example 2).

TABLES can provide quantiles and numerous nonparametric measures of association and significance for ordinal or nominal data and produce matrices of non-parametric correlations.

COMMAND FEATURES

Variables and tables may be combined by using the COMBINE option. Missing-data values are optionally deleted on an individual analysis basis. When the STRATA option is not specified, or is set to the value 0, univariate frequencies are displayed with optional percentages. When the STRATA option is specified, bivariate frequencies are produced, with the strata variable as the row variable. Additional control is provided by the REPETITION option, used to define multiple analyses for up to 25 categories, which does not require sorted data. Use RECODE and SORT DATASET to create appropriate repetition variables and code categories.

When the CONCAT option is used, tables are concatenated into a compressed form. For each variable in the VARS list the code values for each STRATA variable appear as successive rows in a single table. A totals line marks the end of the values for that variable. If a strata variable has no code value labels, the first 16 characters of the variable name are used for each row label instead.

Tables can be saved as matrix files (MPUT) suitable for input to [CHANGE RESPONSE](#).

SPECIAL TERMINOLOGY

Strata: Strata are disjoint sub-collections of data cases.

Strata (row) Variable: A variable used to partition the data set into disjoint sub-collections of data cases. Each code (except for missing-data codes) of the strata variable defines a single stratum. You may create strata variables with RECODE if they are not already in the data set. For frequency tables, the strata variable corresponds to the row variable.

PRINTED OUTPUT

Frequency Tables (optional)

Ranks (PRINT=RANKS)

- Sum of ranks/stratum
- Average rank/stratum
- N for each stratum

Chi-square Statistics (STATS=CHI)

- Chi-square
- Cramer's V
- G-square (likelihood ratio chi-square (Hays, p. 737)
- Yates correction (when table is 2 X 2)
- Contingency coefficient

Chi-square and G goodness-of-fit tests (STATS=GCHI)

- Chi-square
- G

Cochran's Q (STATS=CQ)

- Requires RECODE (see example 4)
- Q
- Probability chance occurrence (if $N_k > 24$ and $N > 3$)

Fisher exact test (STATS=FISHER)

- Probability of chance occurrence for 2 x 2 tables
- Probability of equal or worse distribution (one-tailed)
- Probability of equal or worse distribution in either direction (two-tailed)

Mann-Whitney (STATS=MW)

- U statistic
- Z approximation (if possible; see Siegel)
- Probability of chance occurrence (if Z approximation given)

Spearman Rho(STATS=SPRM)

Statistics for Nominally Scaled Data (STATS=NOMINAL)

- Lambda (symmetric), lambda a, lambda b
- Standard scores of the lambdas
- Approximate standard errors (ASE) of lambdas
- Leik-Gove D (corrected)
- Goodman and Kruskal's tau a and tau b
- Agreement coefficients for 2x2 tables: Eta, Phi, Phi/phimax, Yule's Q, Yule's Y, Tetrachoric r, Koppa

Agreement coefficients for 3x3 tables: Approval coefficient, Lijphart index

Gini coefficient and Lorenz Plot (STATS=GINI)

Lorenz distribution deciles and plot
Gini coefficient based on Lorenz distribution deciles

Kappa Statistic (STATS=KAPPA)

Data entry table
Table of weights
Kappa statistic and variance of Kappa

Kendall's coefficient of concordance W (STATS=CONCORDANCE)

W
Probability of chance occurrence

Kolmogorov-Smirnoff two-sample test (STATS=KS)

D
Probability chance occurrence for one- and two-tailed tests
For n, m 25 or less: nmD, m, and n

Kruskal-Wallis H (STATS=KWH)

Degrees of freedom
H, unadjusted, and probability of chance occurrence
H, adjusted, and probability of chance occurrence

Quantiles (NTILE=n)

Quantiles, e.g., deciles

Sign test (STATS=SIGN)

Number of differences with positive sign
Z approximation (if $N > 35$; see Siegel)
Probability of chance occurrence

Statistics for Ordinally-scaled Data (STATS=ORDINAL)

Kendall's tau a, tau b, tau c
Gamma statistic and variance of gamma
Somers' d
Probability of chance occurrence of the taus, gamma, and Somers' d
Agreement coefficients for 2x2 tables: Eta, Phi, Phi/phimax, Yule's Q, Yule's Y, Tetrachoric r,
Kappa
Agreement coefficients for 3x3 tables: Approval coefficient, Lijphart index

Wilcoxon (STATS=WILC)

Sum of positive ranks
Z approximation (if $N > 15$; see Siegel)
Probability of chance occurrence (if Z approximation given)

INPUT DATA

Statistics produced by TABLES normally require data measured at the ordinal or nominal level. Consequently, decimal value category codes are always truncated to the nearest integer. When STATS=KAPPA, a MicrOsiris matrix may be provided for the weights table (see the option KMATRIX).

If the data contains blanks or invalid codes and there are no missing-data codes defined in the dictionary, "MISSING DATA" is displayed instead of a code value.

RESTRICTIONS

All codes accessed analysis must be within the range -2,147,483,647 to 2,147,483,647. Larger codes are dropped from the calculations.

For nonparametric statistics and tests, e.g., Kendall's W, the data must be ranks with the rankers as the stratum (row) variable.

OPTIONS

Choose TABLES from the command screen and make selections.

For a Runfile use:	TABLES Filter statement (optional) Job Title Keyword choices from below
--------------------	----------------------------------------------------------------------------------

PRINT=DICT Print the input dictionary.

RECODE=n Use RECODE n, previously entered via the RECODE command.

WT=n Use variable n as a weight variable

CORR=(SPRM|GAMMA|LAMBDA|TAUC|CRAMER
Produce correlation matrix instead of TABLES and writes it to MATOUT. Matrix number will be 1. No tables or statistics are displayed--just the matrix of correlations and only one analysis statement is allowed. *Default:* No correlation matrix is produced.

Analysis Statements

The user may supply any number of analysis statements unless CORR was specified. All options desired must be specified on each analysis statement. If CORR was specified, only the VARS and DELETE keywords are used and no tables are displayed.

VAR=variable list

List of one or more variables; for bivariate frequencies, each variable in the list constitutes a column variable. Each variable in the list constitutes one analysis.

Default: None; VARS is required.

STRATA=variable list

Stratum variable numbers. Each variable in the list will be used with each variable specified with the VARS list to form a separate analysis, unless the COMBINE option is also specified. There are as many strata in a given analysis as there are codes for the stratum variable. For Mann-Whitney statistics, there must be two and only two strata per strata variable. For Kruskal-Wallis statistics, there must be at least two strata per strata variable.

EP=(list of numbers)

Expected percentages for STATS=GCHI. Must be integers and sum to 100. There must be exactly the same number of values as there are categories in the variable(s) being tested.

REPETITION=(Vn=list1[\$name1]/list2[\$name2]...) **Can't be used with CONCAT or COMBINE)**

Defines up to 25 subsets of cases where each subset is defined by Vn=list/n. One analysis is done for each subset. Example: REPE= (V1=1\$Primary/2-5\$Other) defines and labels two repetitions, one for V1=1 and the other for V1 in the range 2-5. See [REPEAT and REPETITIONS](#) for more information.

STATS=(CHI,CQ,FISHER,FRIED,GCHI,GINI,KAPPA,KS,KW,KWH,MW,NOMINAL,ORDINAL, SIGN,SPRM,WILCOXON)

CHI	Chi-square, Cramer's V, and G-square (likelihood ratio chi-square).
CQ	Cochran's Q (needs RECODE; see Example 4)
FISHER	Fisher exact test. Sample size must be 100 observations or less.
FRIED:	Friedman test
GCHI:	Chi-square and G goodness-of-fit tests
GINI	Gini coefficient and Lorenz plot (can't use strata variable).
KAPPA	Cohen's Kappa.
KS	Kolmogorov-Smirnov two-sample test
KW	Kendall's coefficient of concordance W.
KWH	Kruskal-Wallis H.
MW	Mann-Whitney U.
NOMINAL	Lambda, Lambda a, Lambda b, Leik-Gove D, Goodman and Kruskal's tau a, tau b.
ORDINAL	Kendall's tau a, tau b, tau c, gamma, Somers' d.
SIGN	Sign test.
SPRM	Spearman rank order correlation (Rho).
WILCOX	Wilcoxon signed ranks test.

DELETE=(MD1,MD2)

MD1	Delete missing-data code 1 from all percentage and statistical calculations for all variables and strata variables.
MD2	Delete missing-data code 2 from all percentage and statistical calculations for all variables and strata variables.

The table still shows the missing-data codes, but is labeled "DELETED".

If the data contains blanks or invalid codes and no missing-data codes are defined in the dictionary, "MISSING DATA" is displayed instead of a code value.

PRINT=(ROW%,COL%,TOT%,RANKS,SINGLE,UNWT)

ROW%	Print percentages based on row (stratum) totals. Use for univariate percentages also.
COL%	Print percentages based on column totals.
TOT%	Print percentages based on total N.
RANKS	Print sum of ranks, average rank, and N for each stratum.
SINGLE	Single-space rows of the tables (horizontal grid lines will be eliminated except at the top and bottom).
UNWT	For weighted data, print the unweighted cell N's.

TITLE='string' A 1- to 80-character title for the table(s).

SUPPRESS=(TABLE,GRID,ROWC,COLC)

TABLE	Suppress printing the frequency table.
GRID	Suppress the "grid" (i.e., lines around the cells).
ROWC	Suppress printing the row codes.
COLC	Suppress printing the column codes.

COMBINE Combine the variables specified by VARS and STRATA by treating each variable as a separate case. Thus two or more bivariate frequency tables may be combined (merged) into a single table. If COMBINE is used, the REPETITION option is ignored if specified.

CONCAT Concatenate the tables specified by VARS and STRATA into a compressed form. For each variable in the VARS list, the code values for each STRATA variable appear as successive rows in a single table. A totals line marks the end of the values for that variable. If a strata variable has no code value labels, the first 16 characters of the variable name are used for each row label instead. If CONCAT is used, the REPETITION option is ignored if specified, no statistics are calculated except percentages (if requested), and COMBINE cannot be used.

MPUT Write table to MPUTn as a matrix n, where n is the ordinal table number (bivariate only).

NTILES=n Used to request n-tiles (e.g., deciles) and to indicate the number of intervals desired. For example, to get deciles specify NTILES=10; to get the median, specify NTILES=2. NTILES can't be specified with stratification.

KMATRIX=n The number of the weight matrix for the KAPPA statistic.
Default: A weight matrix with 1's on the diagonal is used.

REFERENCES

Hays, W. L. *Statistics for the Social Sciences*. Second edition; New York: Holt, 1973.

Siegel, S. *Nonparametric Methods for the Behavioral Sciences*. Second edition. New York: McGraw-Hill, 1988.

Agreement: coefficient: Weisberg (1968). Page 106. See also Anderson (1966), pages 61-64.

ETA: Weisberg (1968), page 125. Weisberg note the measure is due to Donald Stokes, University of Michigan.

Koppa: MacRae (1970), page 43. Koppa is an adjusted agreement measure (-1 to 1) showing the proportion of cases with identical scores on both variables.

Lijphart index: MacRae (1970), page 213. Anderson (1966), page 53.

Phi: MacRae (1970), page 46, formula 3.1. This is the Pearson product moment correlation between two variables scored in a point fashion.

Phi/phimax: Weisberg (1968), pages 119, 128. This version is similar to the usual phi/phimax (see MacRae, 1970, page 47.)

Tetrachoric r: MacRae (1970), page 48, formula 3.3. It is an estimate of what the correlation would be if the underlying traits were continuous and normally distributed.

Yule's Q: MacRae (1968). This is a special case of the Goodman-Kruskal gamma coefficient.

Yule's Y: MacRae (1970), page 50, formula 3.5.

EXAMPLES

Example 1: Marital Status by Education of Head of Household.

*** CROSS TABULATIONS AND NONPARAMETRIC TESTS ***

MARITAL STATUS BY EDUCATION

Dataset scf

327 cases accepted.

Variable V30 MARITAL STATUS
Strata(row) V32 EDUC OF HEAD

Marital Status by Education of Head of Household

	Married	Single	Divorced	Widowed	Separated	Totals
0-8th Grade	8	3	6	2	0	19
9th Grade	52	2	17	2	0	73
10th Grade	35	1	9	8	1	54
11th Grade	56	5	7	4	2	74
Completed HS	32	3	2	1	0	38
Some College	25	2	3	2	3	35
College Degree	15	1	1	1	0	18
Graduate Degree	13	1	0	1	0	15
	0	0	1	0	0	1
Totals	236	18	46	21	6	327

Example 2: Univariate frequencies with COMBINE. One table is produced containing all the values for variables V1, V2, and V3.

*** CROSS TABULATIONS AND NONPARAMETRIC TESTS ***										
COMBINED UNIVARIATE FREQUENCIES										
Dataset sample										
5 cases accepted										
Variables										
V1 Better or Worse										
V2 Income (000)										
V3 Children										
Code	-4	0	1	2	3	4	5	37	99	Totals
Frequency	1	2	4	1	2	2	1	1	1	15

Example 3: Chi-square statistics.

```

*** CROSSTAB AND RANK ORDER STATISTICS ***

Chi-Square statistics

Dataset scf
326 cases accepted.
Variable      V26      NO. OF CHILDREN
Strata(row) V37      RACE

      |      0 |      1 |      2 |      3 |      4 |      5 |      6 |      7 | Totals
White  1 |  167 |    23 |    48 |    27 |    19 |    10 |     3 |     2 |   299
Black  2 |   11 |     6 |     3 |     1 |     0 |     3 |     0 |     0 |    24
Other  3 |     1 |     2 |     0 |     0 |     0 |     0 |     0 |     0 |     3
      |-----|-----|-----|-----|-----|-----|-----|-----|-----|
Totals 179    31    51    28    19    13     3     2   326

Degrees of Freedom 14

Chi-Square 26.55  Probability .02

Likelihood Ratio (G-Square) 20.24  Probability .13

Cramer's V .20  Contingency Coefficient .27

```

Example 4:

Cochran's Q with k=3 related samples. For Cochran's Q, recode the k related samples to 1-k indicating the successful responses, leaving the unsuccessful responses as 0s, and create R1 equal to the square of the number of successes for each case times 100. Use COMBINE to make a single table as shown below.

```

RECODE
  R1=0
  IF V1 EQ 1 THEN R1=R1+1
  IF V2 EQ 1 THEN R1=R1+1 AND V2=2
  IF V3 EQ 1 THEN R1=R1+1 AND V3=3
  R1=R1*R1*100
END
TABLES
TITLE
RECODE 1
  V=V1-V3,R1 COMBINE STATS=COCHRAN
END

```


SORT DATASET

File Assignments: DATASET Input data

GENERAL DESCRIPTION

Sorts a MicroSiris dataset into ascending order on up to nine ID variables. Non-MicroSiris files may be sorted by using the FIXED option of [IMPORT](#) to create a dictionary for the data file.

OPTIONS

Choose SORT DATASET from the command screen and specify ID variables.

OUTPUT DATA

The sorted dataset. The original data file is preserved with the original dataset name with the date and time added and the suffix '.BAK' added.

EXAMPLE

Sorting a data file with ID variables 1 and 3.

```
*** SORT DATASET ***

Dataset sample

Sort by variables:

  VNUM      NAME                TYPE  LOC  WID  NDEC      MD1      MD2  REFNO
   V1  Age                  C    1   4    0        0      99    1
   V3  Income              C    9   4    0                2

Record length is    20

Loading ID values...

Sorting      5 records
```

TEXT-TO-CSV FILE CONVERSION

File Assignments:	TEXT_FILE	Input file
	CSV_FILE	Output file

GENERAL DESCRIPTION

TEXT-TO-CSV creates a CSV file of numbers and text suitable for Excel from a text file. Text strings are separated from numbers and put in separate cells. Thus the string "Correlation Coefficient: .37" will appear in two adjacent cells.

When a command finishes, you have the option to save it in text format or CSV format, so you do not need this command for MicroSirius output. It is included as a utility for other text files you may have.

Special Features

Grid lines, i.e. lines that contain a series of 4 or more dashes or underscores like those used in TABLES are skipped.

Vertical bars (|) grave accents (`) are removed and treated as cell breaks.

'?' is used as a place holder for a cell when preceded by a blank. Thus ' ? ? hello' puts 'hello' in the third cell.

A string enclosed in double quotes surrounded by blanks is treated as a single cell except that extra blanks are removed.

[PLOT] at the beginning of a line signals the following lines are to go into the first cell. [END] at the beginning of a line signals stop putting lines into the first cell. To display properly in Excel, these lines should have a monospace font like Consolas, applied.

TRANSFORM -- FILE TRANSFORMATIONS

File Assignments:	DATASET	Input data
	DATAOUT	Output data

GENERAL DESCRIPTION

Creates a new MicrOsiris dataset from an existing one. Both the dictionary and the data file can be altered by TRANSFORM. You can use TRANSFORM to convert the mode (type) of variables for compatibility with other systems and for sub-setting of cases (see also [EXPORT](#)). Use TRANSFORM to insert new variables created or modified by RECODE into the dataset, thereby making permanent copies of them.

COMMAND FEATURES

Changing the Dictionary: For any variable in the variable list you can change:

- Storage mode of a variable (TYPE);
- Width of a variable (WIDTH).

Changing to TYPE=C and an appropriate width can greatly reduce the size of a large dataset if you have many variables with a maximum value of 1, 2, or 3 digits.

Variables in the Output Dictionary and Data File: Variables always appear in ascending order in the output dictionary, regardless of their order in the Options variable list or the dictionary descriptor records (if any). Take care that there will be no duplicates caused by RECODE; e.g., R100 becomes V100 in the output dictionary.

SPECIAL TERMINOLOGY

Character numeric type: Each digit of a number, its sign or exponent indicator (e.g. 3.14, 1.5E9), occupies one location of the total field reserved for it on a storage device. Character numeric can be used to store any number up to 15 characters.

Floating-point type: Data values occupy four or eight contiguous storage. Floating-point mode may be used to represent any number in the range -10^{75} to 10^{75} with accuracy to seven significant digits when four storage locations are used and 16 significant digits when eight locations are used. This is the most efficient mode for interval measured data.

OPTIONS

Choose TRANSFORM from the command screen and make selections.

PRINT=(DICT,OUTD)	
DICT	Print the input dictionary.
OUTD	Print the output dictionary.

RECODE=n Use RECODE n, previously entered via the RECODE command.

VAR=(variable numbers)|ALL

Use the variables specified. If V=ALL is given, all variables in the dictionary are used plus any RECODE variables.

Default: VAR=ALL.

Variable Specification Statements (optional)

For any variable or set of variables in the variable list, the following options may be used to override the values specified under Options above. This statement may be repeated as many times as required.

VAR=variable list|ALL Variables to which the following parameters apply. ALL implies this is the only specification statement and is used to change all variables to type CHARACTER or FLOATING and a constant width, e.g., TYPE=FLOAT, WIDTH=4.

TYPE=CHARACTER|FLOATING|ALPHABETIC
CHARACTER

Variables are to be character numeric. Recode variables will default to WIDTH=10, NDEC=2.

FLOATING Variables are to be in floating-point mode, recommended for decimal or continuous data.

ALPHABETIC

Variables are alphabetic.

Default: FLOATING.

WIDTH=n Output variable width. Always 4 or 8 for FLOATING.

Default: Original width for types CHARACTER and ALPHABETIC and 8 for FLOATING, except when converting from floating-point variables to [character numeric](#) type when the width is set to accommodate largest possible value.

EXAMPLE

Creating a new variable by bracketing on income.

Recode statement: r100=brac(v268,else=9,0-20=1,<80=2,<150=3,<300=4,>299=5)

Options: recode=1 vars=v3,v20,v24, v37,,v189,v193, v251,v268,r100

Dictionary statement: var=r100 width=1 type=integer ndec=0

T-TEST -- COMPARE MEANS

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

Tests the mean of a variable, the difference between means (paired observations), or the difference between two group means. The paired observations test requires the variable selected to be the difference under study; use RECODE to create it if necessary.

PRINTED OUTPUT

Statistics are printed for each analysis as described below.

Mean vs. Hypothesized Value and Paired Observations

- Hypothesized value
- Mean
- Standard deviation
- Standard error
- N

T statistic and its associated probability (two-tailed test; for a one-tailed test, use half the associated probability.)

Degrees of freedom

Difference Between Group Means

- Mean for each group
- Standard deviation for each group
- N for each group
- Difference of means
- Z and its associated probability (separate variances)
- T and its associated probability (pooled variances)

OPTIONS

Choose T-TEST from the command screen and make selections.

For a Runfile use:	T-TEST
	Filter statement (optional)
	Job Title
	Keyword choices from below

RECODE=n Use RECODE n, previously entered via the RECODE command.

WT=n Use variable n as a weight variable

REPETITION=(Vn=list1[\$name1]/list2[\$name2]...)

Defines up to 25 subsets of cases where each subset is defined by Vn=listn. One analysis is done for each subset. Example: REPE= (V1=1\$Primary/2-5\$Other) defines and labels two repetitions, one for V1=1 and the other for V1 in the range 2-5. See [REPEAT and REPETITIONS](#) for more information.

Analysis Statements

More than one statement may be supplied.

VAR=variable numbers

List of variables to test.

MEAN=x

x is the hypothesized mean or difference of means; not used for differences of group means.

Default: MEAN=0.0.

GROUP=variable

The variable defining the groups to test. All cases for which the GROUP variable assumes the value G1 are in GROUP 1; those with the value G2 are in GROUP 2. Cases with other values are skipped.

Default: Group tests are not performed.

G1=n

Group 1 code for differences between group means. Must be in the range 0-99.

Default: 1

G2=n

Group 2 code for differences between group means. Must be in the range 0-99.

Default: 2

REFERENCES

Snedecor, George W. and William G. Cochran. *Statistical Methods*, Chapter 4. Sixth edition. Ames, Iowa: Iowa State University Press, 1976.

EXAMPLE

Analysis1 tests the null hypothesis that the mean SAT score is 500. Analysis 2 compares group means for variable GPA; RECODE was invoked first to create recode number 1 to select the two groups.

Recode statement: IF V1 GT 3.16 THEN R1=2 ELSE R1=1

*** COMPARE MEANS ***

Testing SAT Scores and GPA

Dataset C:\DEVELOPMENT\PROJECTS\TESTDATA\STAT

Transforming the data by RECODE 1 read from C:\DEVELOPMENT\PROJECTS\TESTDATA\T-TEST\HYPOTH.RUN

50 cases accepted

Testing Hypothesized Mean

	Name	Hypothesized Mean	Mean	Standard Deviation	Standard Error	N	t	p(t)	df
V2	Sat-v	500.0	540.6	97.44	13.78	50	8.6898	0.01	49

Difference Between Two Means

Group variable R1 Recode variable R1 G1=1 G2=2

	Mean(1)	Mean(2)	Difference	Standard Deviation(1)	Standard Deviation(2)	N(1)	N(2)	Z	p(Z)	t	p(t)	df	NAME
V1	2.653	3.554	-0.9013	0.23934	0.25813	24	26	-12.81	0.00	-12.77	0.00	48	Gpa

USTATS -- UNIVARIATE STATISTICS

File Assignments:	DATASET	Input data
	ZSCORES	Recode statements for Z-scores(optional)

GENERAL DESCRIPTION

Computes descriptive statistics for a given set of variables. Optionally computes the same statistics for each variable for each specified repetition.

COMMAND FEATURES

Missing-data codes are always deleted from the computations, and the standard deviation is calculated as the square root of the unbiased estimate of the population variance.

Additional control is provided by the REPETITION option, used to define multiple analyses for up to 25 categories.

Use option IMPUTE or ZSCORES to create recode to impute means for missing data or standardized values. Subsequent commands can immediately use the recode to use the imputed values.

PRINTED OUTPUT

For ungrouped data, sums, means, standard deviations, skewness, kurtosis, coefficient of variation, range (minimum and maximum) and the Jacque-Bera test for normality and its probability value may be requested.

For grouped data, the above statistics, except Jacque-Bera may be requested. A table of the requested statistics is displayed with all ordered combinations of the grouping variables as rows and the statistics for each variable as the columns.

RESTRICTIONS

Grouping variable codes must not be negative.

OPTIONS

Choose USTATS from the command screen and make selections.

For a Runfile use:	USTATS Filter statement (optional) Job Title Keyword choices from below
--------------------	----------------------------------------------------------------------------------

NDEC=n The minimum number of decimal places use for all statistics, where $-1 < n < 8$.
Default: NDEC=3. If n is less than the maximum number of decimal places specified in the dictionary for all variables, NDEC is increased to that value (up to 8).

RECODE=n Use RECODE n, previously entered via the RECODE command.

VAR=(variable numbers)
If V=ALL is given, all non-alphabetic variables are used.

STATS=(SUM,MEAN,SDEV,SKEWNESS,KURTOSIS,COFVAR,RANGE,JACQUE-BERA)
The statistics to compute. The Jacque-Bera test is computed for ungrouped data only. *Default:* SUM, MEAN, SDEV, RANGE.

WT=n Use variable n as a weight variable

REPETITION=(Vn=list1[\$name1]/list2[\$name2]...)
Defines up to 25 subsets of cases where each subset is defined by $V_n = \text{list}n$. One analysis is done for each subset. Example: REPE= (V1=1\$Primary/2-5\$Other) defines and labels two repetitions, one for V1=1 and the other for V1 in the range 2-5. See [REPEAT and REPETITIONS](#) for more information.

Cases not belonging to any repetition are collected and displayed in a separate table, and a table of all cases is displayed as well.

IMPUTE|STAND
Create recode for recoding missing-data values to the variable means or to standardize values for use in subsequent commands. For STAND, the variable names are the original names appended with -Z. If REPETITIONS are used, the recode applies only to the last repetition.

OUTPUT Write dataset of imputed or standardized data if IMPUTE or STAND specified.

EXAMPLES

Example 1: Univariate statistics for 5 variables.

*** UNIVARIATE STATISTICS ***

UNIVARIATE STATISTICS FOR 5 VARIABLES

Dataset C:\DEVELOPMENT\PROJECTS\TESTDATA\SAMPLE

		Standard							Jacque	
		N	Mean	Deviation	Skewness	Kurtosis	Minimum	Maximum	Bera	P(JB)
Better or Worse	V1	4	1.000	3.559	-1.331	1.500	-4.000	4.000	1.556	0.378
Income (000)	V2	4	10.725	17.538	1.986	3.952	0.900	37.000	5.232	0.105
Children	V3	5	2.200	2.168	0.559	-2.368	0.000	5.000	1.429	0.398
Weight 1	V4	5	1.000	0.000	0.000	0.000	1.000	1.000	0.000	1.000
Assets	V5	5	2.558	1.266	0.708	0.226	1.110	4.430	0.429	0.643

WILDCODE CHECK

File Assignments:	DATASET	Input data
-------------------	---------	------------

GENERAL DESCRIPTION

Tests whether a set of variables has only legitimate data values and lists all invalid codes by case ID and variable number. Once the bad code values are identified, Correct them with RECODE with TRANSFORM or EDIT DATASET.

COMMAND FEATURES

WILDCODE CHECK can find all non-numeric codes present in numeric data fields as well as illegitimate numeric codes. Only numeric variables are checked, and any non-integral value is rounded before checking.

SPECIAL TERMINOLOGY

Wild Code: A variable value that is legal for the variable type but is outside the range of valid codes is considered a wild code. For example, if V7 is sex of respondent (coded 1 for males and 2 for females), a value of 3 is a wild code; a value of "A" is "bad data."

PRINTED OUTPUT

Wild codes and non-numeric codes are documented, one identification line for each record containing bad values, plus one line per bad value. The total number of cases (entries) processed, the number of entries with errors, and the total number of illegal codes found is printed at the end.

OPTIONS

Choose WILDCODE CHECK from the command screen and make selections.

ID=variable list

Up to 10 variables may be specified for use as identification for each case containing illegal codes. If not specified, sequential case numbers are used. ID variables may be alphabetic.

MAXERR=n Maximum number of illegal codes allowed before WILDCODE will stop.
Default: MAXERR=99

RECODE=n Use RECODE n, previously entered via the RECODE command.

Code Specification Statements

Use the following options to specify valid numeric codes for as many sets and variables as desired. Include one statement for each variable or set of variables with unique codes. If no code specification statements are given, WILDCODE CHECK checks all numeric variables in the dictionary for non-numeric codes. If more than one statement, the values specified on each subsequent statement retain the values of the preceding one except for VARS and CODES. For example, if you specify INVALID on statement one, it is remembered for statement two.

VARS=(variable numbers)|ALL

The list of variables to which the following options apply. If V=ALL is given, no other code specification statements may be given. Alphabetic variables will not be checked.

Default: ALL

VALID|INVALID Indicates whether the CODES option specifies valid or invalid codes.

Default: VALID

CODES=(list) A list of integer codes to compare with the variables specified by the VARS list. Non-integral values are rounded before checking.

Default: All numeric codes are valid.

MIN=n The minimum valid code.

Default: -1000000000

MAX=n The maximum valid code.

Default: +1000000000

The fewer the codes specified in the CODES list, the more efficiently WILDCODE will execute. Judicious use of the VALID|INVALID, MIN, and MAX options may save you considerable execution time. Thus V=1 MIN=0 MAX=9 is much better than V=1 CODES=(0-9). MIN and MAX values are always checked first.

EXAMPLE

The following example specifies:

The only valid codes for variables 3, 4, 5, and 7 are in the integer range 0 to 9;

1. All numeric codes for V8 are valid except 7;
2. The only valid codes for V10 are in the integer ranges 1 to 5, and 8 and 9;
3. Variable 11 is to be checked only for non-numeric codes;
4. Variable 12 must be in the ranges 0 to 6, and 81 to 999.
5. Variables 1 and 2 are used as ID variables.

Options: ID=V1,V2

Code specification statements: V=V3-V5,V7 MIN=0 MAX=9

V=8 INVALID CODE=7

V=10 CODES=(1-5,8,9) MIN=1 MAX=9

V=11

V=12 MIN=0 MAX=999 INVALID CODES=(7-80)

*** WILD CODE CHECKER ***

DOCUMENT WILDCODES

Dataset: indict

Reading data...

ID =1 2

V10 Income bracket Wild code: 0

ID =2 3

V12 Territory Wild code: 23

ID =6 6

V10 Income bracket Wild code: 6

5 records checked for 8 variables.

3 wild codes found in 3 records.

RUNFILES

When you enter control statements interactively you use the Command Screen to select the command, specify the dataset to use and choose options.

Instead, for commands that take multiple analysis statements, you may wish to store them in a file, called a Runfile, for later use or repeated use by MicroSiris. This is especially convenient for long recodes that may need some testing before final use. Not all commands can use Runfiles; if a command can use a Runfile it is noted under the options section of the command write-up.

Suppose you have the following RECODE statements stored in the Runfile file BRACKET.RUN:

```
RECODE  
RECODE 1
```

& Blank lines ok in RECODE and between other commands & but not within other commands

& Recoding four levels -> two levels

```
V4= BRAC( V4, TAB=1, ELSE=9, 0-1=0, 2-3=1)  
V5= BRAC( V5, TAB=1)  
V6= BRAC( V6, TAB=1)
```

```
NAME R13'Major Depression R13'
```

```
LABEL='1=yes, 0=no)
```

```
NDEC V13(0)
```

```
IF (V10=1 OR V11=1) AND ((V4+V5+V6) GE 5) THEN R13=1 ELSE R13=0
```

```
END
```

Then you can specify Runfile name BRACKET in the Runfile box as shown:

The screenshot shows the 'Command Screen' window with a yellow title bar. It contains several buttons at the top: 'Import Data', 'Data Entry with Excel', 'What Statistic to use?', 'About MicroSiris/ Check for Updates', and 'Preferences and Settings'. On the left is a list box with commands: AGGREGATION, ANOVA, ANOVAR, CANCELL, CAP, CHANGE RESPONSE, CHART, CLUSTER, COMPARE, CONSISTENCY CHECK, CONJOINT, CORRELATIONS, and DESCRIBE. In the center, there is a 'Data File Directory:' field with the path 'C:\DEVELOPMENT\PROJECTS\TESTDATA\'. Below this is a label '<--Choose command or use existing Runfile-->' followed by a text box containing 'BRACKET' and a 'Browse' button. To the right of the text box is a 'View/Print Output' button. At the bottom, there are 'HELP', 'OK', and 'QUIT' buttons. The 'Output' section has two radio buttons: 'Screen (save at end)' which is selected, and 'Deferred' with a question mark icon next to it.

You can use this Recode file in the Runfile box or include it in another command Runfile :

```
INCLUDE BRACKET.RUN
TRANSFORM DATASET=HEALTH DATAOUT=HEALTHMD
Create new variable, R13                ! R13 becomes V13 in output dataset
RECODE 1 VARS=V1,V4-V10,R113
VAR=R13 WIDTH=1 TYPE=FLOAT             ! Dictionary statement
END
```

When using Runfiles, MicroSiris assumes they can be found in the default data directory, but you may browse to find your Runfiles by clicking the browse button next to the Runfile box.

You can edit existing Runfiles by using the VIEW button on the command screen.

Constructing a Runfile

Use a text editor like NOTEPAD or WordPad to create a text file containing commands.

Structure of a Runfile

When running the command from a [Runfile](#), the control statements have a standard syntax which usually consists of:

- Command name and file assignments
- Filter statement (optional)
- Job Title
- Options

In the command write-ups, Options are defined as keywords for use in Runfiles.

Runfiles contain the command name and any required file assignments, and must have the suffix .RUN. Lines in a Runfile may be continued on successive lines as needed by placing a dash (-) as the last character in each line being continued.

Command Name

The command name is usually the first word of the title to each command write-up and is noted under the OPTIONS section of each write-up.

Filter and Job Title Statement

Most MicroSiris commands require a Job Title, optionally preceded by a [Filter statement](#) when invoked from a Runfile. If you don't want to supply a Job Title, leave the line blank. The commands use this to title the output along with the execution date and the command name.

Options (Keywords)

When using a Runfile, Options and the values you choose are specified with keywords on the Options line in a Runfile. Each command write-up indicates its options and their keywords.

Keyword Syntax

A keyword is a word or word abbreviation used to identify an option or assign a value to a variable. Examples are:


```
WRITE, WT=V3, PRINT=(DICT,PAIR)
DEPV=V5 STEP RESI
```

All keywords are defined in each command description using a standard notation :

1. A vertical bar (|) indicates you may choose only one of the mutually exclusive options, e.g., FLOAT|ALPHA.
2. A comma indicates you may choose all, some, or none of the items, e.g., STATS=(CHI,NOMINAL,GINI).
3. Words in uppercase are keywords. Words or phrases in lowercase indicate that you should replace the word or phrase with an appropriate value, e.g., MAXC=n and WT=Vn.

A blank line or a line with a single asterisk (*) indicates all defaults are taken.

Keywords are separated either by a comma or one or more blanks.

Rules for Specifying Associated Values

Many keywords are of the form "X=Y" where X is the keyword and Y is a value or name assigned to X. The right-hand side of such keywords, e.g., Y, is called the associated value.

If the associated value is a list of items and two or more items are selected, these must be separated by a comma or one or more blanks and be enclosed in parentheses, unless the associated values are variable numbers.

Example: PRINT=(XPRO,USTATS)

If the associated value is a list of numbers, a range of values is indicated by a dash (-).

Example: CODES=(1,5-15)

If the associated value is a character string, it must be enclosed in primes if it contains blanks or commas. A prime may be used in the string if the string itself is not enclosed in primes. Two consecutive primes are required to represent a single prime if the string itself is enclosed in primes.

Examples: NAME='EDUCATION: WAVE 1'
NAME=KEVIN'S
NAME='KEVIN"S'

Rules for Specifying Variables

Variables from the input file are represented either by a "V" followed by the variable number or by the variable number only--either "V2" or "2" is acceptable. Recode variable numbers must always be preceded by an "R", e.g., "R2". Ranges of variables are indicated by a dash, e.g., R1-R5.

Examples: VARS=V1-V5,V8,V11,R14
WT=V16

MicrOsiris Runfile Commands

MicrOsiris Runfile commands provide comment and control functions.

Commands must be entered without leading blanks.

&[comment]

&[comment] can be used at the end of a command or options statement to document that command as well as on a separate line. ! is equivalent to &.

Examples: & CREATING INDEX OF JOB SATISFACTION

&INCLUDE filename

Include statements from another file. When the last statement in the included file is read input resumes from the Runfile. Example:

 &INCLUDE filename

You may use an &INCLUDE statement in an "INCLUDEd" file.

&INCLUDE is particularly useful for including recode statements.

Example: Your Runfile is named RECODE.RUN and consists of the following:

```
RECODE
  RECODE 2
  &
  & Collapse V4: Age
  &
  V4=BRAC(V4,0-15=1,16-30=2,31-45=3,46-60=4,61-75=5,ELSE=6)
```

You could use this as a Runfile by itself, but you can include several commands in a Runfile. For example, using the above recode file combined with a TABLES command in one file, you could put the following commands in a file:

```
&INCLUDE RECODE.RUN
TABLES DATASET=sample
Bivariate and Univariate Frequencies
WT=V9 RECODE 2
vars=v1-v3 strata=v5 suppress=(rowcodes, colcodes)
vars=v1-v4 suppress=(rowcodes, colcodes)
end
```

!Job Title
!Options
!Analysis statements

END

This command signals the end of control statements and data from the input stream for the previous command. It is also used as the last item in a Runfile. It is required only for procedures that do not have a fixed number of control statements, such as ANOVA. If you type END while a MicrOsiris command is awaiting label or Filters, etc., that command terminates.

Example:

```
ANOVA dataset=SCF
INCLUDE V37=1 AND V26=0 AND V193=<9
Survey of Consumer Finances
print=dict
DEPV=V268 VARS=V193
DEPV=V189, VARS=193,37
END
```

Note: If the preceding command is not expecting any more input, an END statement will terminate the Runfile--all succeeding statements are ignored. For example, LIST DATASET only requires a label statement and an options statement. Therefore, an END statement after the option statement is treated as the end of the Runfile.

Procedure Commands for Runfiles

Procedure commands--equivalent to those commands you can pick off the Command Prompt Screen--perform specific analysis and data management tasks. They nearly always require input/output data file assignments. Procedure commands take the form:

Procname [file assignments]

This command initializes execution of the command specified by "procname".

Specific command descriptions appear with each command write-up preceding this section. All required setup statements must follow immediately, and any necessary recode instructions must already have been defined with RECODE prior to the command.

The file assignments define all input data files required by the procedure and all output data files. When you select a command from the command prompt screen, a list of required file assignments appears. For use within Runfiles, the exact language for defining these files is described in the next section.

File Assignments

MicroSiris datasets and matrix files are made available to commands in Runfiles with file assignments, telling the commands which disk files are being read from or written to. The syntax is of the form:

logicalname=filename

For each assignment, supply the name of the file or device. For many commands, the only logical name assignment required is DATASET. Commands creating an output dataset also require DATAOUT. For example:

```
CORRELATIONS DATASET=ABC
TRANSFORM DATASET=OLD DATAOUT=NEW
```

Rules for Assigning Files

1. Use blanks to separate file assignments.

3. Filename can be any legal file/path name, but must be enclosed with single quotes (') if they contain blanks.
5. The logical file name assignments required for each command are given at the beginning of each command write-up.

Any DATASET assignment remains assigned until overridden by a new file assignment.

In general, file assignments other than matrix file assignments remain until explicitly overridden. Matrix file assignments are erased once the matrices are read in because subsequent commands may generate new matrices with the same numbers.

Printed Output (SPRINT)

The SPRINT assignment for printed output defaults to the screen when you start MicroSiris, but you may assign SPRINT to a separate file for each command. The syntax for the SPRINT assignment takes the form:

SPRINT=filename
or SPRINT=filename(APPEND)

where filename is any legal file name. If (APPEND) is not used and filename already exists, it is replaced. Examples follow:

```
LIST DATASET DATASET=MYDATA SPRINT=LIST DATASET.PRT  
CORRELATIONS SPRINT=P(APPEND).
```

Matrix File Formats

If the matrix was not created by MicroSiris, you can use Excel to create a CSV file with all matrix parameters and data values and refer to it for a given command via the file assignment MATIN to enter it into MicroSiris.

The first three lines in a matrix file contain the following:

MATRIX, n *Matrix number for reference by a command (optional, default 1)*

TITLE, 'name' A 1- to 100-character title for the matrix (optional, blanks)

CASES, n Number of cases from which the matrix was derived (optional, default 0).

Column Names

NAMES, followed by all column variable numbers and names separated by commas.

Row Number, Name and Matrix Rows

Each rectangular matrix row begins with the row variable number and name and a comma and the row values separated by commas. Begin each row on a new line, e.g.,

1 name1, 1.2, 1.3.....

For symmetric matrices, enter only the lower triangle, including diagonal elements (See example below) and the row variable and names are optional.

Means and Standard Deviations (optional, symmetric matrices only)

MEANS followed by the means for variables in column order, separated by commas.

STDDEV followed by the standard deviations in the same order, separated by commas.

For example, a symmetric matrix with means and standard deviations looks like this:

```
MATRIX,1
TITLE,CORRELATIONS
CASES,5
NAMES,V1 Better or Worse,V2 Income (000),V3 Children
1
0.70295, 1
0.27541, 0.723788, 1
MEANS,1, 10.725, 2.2
STDDEV,3.559026, 17.53765, 2.167948
```

The easiest way to enter this or other matrices is to use Excel to create the CSV file. The above matrix in Excel looks like this:

	A	B	C	D
1	MATRIX	1		
2	TITLE	CORRELATIONS		
3	CASES	500		
4	NAMES	V1 Better or Worse	V2 Income (000)	V3 Children
5	1			
6	0.70295002	1		
7	-0.27541132	0.72378791	1	
8	MEANS	0.8	28.38	2.2
9	STD	3.1144823	42.298605	2.1679483

A rectangular matrix looks like this:

	A	B	C
1	MATRIX	1	
2	TITLE		
3	CASES	0	
4	NAMES	V101 DIMENSION X	V102 DIMENSION Y
5	V1 POINT 1	-4	0
6	V2 POINT 2	-3	-4.414
7	V3 POINT 3	0	-1.414
8	V4 POINT 4	0	0

Save as a CSV file with the suffix MTX and assign via a MATIN file assignment.

To save with the suffix .MTX, you must put the name you choose in quotes, e.g., "MYMATRIX.MTX" Otherwise Excel will append .CSV to the name, e.g. MYMATRIX.MTX.CSV.

Appendix -- Internal File Formats

A MicroSiris dictionary or data file usually has a 10-byte HDR record at the beginning indicating the record length and file structure. The older type 1 (original OSIRIS format) and type 3 (OSIRIS III and UNESCO IDAMS) dictionaries are still accepted and may be converted to the current MicroSiris dictionary format. The format of this record is:

HDR record

Positions 1-3 HDR (character)
Positions 4-5 Record format (binary) 0 for undefined length, 1 for fixed length
Positions 6-7 Record length (binary) Fixed record length or the maximum record length
Positions 8-10 Reserved

Subsequent records are the same exact record length if the record format is fixed, with no end of record character or if undefined each record is terminated with a carriage return (binary 13) and a line-feed character (binary 10).

Dictionary File

A MicroSiris dictionary is comprised of a dictionary descriptor record and one or more variable descriptor records and code category labels following each variable descriptor record. Each record is 80 characters long.

Dictionary descriptor record.

This record indicates the type of dictionary, which allows MicroSiris to interpret OSIRIS IV and IDAMS dictionaries as well as MicroSiris dictionaries.

Columns	Content
---------	---------

1-3	blank
4	'5' for MicroSiris, '3' for OSIRIS IV and IDAMS
5-20	Blank for MicroSiris, obsolete information for OSIRIS IV and IDAMS.
21-25	If non-blank, the data record length to use for data files with no HDR record.
26-80	blank

Variable descriptor records(slightly different for IDAMS 3 and OSIRIS IV)

Columns	Content
---------	---------

'T'	
2-6	variable number
7-14	blank
15-38	variable name (can't contain \)
39	variable type code (A=alphabetic, C=character numeric, F=floating-point real)
40-41	blank

42-46 variable location
 47-49 variable width (for type F must be 4 or 8, for type C must be <17)
 50-51 number of decimal places
 52-53 blank
 54-62 MD1 (blank for default 15000000000)
 63-71 MD2 (blank for default 16000000000)
 72-80 blank

Code category labels

Columns	Content
---------	---------

1	'L'
2-6	variable number
7-13	blank
14	'0'
15	'#' Indicates labels strings of variable length follow and are of the form:

N1[\$name1]\N2[\$name2]\...

'*' indicates code labels are 16 characters long and the code value width is 9. Otherwise code labels are 8 characters long and the code value width is 5 (OSIRIS IV and IDAMS format).

16-80 code values and labels.

Labels may not contain the double quote character (").

Appendix -- SrcWare User Guide¹

SrcWare invocation

SrcWare may be invoked by double-clicking on the Srcware.exe file (in C:\Program Files\MicrOsiris\srclib) or an icon pointing to it. This brings up the SrcWare editor, from which you can open, edit and save, and run the SrcWare setup and data files created by DISCRIM, IMPUTE, and REGRESS. See [SrcWare User Guide](#) and [IVEware User Guide](#) for more information.

Example, SCF.SET created by REGRESS:

```
%GETDATA(NAME=SCF_DATA, SETUP=NEW);
datain scf_data;

METADATA;
VARIABLES
  NAME=V20    LABEL="Age of head" TYPE FLOATING WIDTH 8 MISSING 1500000000;
  NAME=V24    LABEL="No. of fu adults" TYPE FLOATING WIDTH 8 MISSING
1500000000;
  NAME=V268   LABEL="Total family inc" TYPE FLOATING WIDTH 8 MISSING
1500000000;
END;
RUN;
%REGRESS(name=SCF_regress, setup=new);
DATAIN SCF_data;

DEPENDENT V268;
PREDICTOR V20 V24;
LINK LINEAR;
RUN;
```

GLOSSARY

alphabetic variable. A *variable* which is not to be treated numerically; any symbol may occur in an alphabetic variable.

ASCII An acronym for American Standard Code for Information Interchange. An ASCII file contains only text and numbers with no encoding or formatting.

binary. Pertaining to a system which involves the use of two possible symbols; e.g., a "binary code" uses two distinct characters, usually 0 and 1. (See [floating-point binary](#).)

byte. A measure of main storage capable of containing a single character or number stored in *character numeric* mode.

case. In a rectangular file, the information obtained from a single unit (e.g., person, institution, or household) of a sample; each case has one or more unique *records* of data.

categorical. A *variable* is categorical if it can assume only a finite set of discrete values, usually integers. (Contrast [continuous](#), see also [nominal](#), [ordinal](#).)

variable value labels. 1- to 8-character labels used to label the codes of *categorical* variables when they are displayed, such as in frequency distributions.

character. A letter, digit, or other symbol used to represent data, to control operations, or to organize.

character numeric. Data storage mode wherein each position of storage contains a code representing a single digit or arithmetic sign or exponent indicator, e.g., 3.14, 1.5E9. Character numeric corresponds to the numbers that can be entered from a keyboard.

CSV file. File containing data items separated by commas, where each new line represents a new row, and each row has one or more fields separated by a comma. CSV files are commonly used for transferring data between databases in a simple text-based format. The CSV file format is supported by almost all spreadsheet software such as Excel and OpenOffice Calc, although Excel uses the list separator of the current local settings, which is a semi-colon instead of a comma for many locales. Many database management systems support the reading and writing of CSV files.

continuous. A variable is measured on a continuous scale if it can take on any real value in the interval for which it is defined, e.g., number of inches of rain falling in a month. (Contrast [categorical](#).)

dataset. A collection of related *records*. A Microsirius dataset is composed of two actual files: a dictionary file and a data file.

dictionary. A reference guide to a data *file* which describes (or defines) each *variable* by indicating its characteristics and record location.

dummy variable. A *variable* that takes on the value 0 or 1, such as a variable created with the CAT option in REGRESSION. (See Draper and Smith, 1981 for a discussion of dummy predictors.)

field. See [variable](#), [record](#).

file. A collection of related [records](#) concerning people, things, or places.

floating-point binary. The mode of data storage wherein a number may be stored in four or eight bytes. Numbers may be any decimal value in the range -1075 to 1075. When stored in four bytes, the number is single precision with seven significant digits. When stored in eight bytes, the number is double precision with 16 significant digits. Numbers in floating-point binary always retain as many decimal places as possible (up to the precision of the storage mode); a MicrOsiris dictionary denotes how many decimal places are to be displayed in the printout. All non-categorical (i.e., non-integer) data are automatically converted to floating-point binary before calculations are made.

format free. Describes values or items listed or entered without a fixed pattern or format.

nominal. A variable is measured on a nominal scale if it can take on only a finite set of values, arbitrarily chosen with no particular order (usually integers). An example is sex of respondent, which can be male or female and coded with any two integers, e.g., (0,1). (Contrast [ordinal](#), [continuous](#).)

ordinal. A variable is said to be measured on an ordinal scale if it can take on only a finite set of [ordered](#) values. An example is a scale which measures agreement and is coded 1 = strongly disagree, 2 = disagree, 3 = agree, 4 = strongly agree. (Contrast [nominal](#), [continuous](#).)

parameter. A specification in a command *setup* which selects one of several alternatives for a particular option.

R-variable. A recode result variable, i.e., a *variable* created by RECODE; "R" followed by a number, e.g., R10.

record. A set of data items relating to one person, place, or object. In a rectangular *dataset*, the set of data items for one *respondent* is a *case* and may be composed of one or more *records*. In MicrOsiris, each data item corresponds to a single value of a [variable](#).

rectangular file. A file in which all *variables* for one *respondent* are stored in one or more *records* of the same length, called a *case*. Most non-MicrOsiris files are rectangular.

respondent. The source of the information being studied or analyzed.

simple random sample. A sample in which each point or item in the population from which the sample was drawn has an equally likely chance of being chosen.

symmetrical matrix. One in which the elements below the diagonal elements are the same as the ones above, i.e., element $a(i,j) = a(j,i)$. MicrOsiris stores only the upper triangle portion (normally without diagonal elements, which are assumed to be 1's). The output file from an CORRELATIONS command is such a matrix.

V-variable. An input *variable*, i.e., one which already exists and is defined in the *dataset dictionary*; "V" followed by a number, e.g., V32.

variable. A quantity or characteristic of an item being measured that assumes different values; e.g., occupation, education. Descriptions of variables are stored in a MicrOsiris *dictionary*. Descriptions of variables correspond to descriptions of *fields* in some other systems.

weight. To assign a numerical coefficient (or the coefficient itself) to an item so that it will assume a desired significance relative to the total *dataset*.

REFERENCES

Agresti, Alan (1996), *Introduction to Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Altman, Edward II., Avery, Robert B., Eisenbeis, Robert A., Sinkey, Joseph F., Application of Classification Techniques in Business, Banking and Finance. Greenwich, CN, Aijai Press, 1981.

Anderson, L.F., Watts, M.W., Jr. and Wilcox, A.R. *Legislative Roll-Call Analysis*. Evanston: Northwestern University Press, 1966.

Andrews, F. M., L. Klem, T. N. Davidson, P. M. O'Malley, and W. L. Rodgers, *A Guide for Selecting Statistical Techniques for Analyzing Social Science Data*. Second edition. Ann Arbor: Institute for Social Research, The University of Michigan, 1974.

Andrews, F. M., J. N. Morgan, J. A. Sonquist, and L. Klem. *Multiple Classification Analysis*. Second edition. Ann Arbor: Institute for Social Research, The University of Michigan, 1973.

Andrews, F. M. and R. C. Messenger. *Multivariate Nominal Scale Analysis*. Ann Arbor: Institute for Social Research, The University of Michigan, 1973.

Bard, Y. *Nonlinear Parameter Estimation*. New York: Academic Press, 1974.

Bock, R. D. "Programming Univariate and Multivariate Analysis of Variance." *Technometrics*, 1963, pp. 5, 95-117. (The method of analysis used by the MANOVA program is outlined here.)

Bock, R. D. "Contributions of Multivariate Experimental Designs to Educational Research." In *Handbook of Multivariate Experimental Psychology*. Edited by R. B. Cattell. Chicago: Rand McNally, 1966, pp. 820-40. (Worked examples of three applications of multivariate analysis of variance.)

Bock, R. D. and E. A. Haggard. "The Use of Multivariate Analysis of Variance in Behavioral Research." In *Handbook of Measurement and Assessment in the Behavioral Sciences*. Edited by D. K. Whitla. Reading, Mass: Addison-Wesley, 1968, pp. 100-42. (A good introduction to multivariate analysis of variance.)

Chaing, Chin Long. *The Life Table and its Applications*. Florida: Robert E. Krieger Publishing Company, 1984.

Chow, G. (1960), "Test of Equality between Sets of Coefficients in Two Linear Regressions," *Econometrica*, 29:591-605.

Cochran, W. G. and G. Cox. *Experimental Design*. Second edition. New York: Wiley, 1957. Corrected printing, 1968. (Univariate only.)

Coleman, J.S. Models of Change and Response Uncertainty. Englewood Cliffs, New Jersey: Prentice Hall, 1964.

Cooley, W.H. and Paul R. Lohnes. *Multivariate Analysis*. Wiley, New York, 1971.

Cox, D. R. *The Analysis of Binary Data*. London: Methuen, 1970. Reprinted: Chapman and Hall, 1983.

Draper, N. R. and H. Smith. *Applied Regression Analysis*. Second edition. New York: Wiley, 1981.

- DuMouchel, W. H. "The Regression of a Dichotomous Variable." Internal memorandum, Ann Arbor: Institute for Social Research, The University of Michigan, 1974.
- Dunn, Olive Jean, and Virginia A. Clark (1974), *Applied Statistics: Analysis of Variance and Regression*, New York: Holt, Rinehart and Winston.
- Fleiss, J. L., J. Cohen and B. S. Everitt. "Large Sample Standard Errors of Kappa and Weighted Kappa." *Psychological Bulletin*, Vol. 72, No. 5, 1969, pp. 323-327.
- Gibbons, Jean Dickinson (1997), *Nonparametric Methods for Quantitative Analysis*, 3rd edition, Syracuse: American Sciences Press.
- Gordon, I. J. *Classification: Methods for the Exploratory Analysis of Multivariate Data*, Chapman and Hall, London, 1981
- Grizzle, J. E. "Multivariate Logit Analysis." *Biometrics*, 1971, Vol. 27, pp. 1057-1062.
- Guttman, L. "A general nonmetric technique for finding the smallest coordinate space for a configuration of points." *Psychometrika*, Vol. 33, 1968, pp. 469-506.
- Harman, H. H. "Factor Analysis." in *Handbook of Measurement and Assessment in Computers*. Edited by D. K. Whitla. New York: Wiley, 1960. Harman presents here the derivation and method for the principal factor solution.)
- Hartwig, Frederick. "Statistical Significance of the Lambda Coefficients," *Behavioral Science*, Vol. 18, No. 4, 1973, pp. 307-10.
- Hays, W. L. *Statistics for the Social Sciences*. Second edition. New York: Holt, 1973.
- Johnson, S. C. "Hierarchical Clustering Schemes." *Psychometrika*, Vol. 32, No. 3, 1967, pp. 241-254.
- Hartigan, J. A. and Wong, M. A. "A K-Means Clustering Algorithm." *Journal of the Royal Statistical Society. Series C (Applied Statistics)*. Vol. 28, No 1 (1979). Pp 100-108.
- Kaufman, Leonard and Peter J. Rousseeuw. *Finding Groups in Data*. Wiley, New York, 1990.
- Kempthorne, O. *The Design and Analysis of Experiments*. New York: Wiley, 1952. Reprinted: Krieger, 1975. (Univariate. See page 38 for a discussion of the technique of the general linear hypothesis model.)
- Kendall, M. G. *Rank Correlation Methods*. Fourth edition. New York: Oxford University Press, 1975.
- Kish, Leslie. *Survey Sampling*. New York: Wiley, 1967. Reprinted: Krieger, 1983.
- Klem, L. *OSIRIS III, Volume 5: Formulas and Statistical References*. Ann Arbor: Institute for Social Research, The University of Michigan, 1974.
- Knuth, Donald E. *The Art of Computer Programming, Volume 2 / Seminumerical Algorithms*. London: Addison-Wesley, 1969.
- Kuder, G. F. and M. W. Richardson. *Psychometrika*, Vol. 2, No. 3, Sept. 1937.
- Lingoes, J. C. "An IBM-7090 Program for Guttman-Lingoes Smallest Space Analysis - I." *Behavioral Science*, Vol. 10, 1965, pp. 183-184.
- Lingoes, J. C. "New Computer Developments in Pattern Analysis and Nonmetric Techniques." In *Uses of Computers in Psychological Research*. Paris: Gauthier-Villars, 1966, pp. 1-22.

- Lingoes, J. C. "An IBM-7090 Program for Guttman-Lingoes Multidimensional Scalogram Analysis - II. *Behavioral Science*, Vol. 12, 1967, pp. 268-270.
- Lingoes, J. C. "Some Boundary Conditions for a Monotone Analysis of Symmetric Matrices." *Psychometrika*, Vol. 36, 1971, pp. 195-203.
- MacRae, D., Jr. *Issues and Parties in Legislative Voting*. New York: Harper and Row, 1970.
- McNemar, Q. *Psychological Statistics*. Fourth edition. New York: Wiley, 1969.
- Nerlove, Marc, and S. James Press. "Univariate and Log-Linear and Logistic Models." Rand Corporation Publication No. R-1306-EDA/NIH, Santa Monica: 1973.
- Overall, J. E. and C. J Klett. *Applied Multivariate Analysis*. New York: McGraw-Hill, 1972. Reprinted: Krieger, 1983. pp. 321, 430, 441-68.
- Potter, R. G. "Application of Life Table Techniques to Measurement of Contraceptive Effectiveness." *Demography*, Vol. 3, 1966, no. 2.
- Rao, C. R. *Linear Statistical Inference and its Applications*. Second edition. New York: Wiley, 1973.
- Roskam, E. & J.C. Lingoes. "Minissa-I: A Fortran IV(g) Program for the Smallest Space Analysis of Square Symmetric Matrices." *Behavioral Science*, Vol. 15, 1970, pp. 204-205.
- Rummel, R. J. *Applied Factor Analysis*. Evanston, IL: Northwestern University Press, 1970. (The algorithm for pair-wise deletion of missing data is described on pp. 258-59.)
- Rutherford, Andrew. *Introducing ANOVA and ANCOVA, a GLM Approach*. Sage Publications Ltd., 2001. Reprinted 2007.
- Schonemann, Peter H. and Robert M. Carroll. "Fitting One Matrix to Another under Choice of a Central Dilation and a Rigid Motion." *Psychometrika*, Vol. 35, 1973, pp. 245-55.
- Siegel, S. *Nonparametric Methods for the Behavioral Sciences*. Second edition. New York: McGraw-Hill, 1988.
- Snedecor, George W. and William G. Cochran. *Statistical Methods*. Seventh edition. Ames, Iowa: Iowa State University Press, 1980.
- Sonquist, J. A., E. L. Baker, and J. N. Morgan. *Searching for Structure*. Revised edition. Ann Arbor: Institute for Social Research, The University of Michigan, 1974.
- Truett, J., J. Cornfield, and W. Kannel. "A Multivariate Analysis of the Risk of Coronary Heart Disease in Framingham." *Journal of Chronic Diseases*, Vol. 20, 1967, pp. 511-524.
- Walker, H. M. and J. Lev. *Statistical Inference*. New York: Holt, 1953.
- Weisberg, H.F. *Dimensional Analysis of Legislative Roll Calls*. (Phd. Dissertation, The University of Michigan, 1968.
- Williams, J. D. "Two-way Fixed Effects of Variance with Disproportionate Cell Frequencies." *Multivariate Behavioral Research*, 1972, pp. 67-83.
- Winer, B. J. *Statistical Principles in Experimental Design*. Second edition. New York: McGraw-Hill, 1971. (Univariate.)

Acknowledgements

The work of developing, maintaining, and documenting the original versions of OSIRIS at the Institute for Social Research was done between 1967 and 1981 by groups headed by Gregory Marks, Judith Rattenbury, John Sonquist, Duane Thomas, and Neal A. Van Eck. The members of these groups, as well as others connected with the Institute, who were active in the design, maintenance, and documentation of OSIRIS software in that period include:

Ozer Babakol	Ellen Grun	Paula Pelletier
Elizabeth Baker	Carol Hafner	George Rabinowitz
Sylvia Barge	Richard Hanson	Judith Rattenbury
Nancy Barkman	Karen Jensen	Stewart Robinovitz
Marty Barrett	David Kappell	Donna Rocheleau
Paul Beck	David Karns	Richard Rutt
David Beckles	Sonya R. Kennedy	Edward Schneider
Ralph Bisco	Laura Klem	David Schupp
Carl Bixby	Ralph Koch	David Seigle
Tina Bixby	Cynthia R. Kruse	Merrill Shanks
Robert Burdette	Jeanne C. Kuo	Robert Sheffield
Bruce Campbell	Sary Luthra	Alice Snider
Jennifer Campbell	Spyros Magliveras	Peter Solenberger
Carol Cassidy	Gregory Marks	John Sonquist
Carol Damroze	Suzanne Marshall	Marianne Stover
Anne Davis	Judy Mattson	Glenn Tarsha
Karen Dickinson	Nancy Mayer	Dale Terrell
Marita DiLorenzi	Pamela Melton	Duane Thomas
George Dunn	Robert Messenger	Joanne Tiene
Lutz Erbring	Arthur Miller	Neal A. Van Eck
Asher Galed	Jack Miller	Stephen Vinter
Carolyn Geda	William Murphy	Vivian Wang
Terry Gleason	Pauline Nagara	Herbert Weisberg
Yildirim Gokturk	Michael Nash	Ingrid Weisz
Judith Goldberg	Neal Oden	Alice Yan
Kathleen Goode		

Many other members of the Institute's research staff made significant contributions to the strengthening of computer resources, including Frank M. Andrews, David G. Bowers, Robert D. Caplan, Jerome M. Clubb, Terrence N. Davidson, Martin Gold, Robert Kahn, John G. Lansing, David A. Lingwood, Warren Miller, James N. Morgan, Stanley E. Seashore, Donald E. Stokes, Burkhard Strumpel, Arnold S. Tannenbaum, James L. Wessel, and Stephen B. Withey.

i Srcware has a third module, SASMOD; options to output special SAS files are not used by IVWARE.

ii Srcware has a third module, SASMOD; options to output special SAS files are not used by IVWARE.