Introduction to DIGRAM

Svend Kreiner

Dept. of Biostatistics, University of Copenhagen

March 2003

Preface

DIGRAM is part of a larger statistical package, SCD, supporting "Statistical analysis of Categorical Data". The program is based on DIGRAM, a DOS program in Kreiner (1989) dedicated to analysis of high-dimensional contingency tables by block recursive chain graphical models. DIGRAM has since then been expanded to cover other types of models and problems as well, while still focusing on problems that could be solved complete or partly by simple tests for conditional independence. The complete SCD package is not finished yet. These notes therefore only describe DIGRAM.

The current version of DIGRAM supports the following types of problems:

Discrete graphical modeling

> Analysis of high-dimensional contingency tables by chain graph models
> Exact conditional tests for conditional independence
> Analysis of ordinal categorical data
> Analysis of independence graphs

Non-parametric loglinear analysis of ordinal categorical variables

Analysis of marginal and conditional homogeneity of repeated measurements

Stepwise MCA analysis of collapsibility across categories of multidimensional contingency tables

Analysis of multivariate Markov chains

Item analysis by graphical and log-linear Rasch models

> Analysis and modeling of item bias and DIF
> Analysis and modeling of local independence
> Analysis of local homogeneity

The purpose of these notes is first to give a short introduction to DIGRAM projects and the user interface of SCD and DIGRAM and second to give a brief presentation of the types of analyses implemented in DIGRAM. Additional details on what goes on during the analysis will be presented elsewhere[1].

SCD and DIGRAM may be freely downloaded from the author's homepage at www.biostat.ku.dk/~skm/skm/index.html

[1] A set of DIGRAM user guides (Kreiner 200?a-d) are being prepared. They may be downloaded from the author's homepage at www.biostat.ku.dk when they are ready.

Table of Contents

Graphical models

Introduction

We start this introduction to DIGRAM with a short prologue introducing some notation and results pertaining to graphical models and independence graphs.

A graphical model is a multivariate statistical model defined by a set of statements concerning conditional independence of pairs of variables given (some of) the remaining variables. The reason for calling the models *graphical* is that the models are characterised by mathematical graphs – networks where variables are represented by nodes some of which are connected by undirected edges or directed arrows. A missing edge or arrow between two variables implies that a certain conditionally independence statement applies for these two variables.

We distinguish between three different types of graphical models:

1) Graphical models for symmetrical relationships where all variables are assumed to be on the same footing.
2) Graphical regression models distinguishing between dependent variables and independent explanatory variables. Graphical regression models will typically be models for a multivariate set of dependent variables, but simple regression models with only one dependent variable is also included in this framework.
3) Block recursive graphical models – often called *chain graph models* - with variables in a series of recursive blocks where all influences between variables in different blocks are assumed to work in one direction only.

Graphical models for symmetrical relationships

The generic graphical model is a model of the joint distribution of a multivariate set of variables assuming symmetrical relationships among variables. These models in other words describe associations or correlations between variables rather than the effect of some variables on other variables.

Definition 1.

A graphical model is defined by a set of assumptions stating that certain pairs of variables are conditionally independent given the remaining variables of the model.

We write $X \perp\!\!\!\perp Y$ when X and Y are assumed to be independent and $X \perp\!\!\!\perp Y \mid Z$ when X and Y are conditional independent given Z in the sense that

$$P(X,Y|Z) = P(X|Z)P(Y|Z)$$

or equivalently

$$P(X|Y,Z) = P(X|Z).$$

Example 1.

Let A, B, C, D, E, F be six variables. The following four assumptions define a graphical model:

$$A \perp\!\!\!\perp C \mid B,D,E,F$$
$$A \perp\!\!\!\perp F \mid B,C,D,E$$
$$B \perp\!\!\!\perp D \mid A,C,E,F$$
$$D \perp\!\!\!\perp F \mid A,B,C,E$$

Note that all statements concerning local independence involve four conditioning variables. We will return to the question whether we always need to condition with the rest of the variables for two variables to be conditionally independent.

The model is called a *graphical* model because the defining properties are encapsulated in a mathematical graph, a set of nodes and edges between nodes, as shown in Figure 1.

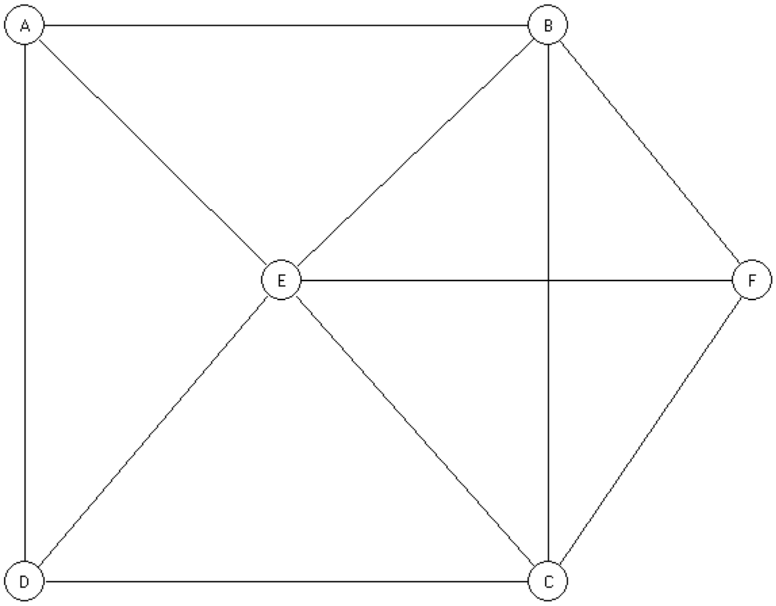


Figure 1. Independence graph defined by the following four assumptions of conditional independence:

$$A \perp\!\!\!\perp C \mid B,D,E,F$$
$$A \perp\!\!\!\perp F \mid B,C,D,E$$
$$B \perp\!\!\!\perp D \mid A,C,E,F$$
$$D \perp\!\!\!\perp F \mid A,B,C,E$$

We call graphs defining graphical models for either independence graphs or interaction graphs. Visual representations of networks are not a new thing in multivariate statistics. They have been used systematically for path analysis and structural equation models for many years before graphical models even existed. Independence graphs are unusual, however, because they are mathematical graphs with properties that can be determined using graph theoretical algorithms to identify properties of the statistical model. Independence graphs in this sense are second order mathematical models. They are graph theoretical models of probabilistic models encoding some, but not all properties of the first order model.

The theory of graphical models will be assumed known by the reader. If not, you are urgently advised to consult some of the following references

Whittaker (1990),
Edwards (1995,2000)
Lauritzen (1996)
Cox and Wermuth (1996)

Short introduction to the subject may be found in Edwards & Kreiner (1983), Whittaker (1993), Wermuth (1993) and Kreiner (1996).

The main properties of independence graphs for discrete[2] data can be summarised in the following way:

Graphical models for discrete data are loglinear. If all variables are discrete then the model is a model for a multidimensional contingency table. Darroch, Lauritzen and Speed (1980) in their seminal paper on graphical models show that a graphical model for discrete variables is a loglinear model with generators defined by the cliques[3] of the independence graph. The model shown in Figure 1 is thus a loglinear model with generators, ABE, ADE, CDE, BCEF. Note that maximal order of interaction among variables in cliques is always assumed.

Graphical models collapse onto graphical marginal models. Collapsing over some of the variables of the model in Figure 1 will always lead to a new graphical model. Some of the unconnected variables in the complete graph may, however, have to be connected in the independence graph of the marginal model.

[2] Graphical models may be defined for both continuous variables and for mixed continuous and discrete variables. The basic definition of graphical models is the same for these types of data, but some of the properties of graphical models for discrete data differs to some extent from the properties of mixed discrete and continuous variables.

[3] A clique in a graph is a maximal subset of complete connected nodes.

The conditional distribution of a subset, of the variables given the remaining variables will be a graphical model. Let V be the complete set of variables of the model. If $V_1 \subset V$ then the independence graph of $P(V_1 \mid V \backslash V_1)$ is equal to the subgraph[4] of V_1.

The separation theorem: Separation implies conditional independence. Two subsets of variables, U and V, are separated by a third subset if all paths from a variable in U to a variable in V move through at least one variable in W. When this happens, U and W are conditionally independent given W: $U \perp\!\!\!\perp V \mid W$. In Figure 1 D and F are separated by both (A,E,C) and (B,C,E). It follows that $D \perp\!\!\!\perp F \mid A,E,C$ and $D \perp\!\!\!\perp F \mid B,C,E$. Conditional independence implied by separation is usually referred to as *global* Markov properties.

Separation implies parametric collapsibility in loglinear models. The separation theorem for separate variables is a consequence of a more general result. If all *indirect* paths between two variables, X and Y, move through at least one variable in a separating subset, S, then the model is parametrically collapsible unto the marginal XYS-table in the sense that all parameters pertaining to X and Y are the same in the complete model and in the marginal model, P(X,Y,S). This means that the association parameters relating to A and D in Figure 1 is the same in the complete P(A,B,C,D,E,F) models and in the marginal models of P(A,D,B,E) and P(A,D,C,E) because all indirect paths from A to D goes through both (B,E) and (C,E).

Marginal models sometimes have a simpler parametric structure than implied by the marginal graphical model. This result is implied by parametric collapsibility. In Figure 1, the marginal graphical model for P(A,D,B,E) is saturated. Parametric collapsibility implies that the ADC-parameters are constant across different levels of B and C not only in the complete model, but also in the marginal model. Therefore, P(A,D,B,E) must be a loglinear model defined by the following generators: ADE,ABE,DEB.

Decomposition by separation of complete subsets[5] leads to decompositions of statistical models implying collapsibility in terms of likelihood inference for certain types of

[4] A subgraph consists of a subset of nodes connected by the same edges as in the complete graph.

models. In Figure 1, the BED clique separates (A,D) from F implying that (A,D) and F are conditional independent given (B;C,E). The joint distribution can therefore be written as

$$
\begin{aligned}
& P(A,B,C,D,E,F) \\
&= P(A,D|E,B,C)P(F|E,B,C)P(B,C,E) \\
&= \frac{P(A,B,C,D,E)P(B,C,E,F)}{P(B,C,E)}
\end{aligned}
$$

For discrete models, the maximum likelihood estimate of the parameters of the loglinear graphical model, Figure 1, will result in estimates of P(B,C,E) which is equal to the observed marginal frequencies in the marginal BCE-table. Maximising the likelihood therefore becomes a question of maximising P(A,B,C,D,E) and P(B,C,E,F) separately. Parameters pertaining to A and D will be estimated in the marginal ABCDE table while parameters relating to F will be estimated in the BCEF-table. The model therefore collapses into two components with respect to likelihood inference for AD-parameters and F-parameters respectively.

The topography of marginal models.

The collapsibility implied by decomposition by complete separators motivate the following definitions:

Let V be a subset of variables defining a marginal model while W is the subset of variables not included in the marginal model. We call W the *exterior* of V.

Assume that the subgraph of W consists of a set of unconnected components, W = $W_1,...,W_r$. For each W_i, we define the *boundary*, $B_i \subset V$, as all variables in V connected

[5] A subset of nodes is complete if all nodes of the subset are connected in the graph.

directly to at least one variable in W_i. It follows that all B_i are complete in the independence graph of the marginal model for P(V).

The *border, B,* of V is the union of all boundaries, $B = \cup_i B_i$

Variables in V\B and edges attached to at least one of these variables or to variables from two different boundaries will be referred to as the *interior* of V, V\B.

It follows from the results on parametric collapsibility that all parameters relating to the interior of a marginal model are the same as in the complete model. If all boundaries are complete in the complete graph then likelihood inference concerning these variables will be the same in the marginal and the complete model. Parameters of the complete model describing relationships between variables in the same boundary of V are not accessible by analysis of the variables of V. We refer to the edges between variables in a given boundary as *fixed* edges because no analysis of V can justify that these edges are removed from the complete model.

Finally we define the *core* of a given problem defined by certain parameters of the complete model as the smallest marginal model where likelihood inference gives the same results as in the complete model. Results on decomposability of graphical models imply that there will always be one smallest core for any given model. In the model Figure 1, the core of the CF problem is the marginal P(B,C,E,F) model.

Graphical regression models

A graphical regression model is a multidimensional multiple regression model of the conditional distribution, P(Y | X) where Y and X are vectors of dependent and independent variables, $Y = (Y_1,..,Y_r)$, and $X = (X_1,..,X_s)$. Conditional independence assumptions define graphical regression models in the same way as for ordinary graphical models. The assumptions are, however, restricted to assumptions concerning two dependent variables or one dependent and one independent variables,

$$Y_i \perp\!\!\!\perp Y_j \mid Y_1,..,Y_{i-1},Y_{i+1},..,Y_{j-1},Y_{j+1}, .,Y_r,X_1,..,X_s$$
$$Y_i \perp\!\!\!\perp X_j \mid Y_1,..,Y_{i-1},Y_{i+1},..,Y_r,X_1,..,X_{j-1},X_{j+1},..,X_s$$

Apart from the restriction on the permissible set of assumptions independence graphs for graphical regression models are defined and play exactly the same role as in the basic graphical models discussed above. Figure 2 shows the independence graph for a regression model with three dependent and five independent models. Edges connecting independent variables have been drawn as fat lines to show that these edges are fixed in the model. Removing one of these edges would imply that two independent variables are conditionally independent given the remaining independent and all dependent variables – a statement that has no bearing on the conditional distribution, P(Y | X).

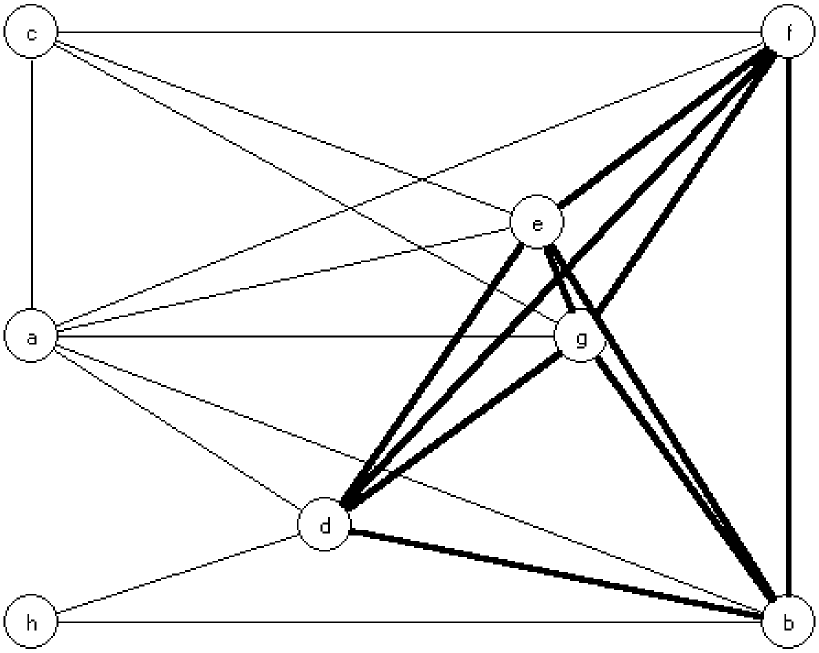


Figure 2. Independence graph for a graphical regression model of P(a,c,h|b,d,e,f,g).

Chain graph models

A block recursive model partitions a vector of variables V = ($V_1,\ldots,V_k$) into a number of subsets of variables ($U_1,\ldots,U_r$) such that $U_i \bigcup U_j = Ø$ for all (i,j) and $V = \bigcup_i U_i$ into and writes the joint distribution of all variables as a product of conditional distributions:

$$P(V) = \prod_1^{r-1} P(U_i \mid U_{i+1},..,U_r) \cdot P(U_r)$$

The models are called block recursive because each component may have more than one variable.

We call the different blocks of a block recursive model for different recursive *levels* and say that $Y \in U_i$ is at a *higher* level than $X \in U_j$ if i<j. A block recursive model thus describes the effect of variables at lower labels on variables at higher levels.

This takes us to the definition of chain graph models.

Definition 2
A chain graph model is a block recursive model where $P(U_r)$ is a graphical model and each of the components $P(U_i \mid U_{i+1},..,U_r)$ are graphical regression models.

Chain graph models are characterized by independence graphs where variables in at different recursive levels are connected by arrows pointing from lower to higher levels while variables at the same recursive level are connected by undirected edges. It follows from Definition 2 that two variables are unconnected if they are conditionally independent of all other variables at the same or lower levels.

Figure 3 shows one such model with eight variables at three different levels.

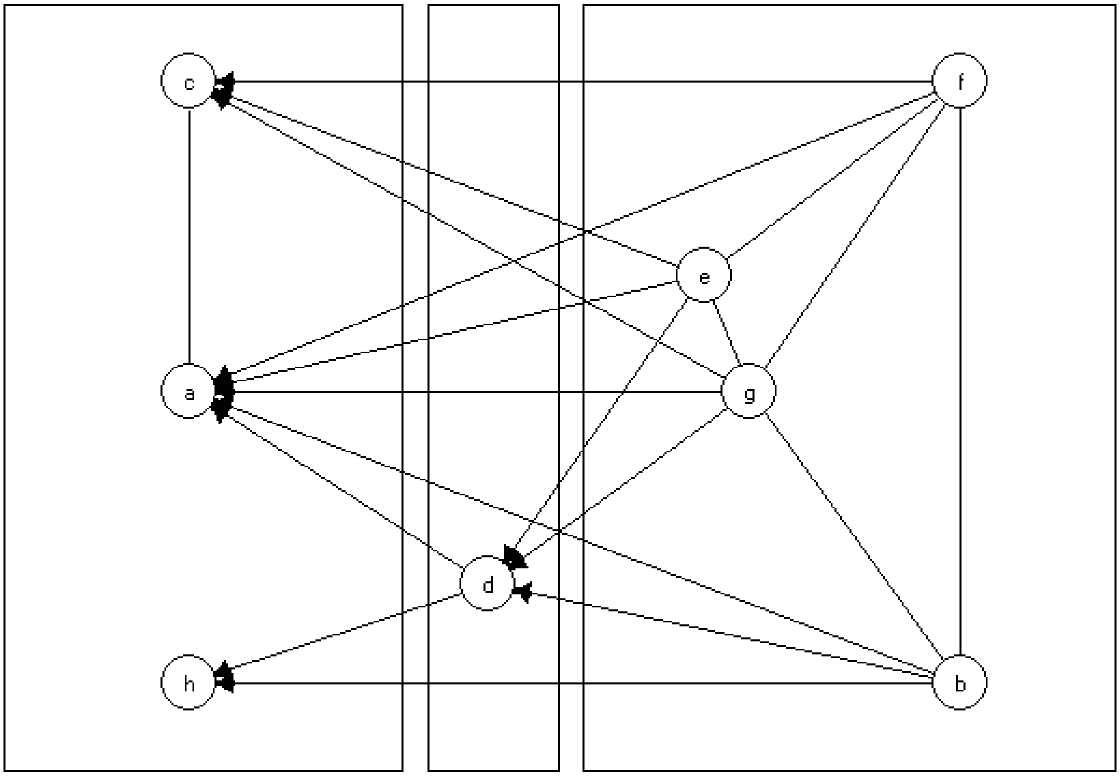


Figure 3. Chain graph model for a model with eight variables at three different levels. Boxes have been drawn around each recursive block in order to put emphasis on the recursive structure.

The model in Figure 3 is based on the following assumptions:

Recursive structure:

$$P(a,b,c,d,e,f,g,h) = P(a,g,h \mid d,b,e,f,g) \cdot P(d \mid b,e,f,g) \cdot P(b,e,f,g)$$

Conditional independence at Level 1:

$a \perp\!\!\!\perp h \mid b,c,d,e,f,g$ $c \perp\!\!\!\perp b \mid a,d,e,f,g,h$ $c \perp\!\!\!\perp d \mid a,b,e,f,g,h$ $c \perp\!\!\!\perp h \mid a,b,d,e,f,g$

$h \perp\!\!\!\perp e \mid a,b,c,d,f,g$ $h \perp\!\!\!\perp f \mid a,b,c,d,e,g$ $h \perp\!\!\!\perp g \mid a,b,c,d,e,f$

Conditional independence at Level 2:

$d \perp\!\!\!\perp f \mid b,e,g$

Conditional independence at Level 3:

$b \perp\!\!\!\perp e \mid f,g$

A chain graph model may also be characterized by the independence graphs of the graphical (regression) models for each of the different levels of the model. The model, Figure 3, in this way is equivalent to two regression graphs and one conventional independence graph. The regression graph for Level 1 was shown in Figure 2. The remaining independence graphs are shown in Figures 4 and 5.

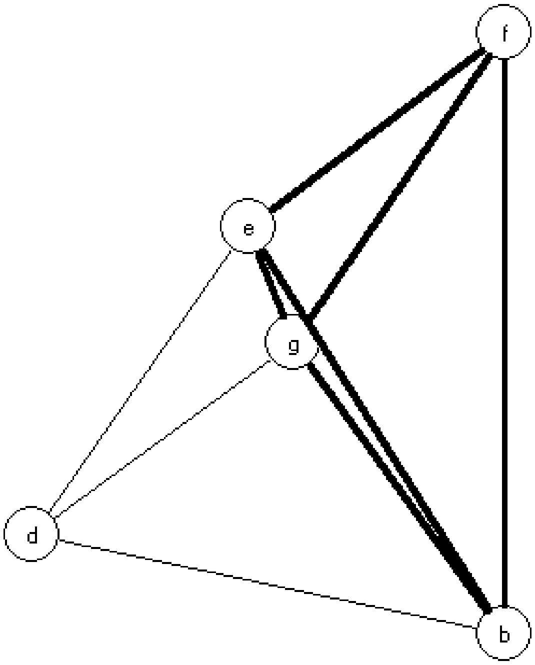


Figure 4. Independence graph for the graphical regression model at Level 2.

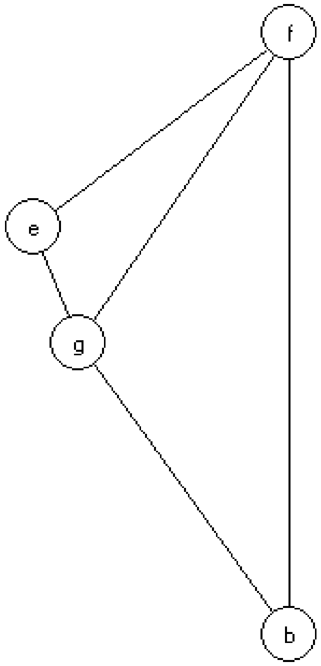


Figure 5. Conventional independence graph for Level 3.

The results concerning separation and decomposition described for conventional independence graphs extend to the (undirected) regression graphs and are therefore also applicable for analysis of chain graphs. It can be shown that additional results concerning separation and decomposition apply to the directed chain graphs if each recursive level only has one variable. We refer to Lauritzen (1996) for a systematic discussion of these results.

Graphical modelling

While analysis of a 5-8 dimensional contingency table is certainly not trivial, it is nevertheless fairly straightforward and has been discussed extensively in many different contexts. What DIGRAM offers for this kind of analysis is an approach - exact conditional tests and tests for ordinal categorical data - that addresses some of the well-known technical problems with which this analysis is burdened (sparse tables, low power of tests, etc.).

Analysis of high-dimensional data is a totally different world. To survive this kind of ordeal, you have to have a very explicit strategy formulated before the analysis starts and you have to recognize the difference between the primary substantive research problems and secondary problems related to completion of statistical models. Given these requirements and given that the primary problems may be operationalized as problems of association, we believe that the family of chain graph models is a good frame of inference and that DIGRAM may be of considerable help for your analysis.

An idealized overall strategy for analysis of high-dimensional data should at the very least consist of the following components:

1) An initial analysis of data (screening) aimed at formulation of a complete base model that you may use as a starting point for your analysis.

2) Further specification and simplification of this model using appropriate exploratory model search strategies.

The above two steps should specifically address secondary problems of model building.

3) The definitive or specific analysis aimed at the substantive research problems and formulation of conclusions. This is not - except in special cases - something that you should approach in an exploratory manner.

Procedures for model checking and for analysis of collapsibility properties of models onto marginal tables are integrated by DIGRAM in all steps of the above strategy.

The two sets of assumptions underlying the recursive graphical models are treated quite differently during the statistical analysis.

Assumptions on recursive structure have to be formulated before the statistical analysis. They are regarded as part of the theoretical foundations determined partly by assumptions on causal structure and partly by the design of the study (which of course cannot be changed by the statistical analysis).

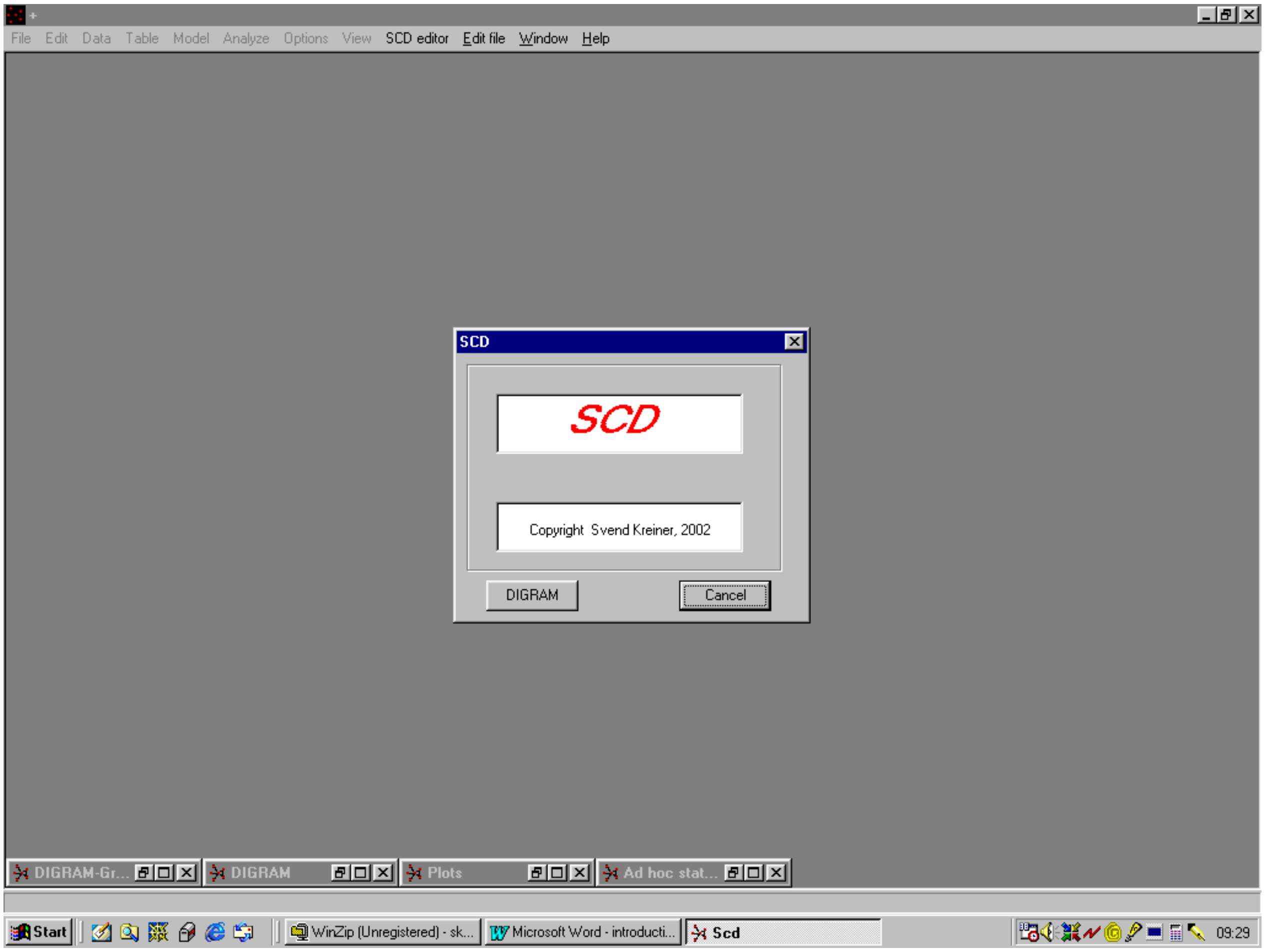
Assumptions on conditional independence may to a certain extent be specified as a starting point for the analysis in which case it will be one purpose of the analysis to check on and improve these assumptions. The typical situation will however be one, where assumptions state that variables are associated (conditional *de*pendence).

Secondary problems will typically be concerned with simplification and parsimony. It will be the purpose of the analysis addressing these problems to search for evidence against conditional independence among secondary variables. If no evidence is found we usually assume that no direct interaction exists.

The substantive (primary) research problems will also be stated in terms of both conditional association/dependence and conditional independence. The typical primary research hypothesis assumes that variables are associated. It is the purpose of the analysis to find evidence supporting association for which reason the focus of the final specific analysis will not in general be on parsimony and simplification. Quite the contrary, in fact.

The user interface

You can start SCD the same way you start any Widows-bases application. Figure 6 shows the opening window of the SCD environment:



Copyright Svend Kreiner, 2002

Figure 6. The SCD window

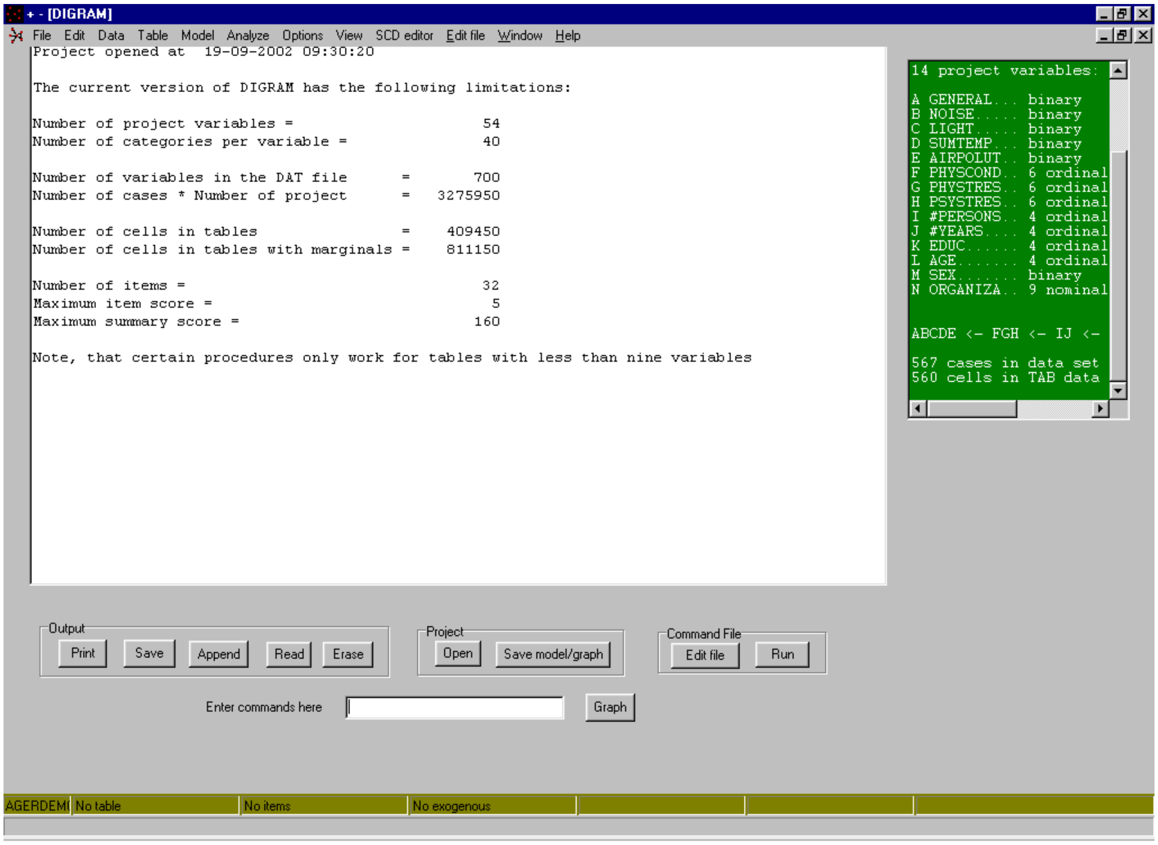
DIGRAM contains two main windows:

- DIGRAM's main form where the statistical analysis of data is carried out,
- DIGRAM's Graph form where different types of interaction graphs are shown, edited and analyzed,

and several smaller dialog forms that will be used during the analysis of data.

The DIGRAM environment

DIGRAM is a command driven program with a graphical user interface with some menus and buttons that in some cases may make life a little easier for you. Figure 7 shows DIGRAM’s regular interface. In addition to the menus and buttons the form contains a large output window, where all output from the program will be written, a smaller (green) project window displaying some details on the current DIGRAM project and a small command field. Commands written in the command field at the bottom of the window will be executed after you enter carriage return (CR) if the field is in focus.



```
Project opened at  19-09-2002 09:30:20

The current version of DIGRAM has the following limitations:

Number of project variables =                    54
Number of categories per variable =              40

Number of variables in the DAT file      =      700
Number of cases * Number of project      =  3275950

Number of cells in tables                =   409450
Number of cells in tables with marginals =   811150

Number of items =                                32
Maximum item score =                              5
Maximum summary score =                         160

Note, that certain procedures only work for tables with less than nine variables
```

```
14 project variables:

A GENERAL...  binary
B NOISE.....  binary
C LIGHT.....  binary
D SUMTEMP...  binary
E AIRPOLUT..  binary
F PHYSCOND..  6 ordinal
G PHYSTRES..  6 ordinal
H PSYSTRES..  6 ordinal
I #PERSONS..  4 ordinal
J #YEARS....  4 ordinal
K EDUC......  4 ordinal
L AGE.......  4 ordinal
M SEX.......  binary
N ORGANIZA..  9 nominal


ABCDE <- FGH <- IJ <-

567 cases in data set
560 cells in TAB data
```

Figure 7. DIGRAM’s main form. The output window shows the limitations of the current version of DIGRAM.

The buttons and menu items should be almost self-explanatory. You can at any time during the analysis

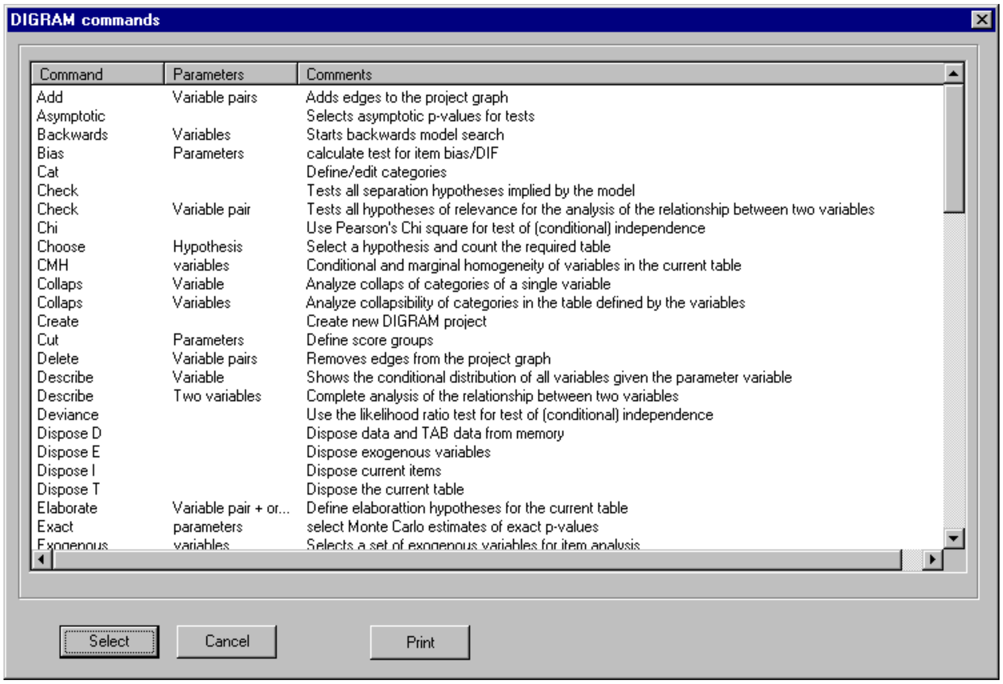
- print, save append, read and erase output
- open a new project
- save the current graphical model
- edit and/or execute commands on command files.

The program uses standard windows controls for reading or saving files, setting up printers etc.

To my experience menus rarely if ever are used, as commands are much more convenient for interactive statistical analyses and buttons are much faster to work with for routine handling of input and output. Menus are nevertheless included for compatibility with other windows programs. You may look through menus to get an idea about the things that are supported by DIGRAM and use them if you prefer to do so, but they will not be described in detail in these notes.

Another way to get an idea about what you can do with DIGRAM is to enter a HELP command. The result is the list of all the commands implemented in DIGRAM shown in Figure 8.

The information shown in the output window in Figure 7 is the information on limitations of the current version of DIGRAM. This information will always be shown when you start the program. If you erase this information and need to remind yourself about these limitations at a later point, you must use a “SHOW L” command. “SHOW P” will inform you about known errors in the program and how to avoid them (if possible) while “Show E” will display information on the DIGRAM environment including information on the arrays allocated by the program.



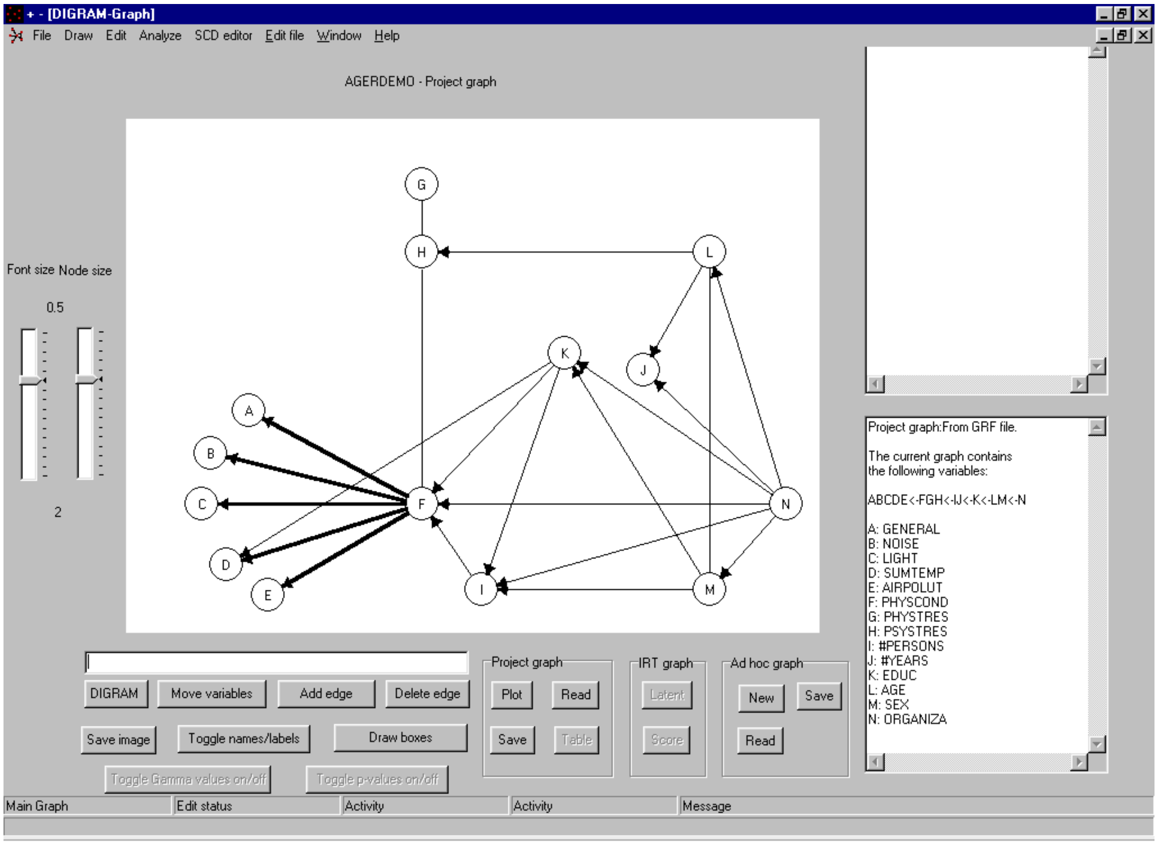
Command	Parameters	Comments
Add	Variable pairs	Adds edges to the project graph
Asymptotic		Selects asymptotic p-values for tests
Backwards	Variables	Starts backwards model search
Bias	Parameters	calculate test for item bias/DIF
Cat		Define/edit categories
Check		Tests all separation hypotheses implied by the model
Check	Variable pair	Tests all hypotheses of relevance for the analysis of the relationship between two variables
Chi		Use Pearson's Chi square for test of (conditional) independence
Choose	Hypothesis	Select a hypothesis and count the required table
CMH	variables	Conditional and marginal homogeneity of variables in the current table
Collaps	Variable	Analyze collaps of categories of a single variable
Collaps	Variables	Analyze collapsibility of categories in the table defined by the variables
Create		Create new DIGRAM project
Cut	Parameters	Define score groups
Delete	Variable pairs	Removes edges from the project graph
Describe	Variable	Shows the conditional distribution of all variables given the parameter variable
Describe	Two variables	Complete analysis of the relationship between two variables
Deviance		Use the likelihood ratio test for test of (conditional) independence
Dispose D		Dispose data and TAB data from memory
Dispose E		Dispose exogenous variables
Dispose I		Dispose current items
Dispose T		Dispose the current table
Elaborate	Variable pair + or...	Define elaborattion hypotheses for the current table
Exact	parameters	select Monte Carlo estimates of exact p-values
Exogenous	variables	Selects a set of exogenous variables for item analysis

Figure 8. The list of DIGRAM commands produced by the HELP command. Click first on a command and second on the Select button the command to transfer the command to the Command field of DIGRAM's main form. Click on the Print button to print the list of commands on I the output window

Click on the Graph button on DIGRAM's main form (or enter a GRAPH command) to replace the main form with DIGRAM's main form shown in Figure 9.

The Graph form in many ways resembles the main form. You communicate with the program using commands, buttons and menu items. The form contains a graph window showing either the main project graph, a graph representing a graphical or loglinear Rasch model or an ad hoc graph that you have defined yourself. The main graph is of course the default graph.

At the right of the form you will find two small fields. An output window at the top and a graph window at the bottom showing the variables included in the current graph. A number of commands are available for manipulation of the graph. These will be described in a subsequent chapter on editing, displaying and analyses of independence graphs. The form also contains a number of buttons that you may use to edit graph and track bar for redefinition of the size of the nodes in the graph. The functionality of these items will be described in Appendix A on the graph editor.



Project graph:From GRF file.

The current graph contains
the following variables:

ABCDE<-FGH<-IJ<-K<-LM<-N

A: GENERAL
B: NOISE
C: LIGHT
D: SUMTEMP
E: AIRPOLUT
F: PHYSCOND
G: PHYSTRES
H: PSYSTRES
I: #PERSONS
J: #YEARS
K: EDUC
L: AGE
M: SEX
N: ORGANIZA

Figure 9. DIGRAM's graph form

To return to DIGRAM's main form you can either click on the DIGRAM button or enter a DIGRAM command in the command field of the GRAPH form.

DIGRAM projects

Figures 7 and 9 show the show the main form and the graph form immediately after a DIGRAM project called ‘AGERDEMO’ has been opened. The project uses data from a study of working conditions in Denmark using 14 out of a much larger data set. The variables and some information on the recursive structure among variables and the number of cases included are shown on the main form (Figure 7). The current graphical model is shown on the graph form (Figure 9). Behind this information lies a number of ASCII files containing both the original and recoded data, variable and category definitions, an incidence matrix defining the interaction graph, and some output that it has been convenient to save during the analysis.

The files are in general defined by the project name followed by an extension defining the contents of the file. The following file types define the standard files associated with a project:

- The DEF file, project.DEF, contains the basic information needed to define the project.
- The DAT file, project.DAT, contains the data set. The name of the DAT file is the only project file name that you may select yourself after you have decided on a project name. If you do not want to use the default name, the alternative name must be defined on the DEF file.
- The VAR file, project.VAR, contains the definitions of the variables included in the project. Notice that the project is not required to use all the variables on the DAT file. Information on cases to be excluded from the project must be written on the DEF file.
- The CAT file, project.CAT, contains category definitions.
- The SYS file, project.SYS, contains a recoded data matrix. Only variables used by the project are included.
- The TAB file, project.TAB, contains the same information as the SYS file in a more compact table format.

- The GRF file, project.GRF, contains information on the current graphical model.
- TMP files, DIGRAM.TMP and project.TMP, used by DIGRAM during the analysis. These files will in most cases be erased by DIGRAM when you exit the program.

In addition to the files mentioned above you might have additional files containing output and/or ad hoc graphs associated with a given program. You are free to call these files whatever you like to call them, but they will only be recognized, as belonging to the project if the file name is equal to the project name.

It is assumed that all files except DIGRAM.TMP are saved in the same folder as the DEF file. You may save the files elsewhere during the analysis, but DIGRAM will not be able to find the files for you if you do so.

IMPORTANT: When you start the program the current version of DIGRAM will try to create the temporary file, DIGRAM.TMP in the same folder where you have saved the SCD program. You therefore must not install SCD in a folder where you do not have permission to write.

The DEF, DAT and VAR files must be defined when you open a project for the first time. CAT, SYS and TAB files will be created if DIGRAM cannot find these files. The GRF file will be defined during the analysis if you decide to save the current model. If the model was revised but not saved during the analysis you will be asked whether you want to save the model, when you exit the program.

All files are ASCII files that you may read and edit using any kind of standard text editor.

The contents and format of the files is described below, together with information on how to change the contents during the analysis. Examples will be given using the files associated with the UKU project enclosed among the examples distributed with DIGRAM.

The data for the UKU project originated in a study of side effects in a comparative drug trial. The data were analyzed and presented by Lingjærde et.al. (1987) and reanalyzed by Kreiner and Edwards (1990) using recursive graphical models and the DIGRAM program.

The study concerned two different antidepressants: Drug 1 without anticholinergic effect and Drug 2, which has a marked anticholinergic effect. Side effects were measured prior to treatment and again after one two and three weeks on active drug. The project only includes information on a single item from the so-called "UKU side effect rating scale" describing side effect by four ordinal categories:

0: not present
1: mild degree
2: moderate degree
3: severe degree

Additional information on the UKU scale may be found in Lingjærde et.al. (1987).

The name for this project is UKU. This name is used for the default names of the files shown below.

Data definitions: The DEF file.

The DEF file defines the data matrix in the DAT file in terms of

- the total number of variables/columns per case (not more than 700 per case in the DAT file),
- filters or conditions for selection of cases.

The first record of the DEF file must contain the number of variables in the dataset (not the number of variables you are going to use in your analysis. Subsequent records should contain filter conditions (one per record) given by

- a variable number,
- a minimum value
- a maximum value.

There is no limit to the number of filters that you may include on the DEF file. Only cases, for who all values of filter variables belong to the intervals defined by the corresponding minimum and maximum values (both included), will be used in the analysis.

Finally you may include information on certain files used by DIGRAM and change some of the default values of DIGRAM parameters. Most of these facilities are reminiscent of the DOS version, but obsolete in the current version for Windows and only included in order to make the program backwards compatible. The only important feature that you need to know about is the specification of alternative data files. To change the name of the DAT file associated with the program you must enter two lines on the DEF file:

```
FILES
D <DAT file name>
```

The first line tells the program that information on files will follow on the next line while the "D" on the next line indicates that the following name is the name of the DAT file.

The DEF file of the UKU project contains nothing but the number of columns in the data file (5). If we had only been interested in a project containing cases without side effects prior to treatment and if the data file was not called UKU.DAT, but CASES.DAT then the contents of the DEF file should have been as shown in Figure 10

```
5
2 0 0
FILES
D CASES.DAT
```

Figure 10. The contents of a hypothetical DEF file for a UKU project of cases with no side effects prior to treatment and with data from a file called CASES.DAT

Note that side effects prior to treatment is the fourth variable of the project according to the VAR file below.

The data matrix: The DAT file.

The DAT file contains the data matrix from which variables for the analysis may be extracted. The file should be organized as a regular data matrix - one case at a time - as a *space* delimited[6] ASCII file with each new case beginning on a new record. There are no restrictions on the number of records per case, nor on the way the variables are distributed across these records. It is only required that variables appear in the correct order and that there are no more than 700 variables per case.

Only numerical values are permitted. Non-numerical codes will be treated as missing values[7]. Text variables with several words delimited by spaces are not permitted.

The first ten records of the UKU.DAT are shown in Figure 11.

```
1 0 0 0 0
1 1 1 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 1 0 0
```

Figure 11. The first six records of the UKU.DAT file

[6] Carriage Return (CR) may be used to separate variables instead of spaces.

[7] Or rather: If a non-numerical code is encountered it will be given a value of –999999, which typically will be regarded as missing because it is lower than the minimum value of the variable ranges defined in the VAR file.

Note, that you have to prepare your DAT file using some kind of standard program. You also have to take care of missing values for the variables, using numerical codes beyond the ranges defined in the VAR file.

Variables: The VAR file.

The VAR contains information defining the variables included in the project. These variables do no have to appear in the same order in the project as in the DAT file:

1. The number of variables in the analysis (no more than 40 in the present version of the program).
2. Definition of project variables. To define a project variable the following information is required:

   A *variable label* - a single letter, which may be used for later reference to the variable during the analysis.

   A *column number* referring to the data matrix from which the variable is selected.

   The *number of categories* defining the project variable.

   The *variable type* (2:nominal/3:ordinal categories).

   Minimum and maximum values of the original variable and one or more *cut points* defining the categories of the project variables. Cut points always define intervals including upper category boundaries.

3. The number of recursive blocks.
4. Cutpoint partitioning the project variables into the recursive blocks.
5. Up to 10 lines of Comments. (Optional).
6. Variable names up to 8 characters per variable (optional, but recommended).

Comments and variable names may be interchanged. Otherwise the information should be given in exactly the order presented above.

A couple of conventions and rules must be followed to when the VAR file is written:

- The *number of project variables* should appear on a single record at the top of the file.
- The information on project variables should appear *space-delimited* on two records per variable:

  Record 1: <Label> <Column number> <Number of categories> <Type>

  Record 2: <Minimum> <Cutpoint(1)>...<Last cutpoint> … <Maximum>

- Standard variable types are:

  2: nominal categories

  3: ordinal categories

- The intervals defining categories contains the cutpoints:

Category 1:	Minimum	≤	values	≤ cutpoint(1)
Category 2:	Cutpoint(1)	<	values	≤ cutpoint(2)
	etc. etc. until			
The Last category:	The last cutpoint	<	values	≤ Maximum

- All values less than minimum or greater than maximum will be regarded as missing values.
- The *number of recursive blocks* should appear on a separate record.
- Variables must be defined in the correct order with the ultimate responses first and explanatory variables later. This means that recursive blocks may be reference to recursive cut points indicating the last variables in the recursive blocks. The recursive cut point may appear on separate records or on one record separated by blanks. If the last recursive cut point is less than the number of project variables, the remaining variables will always be treated as purely explanatory variables.

The final part of the VAR file contains *comments* and *variable names.*

Variable names must be initiated by a record containing the word "VARIABLES" (or at least the first three characters), followed by records containing variable labels and names separated by blanks.

Only the first 8 characters of the variable names will actually be used, but you may of course include as many as you like on the VAR file.

A record containing "COMMENTS" will be taken to indicate that the contents of following records should be treated as comments. Only the first ten lines of comments will be used.

Immediately after having read the last recursive cut point, the program will be in COMMENT mode and will remain so until "VARIABLES" appear.

Figure 12 shows the contents of the UKU.VAR file.

```
5
D 5 3 3
0 0 1 3
C 4 3 3
0 0 1 3
B 3 3 3
0 0 1 3
A 2 3 3
0 0 1 3
T 1 2 2
1 1 2
4
1 2 3 5
Ratings of side effects in a comparative drug treatment.
Side effects are rated pre-treatment (STATUS 0), one, two
and three weeks after treatment (STATUS 1-3).
Ratings were scored:
0 : not present      1 : mild degree
2 : moderate degree  3 : strong degree
The last two categories have been collapsed for the
analysis by DIGRAM.
Treatment was by one of two drugs.
VARIABLES
T TREATMENT
A STATUS 0
B STATUS 1
C STATUS 2
D STATUS 3
```

Figure 12. The contents of the UKU.VAR file defining five project variables

The UKU.VAR file defines variables in the following way:

Notice first that we are going to use all 5 variables of the data set. We did not have to do that, but could have requested an analysis of a subset of variables. The order of the project variables is not however the same as the order columns in the data matrix.

The first project variable is given the label D:

```
D 5 3 3
0 0 1 3
```

This variable is the fifth variable in the data matrix. The variable has three ordinal categories defined by collapsing of categories 2 and 3 of the original variable.

The categories of D are given by

Minimum = 0
Cut points = 0 and 1
Maximum = 3.

The three categories of D are therefore defined by the following values of variable 5 in the original data matrix:

Category 1 : 0 (not present)
Category 2 : 1 (mild degree)
Category 3 : 2-3 (moderate to strong degree)

The name of variable D may be found in the list of variable names at the bottom of the file. Variable D is "STATUS 3", that is the measure of side effects after three weeks on drugs.

Five variables, D, C, B, A, T are defined in this way. The column numbers of the original data matrix are 1, 2, 3, 4, 5.

The recursive structure is defined immediately after the variables in terms of the reference numbers:

```
4
1 2 3 5
```

We have four recursive levels or blocks defined by four recursive cut points in the sequence of variables.

Level 1 : D
Level 2 : C
Level 3 : B
Level 4 : A and T.

The recursive structure reflects underlying temporal structure of the variables:

$$D \leftarrow C \leftarrow B \leftarrow AT$$

Categories: The CAT file.

The CAT file contains category names for all project variables. CAT files are optional, but recommended. They may be created and modified before you start DIGRAM or at any time during an analysis.

Each record of a CAT file must contain a label of a project variable a value indicating a category and the corresponding category name. The category names on the file may be as long as you wish, but DIGRAM will only use the first eight characters.

```
D 1 None
D 2 Mild
D 3 mod+strong
C 1 None
C 2 Mild
C 3 mod+strong
B 1 None
B 2 Mild
B 3 mod+strong
A 1 None
A 2 Mild
A 3 mod+strong
T 1 Drug1
T 2 Drug2
```

Figure 13. The contents of the UKU.CAT file

Graphs: The GRF file.

The GRF files will in most cases be created by DIGRAM so you do not really have to worry about the way the information on the graph is formatted. The format of the file is free with blanks or <carriage return> as delimiters.

The *first* record should have

The number of variables/vertices
The number of recursive levels

The *second* record should have the cut points defining the recursive blocks in terms of the sequence of variables in the same way as in the VAR file.

The *third* record contains variable labels in the correct order.

The *next records* define the status matrix. The matrix should be symmetrical with status codes given by:

1 : conditional independence - no edge
2 : potential dependence - an edge will be included
3 : conditional dependence - a fixed edge included

Status 3 point to a fixed edge, where a decision has been made, that two variables are locally dependent. None of the models searching strategies implemented in DIGRAM will consider associations with fixed edges.

Status codes above 3 will indicate an edge for which special information is available. These codes will be used by certain automatic procedures of DIGRAM.

The *last part* of the GRF file may have information on the coordinates of the nodes in the graph plot. This information must be initialized with a record containing the word "COORDINATES" followed by a record for each NODE containing

The label
the x-coordinate on the screen
the y-coordinate.

One of the graphs considered for the UKU study is shown in Figure 14 below.

```
5 4
1 2 3 5
D C B A T
3 3 1 1 1
3 3 3 1 1
1 3 3 3 3
1 1 3 3 1
1 1 3 1 3
COORDINATE
S
D 32 102
C 96 102
B 160 102
A 223 102
T 289 64
```

Figure 14. A UKU.GRF file

Note that labels and recursive structure are the same as in the UKU.VAR file.

The graph assumes conditional independence between D and B, D and A and D and T. We have conditional independence between C and A and T, and we have independence between A and T. (The model implied is a Markov chain of order 1 including a short term effect of treatment).

GRF files for ad hoc graphs

You may at any time draw ad hoc graphs and save them on GRF files having the same format as the project GRF file. The only difference between an ad hoc GRF file and the project graph is that information on variable names are included on the ad hoc GRF file in the same way as variable names are included on the VAR files.

The facilities for dealing with ad hoc graphs will be described in the notes on “Manipulation of DIGRAM files”.

Files with recoded data: The SYS and TAB files

The first thing that happens when DIGRAM attempts to open a new project, is that data is extracted and recoded from the DAT file. The recoded data is saved on two different files. First as a recoded data matrix on the SYS file and second as a table of counts on the TAB file.

Categories are always enumerated from 1 to maximum number of categories on the SYS and TAB files while zero is reserved as the code for missing values. The TAB file contains records for all cells with positive counts in the multivariate contingency table created by all the project variables. Each cell in the table is defined by a record containing category values followed by the observed number of cases in the cell.

Figure 15 and 16 shows (part of) the contents of the SYS and TAB files. It is seen on the TAB files that 27 cases with Drug 1 and 10 cases with Drug 2 has no side effect at all.

```
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
2 1 1 1 1
3 1 1 1 1
```

Figure 15. Five records from the UKU.SYS file

```
1 1 1 1 1 27
1 1 1 1 2 10
1 1 1 2 1 3
1 1 2 1 1 4
1 1 2 1 2 2
.. .. ..
3 3 2 1 1 1
3 3 2 1 2 1
3 3 2 3 2 1
3 3 3 1 2 2
3 3 3 3 2 1
```

Figure 16. The first five lines and the last five records on the UKU.TAB file

Creating DIGRAM projects

The bad news is that you have to create DIGRAM projects yourself. It is a common experience shared by most applied statisticians that moving data from a standard statistical program like SAS, SPSS or STATA to a specialized standalone program like DIGRAM takes inordinately long time. Work is in progress developing programs and procedures that may ease the transition, but until this is finished you are on your own.

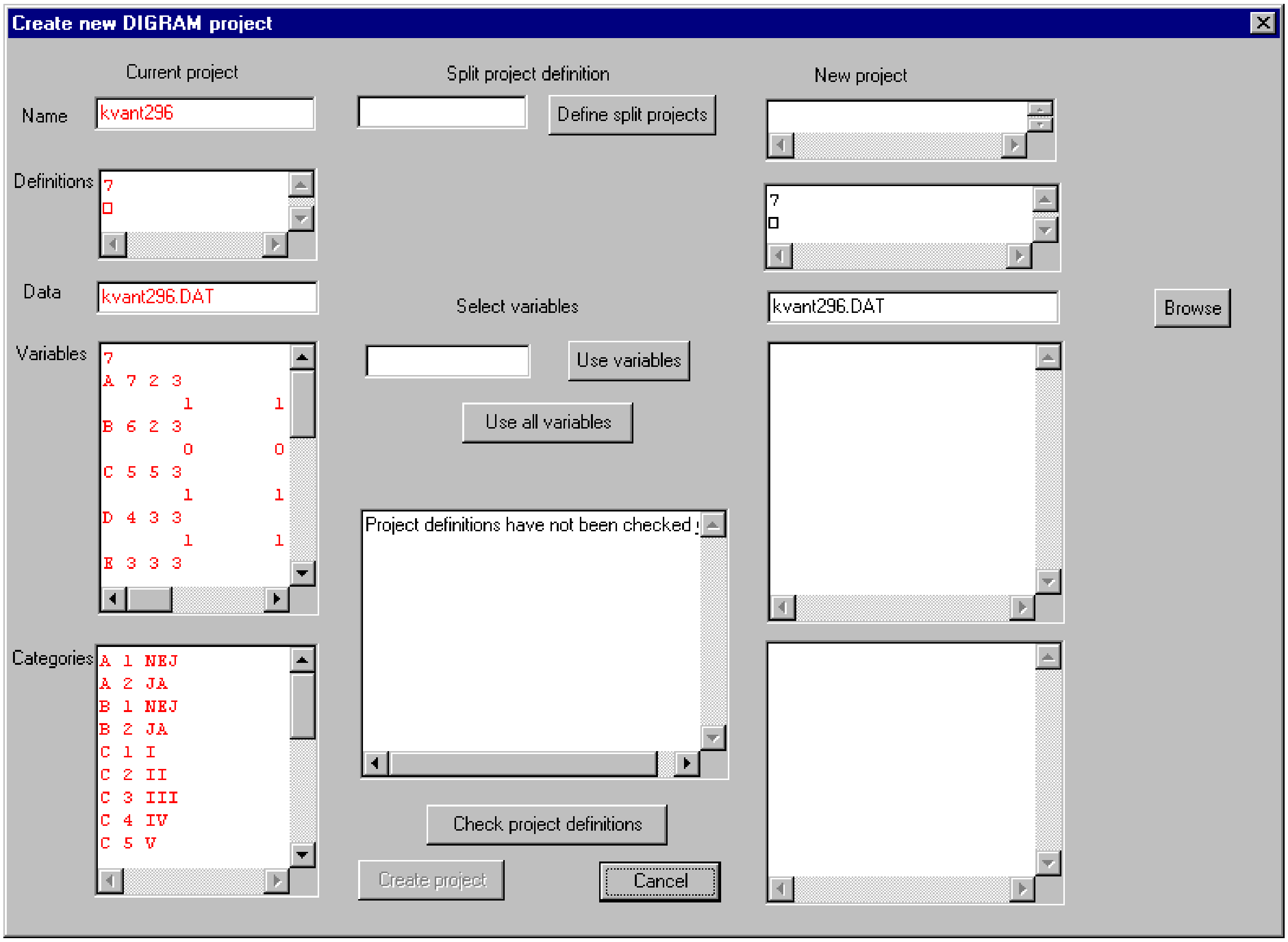
We do believe that the work is manageable. You have to remember that

1) DIGRAM only reads standard free formatted ASCII text files,
2) DIGRAM only accept numerically coded data (including numerical codes for missing values).

Variable definitions may seem cumbersome, but they are really not that bad once you have tried it a few times. To make things easier three commands are available opening a project definition dialog, where you can develop and test variable definitions before they are realized. The three commands are

EXPORT variables	which exports variables from the current to a new project,
SPLIT variable	which creates so-called split projects by stratification according to the levels of the split variable,
CREATE	Which lets you open a create a new project from scratch.

All three commands open the same dialog shown in Figure 17.



Name: kvant296

Definitions:

```
7
□
```

Data: kvant296.DAT

Variables:

```
7
A  7  2  3
       1        1
B  6  2  3
       0        0
C  5  5  3
       1        1
D  4  3  3
       1        1
E  3  3  3
```

Categories:

```
A  1  NEJ
A  2  JA
B  1  NEJ
B  2  JA
C  1  I
C  2  II
C  3  III
C  4  IV
C  5  V
```

Project definitions have not been checked !

New project:

```
7
□
```

kvant296.DAT

Figure 17. The "Create new DIGRAM project" dialog

The project definitions of the current project will be shown in the column to the left if a project is already open when the dialog, Figure 17, is invoked. It will be assumed that the DAT file of the new project will be the same as the DAT file of the current project.

Exporting variables from the current to a new project

To export variables from the current to a new project you must write the labels of the variables you want to use in the "Select variable" field and then press the "Use variables" button. A star among the export variables will leave room open among the variables in the new project, where you can define a new variable. You are also welcome to change both the definitions of the variables that you have selected for export and the filters used for the current project.

DIGRAM will transfer category definitions for all the variables you have decided to export when you press “Use variables”. You can (and should if new variables have been included) edit the category definitions, if you want to do so.

Defining split projects

To define split projects enter a label in the “Split project definition” field and press the “Define split projects” button. DIGRAM will assume that you want to export all variables except the split variable to the new project, and will therefore write all the labels of the variables in the “Select variable” field. You can, however, edit this field removing variables or adding ‘*’s for new variables before you press “Use variables”.

Creating new projects

You can of course create completely new projects if you want to do so. Fill out all information – project names, information on DAT names, variable definitions and category names – check the validity of definitions and press “Define project”. The project must be saved in the same folder as the DAT file. To be sure that this happens you should use the file browser to identify and select the DAT file.

Checking definitions

Press the “Check project definitions” when you think the project definitions are ready. If the definitions are usable, the “Create Project” will be enabled. Pressing this button finally creates all the files you need for the new project, including a GRF file copying the parts of the current model that can be reused in the new project.

A short guided tour

We will use the UKU project on side effects in a comparative drug trial described above to illustrate the basic features of DIGRAM. The project contains information on repeated ordinal measurements of side effects after treatment with categories going from no side effects to severe side effects. The analysis of these data will attempt to answer the following questions:

1) Do the two different treatments differ with respect to the risk of side effects?
2) Is the difference between treatments a short term and/or a long term phenomena?
3) Do the risk of side effects increase or decrease with time?

To answer these questions we will attempt to develop three different types of models:

1) A chain graph model.
2) A Markov Chain model.
3) A generalized Rasch model taking heterogeneity of persons into account.

During the guided tour of the analysis of the UKU project you will learn how to

a) describe data,
b) revise variable definitions,
c) create and analyze tables,
d) define, display and analyze graphical models,
e) generate tables and hypotheses by analysis of graphs,
f) test model based hypotheses,
g) select models,
h) check models,
i) describe relationships,
j) analyze data by loglinear models,
k) analyze data by Markov chain models,
l) analyze data by generalized Rasch models.

The guided tour is intended to give you a rough idea about DIGRAM's capabilities. Technical details will be skipped and many commands will not be discussed here. The complete set of commands will (in time) be described in the user guide and technical details documented in separate papers dedicated to special topics.

Examining and describing data

SYS and TAB files will be generated the first time DIGRAM opens a new project. Before you proceed with the analysis you should however examine data to check whether variables have been properly defined. Three commands are available for initial examination and description of data:

FREQUENCIES produces marginal frequencies for all variables (Figure 18).

DESCRIBE variable generate tables showing the conditional distribution of the remaining project variables given the variable referred to by the parameter of the DESCRIBE command (Figure 19).

COLLAPS <variable> Examines whether or not some of the categories of the variable given as a parameter to the COLLAPS command are collapsible in the sense that differences between conditional distributions of other variables given these variables (Figure 20). Collapsibility across categories will be examined for all polytomous variables if the COLLAPS command is issued without parameters.

Remember that you only have to include the first three characters of commands.

Note that DESCRIBE and COLLAPS with more than two variables result in somewhat different analyses then those described here.

Parts of the description of the UKU data are shown in Figures 18 to 20 below. P-values associated with Goodman and Kruskall's γ coefficients are here as elsewhere one-sided p-values. We recognize that the use of one-sided p-values is somewhat unorthodox, but trust that you as a user will be able to multiply p-values with two whenever two-sided p-values are appropriate.

```
Reference no.   1
Variable no.    5
D: STATUS 3

    D COUNT     PCT  CUMPCT
----------------------------
    0    59   59.00   59.00
    1    27   27.00   86.00
    2    13   13.00   99.00
    3     1    1.00  100.00

TOTAL   100

This variable was
categorized

Categories    count   pct
 1     None      59  59.0
 2     Mild      27  27.0
 3 mod+stro      14  14.0
```

Figure 18. ***FREQUENCIES:*** *Marginal frequencies are generated for all variables. This figure only shows the frequencies for the first project variable appearing in column 5 of the original data matrix. Frequencies are presented first for the variable of the original data matrix and second for the categorized project variable*

```
Conditional distributions given TREATMEN(T)

                     TREATMEN(T)
                    Drug1 Drug2       Current status
--------------------------------------------------------------
                        %     %
         STATUS 3(D)               ***     direct relationship

                None  76.5  43.1
                Mild  15.7  37.3
            mod+stro   7.8  19.6

               Total    51    51
-------------------------------------------------------------
***: At least one p-value less than or equal to 0.001
 **: At least one p-value less than or equal to 0.01
  *: At least one p-value less than or equal to 0.05
```

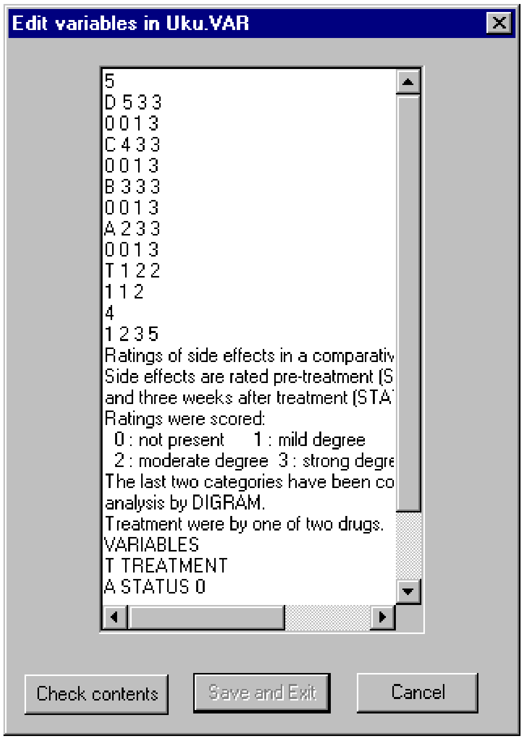
Figure 19. ***DESCRIBE T:*** *Conditional distribution of project variables given treatment (T). Only the very first distribution is shown here. The current status refers to the relationships between variable defined by the current graphical project model.*

```
+++++   Collaps of categories  ++++++
        D:STATUS 3     Categories =  1-2  3

Test against     chi**2   df   p     gamma     P
   C STATUS 2       0.5    2 0.798   0.172  0.260
   B STATUS 1       3.5    2 0.173   0.448  0.034     +
   A STATUS 0       1.5    2 0.479  -0.110  0.370
   T TREATMEN       0.0    1 0.886   0.048  0.443

+++++   Collaps of categories  ++++++
        D:STATUS 3     Categories =  1  2-3

Test against     chi**2   df   p     gamma     P
   C STATUS 2       1.1    2 0.584   0.255  0.197
   B STATUS 1       0.9    2 0.643  -0.273  0.176
   A STATUS 0       1.1    2 0.574   0.250  0.238
   T TREATMEN       0.9    1 0.332   0.371  0.156
```

Figure 20. ***COLLAPS D:*** *Examination of collapsibility across categories of variable D. Collapsibility across categories is examined in all two-way tables relating variable D to other variables. For ordinal variables collapsibility will only be considered for adjacent categories. Some evidence against collapsibility of the first two categories is found in the BD-table (γ = 0.448, p = 0.034).*

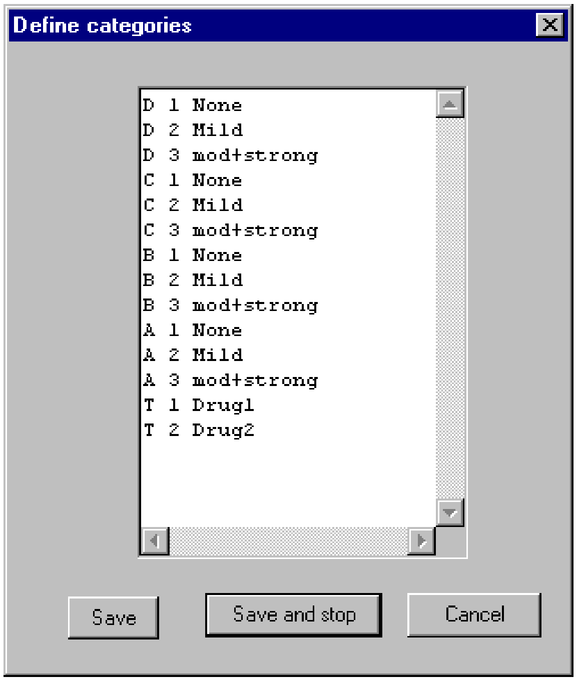
The main purpose of the data description illustrated above is to examine whether there is any reason to change the definition of project variables. If this seems to be the case, you can change variable definitions selecting either the Data|Revise variables menu or invoking the VARIABLES command. This will result open the dialogue shown in Figure 21, where you may edit the contents of the VAR file.

Changing variable definitions may imply that category definitions also have to be modified. This will not happen automatically so you have to take care of this yourself. Select Data|Define/Revise categories or issue a CATEGORIES command to open the edit dialogue shown in Figure 22.



```
5
D 5 3 3
0 0 1 3
C 4 3 3
0 0 1 3
B 3 3 3
0 0 1 3
A 2 3 3
0 0 1 3
T 1 2 2
1 1 2
4
1 2 3 5
Ratings of side effects in a comparativ
Side effects are rated pre-treatment (S
and three weeks after treatment (STA
Ratings were scored:
  0 : not present      1 : mild degree
  2 : moderate degree  3 : strong degre
The last two categories have been co
analysis by DIGRAM.
Treatment were by one of two drugs.
VARIABLES
T TREATMENT
A STATUS 0
```

Figure 21. ***VARIABLES*** *will open this dialogue, where variable definitions may be edited. You have to check the contents of the new variable definitions before you can save the results. New SYS and TAB files will be generated if the new variable definitions indicate that this is necessary.*



```
D 1 None
D 2 Mild
D 3 mod+strong
C 1 None
C 2 Mild
C 3 mod+strong
B 1 None
B 2 Mild
B 3 mod+strong
A 1 None
A 2 Mild
A 3 mod+strong
T 1 Drug1
T 2 Drug2
```

Figure 22. ***CATEGORIES:*** *Change categories in any way you want to. The current version of DIGRAM will not check the consistency of variable and category definitions.*

Creating and analyzing multidimensional contingency tables 1: Tests of conditional independence

DIGRAM is first of all a program for analysis of multidimensional contingency tables. In this section we will show you the basics of creating and analyzing tables. You will learn how to

1) tabulate data and display tables,
2) create and test hypotheses of conditional independence,
3) fit loglinear models,
4) test for marginal and conditional homogeneity of repeated measurements,
5) test for collapsibility across categories in multidimensional tables.

An important distinction is the difference between model-based and model-free tables. DIGRAM will fit chain graph models to the complete set of project variables. Collapsibility properties of these models generates marginal tables and models in which specific problems may be addressed, and you at any point of the analysis ask for tables generated by the model in this way. You are however not restricted to looking at model-based tables. You can of course generate any table you want to and ask for any analysis of this table, as long as the problems to be addressed respect the recursive structure of the data.

This section only describes model-free analyses of tables. Model based analysis will be described after the next section on definition and modification of graphical models.

In the example in this section we disregard the measurement of side-effects prior to treatment focusing instead on a four-way tables with treatment and the three repeated measurements after treatment. The first problem we will address in connection with this table is the question of the degree to which the third measurement (D) depends on the first when both treatment (T) and the second measurement (C) is taken into account.

The table we need to create thus is the DCBT-table. The hypothesis that we will test is the hypothesis of conditional independence of D and B given C and T,

$$D \perp\!\!\!\perp B \mid C,T$$

Two commands are needed to create the table and the hypothesis:

TABULATE DCBT Creates the table[8].

HYPOTHESIS DB Creates the hypothesis.

If you want to see the table before you do anything with it you have to enter

SHOW T Prints the table in a fairly compact format.

[8] Previous versions of DIGRAM permitted only tables with up to eight variables. This limitation has been relaxed, but tables are still limited with respect to the number of cells in the table. Use **SHOW L** to get information on the limitations in DIGRAM.

You can also get a look at the table if you add a T-, R- or C-parameter. To the TEST command described below.

The results of using these commands are shown in Figure 24 below. During tabulation the following things happen:

1) The table is counted.
2) All margins are calculated.
3) The marginal graphical model for the table is determined.

All edges/arrows representing association parameters in the marginal model are fixed if the complete model is not parametrically collapsible onto the marginal model with respect to the parameters represented by the edges/arrows in the model. You can of course estimate the degree of association for all edges in the marginal model and test conditional independence between variables connected by fixed lines, but these results will have no bearing on the complete project model.

To understand what goes on when the marginal model is created some knowledge about the complete model is necessary. In the example discussed in this section the graphical model shown in Figure 23 was used as the project model. Whether or not this model is adequate will be examined later in these notes.

Figure 24 shows the information on the marginal table and model and the hypothesis generated by the above three commands.

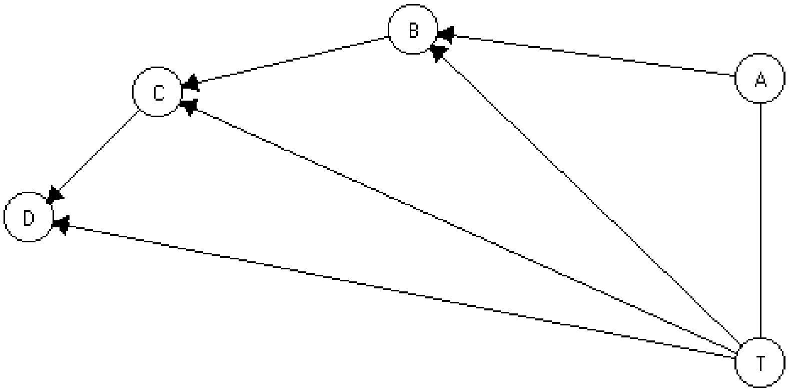
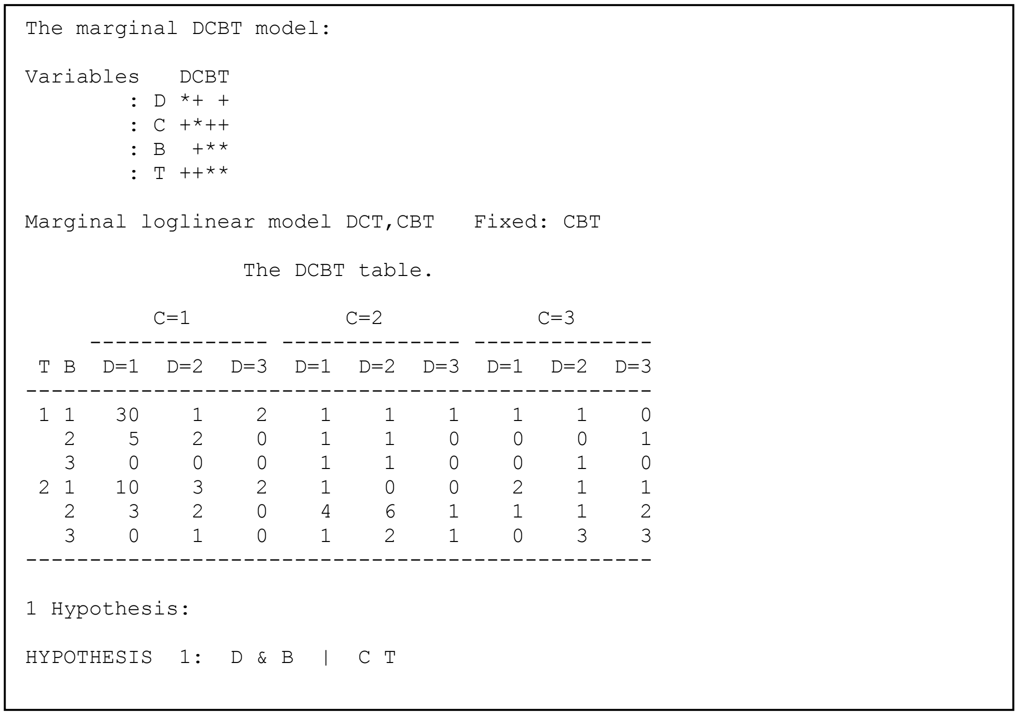


Figure 23. The graphical model used as the project model in the guided DIGRAM tour



```
The marginal DCBT model:

Variables   DCBT
        : D *+ +
        : C +*++
        : B  +**
        : T ++**

Marginal loglinear model DCT,CBT   Fixed: CBT

                 The DCBT table.

          C=1            C=2            C=3
     -------------- -------------- --------------
 T B  D=1  D=2  D=3  D=1  D=2  D=3  D=1  D=2  D=3
-------------------------------------------------
 1 1   30    1    2    1    1    1    1    1    0
   2    5    2    0    1    1    0    0    0    1
   3    0    0    0    1    1    0    0    1    0
 2 1   10    3    2    1    0    0    2    1    1
   2    3    2    0    4    6    1    1    1    2
   3    0    1    0    1    2    1    0    3    3
-------------------------------------------------

1 Hypothesis:

HYPOTHESIS  1:  D & B  |  C T
```

Figure 24. ***TABULATE DCBT, SHOW T and HYPOTHESIS DB***

The three commands needed to generate the table and hypotheses are almost self-explanatory. Once the table has been generated you may of course define all the hypotheses you would care to test in the given table.

The parameters to the HYPOTHESIS command may consist of:

- Pairs of variables: several hypotheses will be defined.
- V* where V is a variable label, defines all possible hypotheses of conditional independence between V and other variables.
- V- generates hypotheses of conditional independence of V and all variables not associated with V in the marginal model for the table.
- V+ generates hypotheses of conditional independence of V and all variables that are associated with V in the marginal model for the table.

In addition to HYPOTHESIS one additional command, ELABORATE[9], generates hypotheses of conditional independence between two variables given some or all the remaining variables in the table.

Once hypotheses have been defined you have to issue a TEST command to perform the test of significance. The way the test works and presents itself depends however on a number of different conditions and parameters, that we have to describe before we can look at test results.

Test statistics:

For ordinal and binary data DIGRAM will always use the partial γ-coefficient[10] to measure and test monotonous relationships. P-values are always one-sided, so you may have to multiply by two, whenever you think a two-sided p-value would be more appropriate.

[9] The format of the ELABORATE command is as follows: ELABORATE V1 V2 n
Where V1 and V2 are two variables and n is the largest number of conditioning variables included in the hypotheses. ELAB DB 1 thus generates three hypotheses: $D \perp\!\!\!\perp B$, $D \perp\!\!\!\perp B|C$, $D \perp\!\!\!\perp B|T$. ELABORATE DB 2 includes the hypothesis shown in Figure 24. The elaboration hypotheses may be useful if one wants to examine the conditions under which an association between two variables disappears.

For nominal data you may choose between χ^2 tests and likelihood ratio deviances. The χ^2 test is the default option, but you can change to likelihood ratio tests if you want to. The commands required for switching between test statistics are

DEVIANCE selects likelihood ratio tests

CHISQUARE selects χ^2 tests

Asymptotic or exact p-values

P-values based on the asymptotic distribution of test statistics are unreliable for tests of conditional independence in large sparse tables. Instead of asymptotic p-values you may select p-values based on the exact conditional distribution of test statistics given the sufficient margins under the null-hypothesis[11]. P-values calculated by DIGRAM are Monte Carlo estimates of these p-values and thus still approximations of the exact p-values. The difference between these and the asymptotic p-values are however that they are unbiased and that you control the precision of the approximate p-values yourself.

Two commands let you switch between asymptotic and exact p-values:

EXACT <nsim <seed>> Selects exact tests. NSIM is the number of tables generated for the Monte Carlo estimates. The default is 1000 tables. Seed is the seed for the random number generator. The default is 9.

ASYMPTOTIC Selects asymptotic tests. Asymptotic tests are the default option for statistical tests.

[10] The relationship between two binary variables can also be measured by the odds ratio statistic and the Mantel-Haenszel statistic. There is however a one-to-one relationship between Goodman and Kruskall's marginal γ-coefficient, so one is as good as the other. Things are not quite that simple for the partial γ-coefficients and the Mantel-Haenszel statistic but both statistics can be viewed as weighted sums of respectively stratified γ coefficient and stratified odds-ratio statistics with no clear indication of one being better than the other. We have therefore not cared to sacrifice the generality of the γ-coefficients in order to introduce odds-ratio statistics as an option for binary variables. This may however change in latter versions of DIGRAM.

[11] See Kreiner (1987) for details of exact conditional tests in multidimensional contingency tables.

Global or local test results

Test results consist of global summary test statistics and stratified local results where the test of conditional independence is performed in all strata defined by values of a single conditioning variable.

The default option is global test only. If you also want local results you first have to invoke the LOCAL command:

LOCAL	Selects local test results.

To deselect global test results you must enter the GLOBAL command:

GLOBAL	Selects global tests only.

Note that DIGRAM will stay in local test mode until the GLOBAL commands has been given and visa versa.

Having defined the parameters for the statistical tests you may enter the TEST command with parameters determining whether or not the table should be printed as well as the test results:

TEST	Prints only test results
TEST T	Prints the table (with relative row frequencies) and test results
TEST O	Prints the table with observed counts and test results
TEST R	Prints the table (with relative row frequencies) and test results
TEST C	Prints the table (with relative column frequencies) and test results
TEST E	Prints the table (with expected counts[12]) and test results

Note that the test mode will change to local if you want to print the table.

[12] Expected counts are calculated under the hypothesis of conditional independence being tested.

The parameters of the TEST command may be combined. TEST RCE prints a table with both expected values and relative row and column frequencies.

Figure 25 presents the test results for the hypothesis shown in Figure 24 while Figure 26 shows part of the table produced after a TEST T command.

```
****  Summary of test results  ****

NSIM = 1000 tables generated for exact p-values

--------------------------------------------------------------------------------------
                         p-values                           p-values
Hypothesis       X²  df asymp exact               Gamma asymp exact               nsim
--------------------------------------------------------------------------------------
 1:D&B|CT      22.5  22 0.432 0.617 (0.577-0.656)  0.39 0.040 0.024 (0.014-0.040) 1000
--------------------------------------------------------------------------------------
-----------------------------------------------------------
** Local testresults for strata defined by STATUS 2 (C) **
                          p-values               p-values
 C: STATUS 2   X²    df asympt  exact  Gamma asympt  exact
-----------------------------------------------------------
 1:    None   9.57     6 0.1437 0.1740   0.40 0.1263 0.0840
 2:    Mild   4.04     8 0.8535 0.9510   0.32 0.1870 0.1900
 3:mod+stro   8.85     8 0.3551 0.6070   0.46 0.0588 0.0900
-----------------------------------------------------------
-----------------------------------------------------------
** Local testresults for strata defined by TREATMEN (T) **
                          p-values               p-values
 T: TREATMEN   X²    df asympt  exact  Gamma asympt  exact
-----------------------------------------------------------
 1:   Drug1  12.25   10 0.2685 0.4070   0.42 0.1733 0.1800
 2:   Drug2  10.21   12 0.5974 0.6490   0.38 0.0456 0.0480
-----------------------------------------------------------
```

Figure 25. ***EXACT, LOCAL, TEST:*** The precision of the Monte Carlo estimates of exact p-values is indicated by 99 % confidence limits for the global test statistics.

The local test statistics will also be summary test statistics unless there is only one conditioning variable. The local γ-coefficient of 0.40 in the strata defined by no side effects at the second measurement (C=1) is thus a partial γ-coefficient for D and B given treatment T. Figure 26 shows the two strata contributing to the local test statistics for C = 1. Note that separate statistics are presented for each strata of the table shown after a TEST T command. P-values presented for isolated strata of the table is always asymptotic p-values.

```
Conditioning variables:

+---------------------------+
|  C STATUS 2 |  T TREATMEN |
|-------------+-------------|
|  1     None |  1    Drug1 |
|  2     Mild |  2    Drug2 |
|  3 mod+stro |             |
+---------------------------+

+TREATMEN
|+STATUS 2
||     +STATUS 1
||     | | D:STATUS 3         |
TC     B |  None  Mild mod+s  | TOTAL |
---------+--------------------+-------+
11  None |    30     1     2  |    33 |
     row%|  90.9   3.0   6.1  | 100.0 |
    Mild |     5     2     0  |     7 |
     row%|  71.4  28.6   0.0  | 100.0 |
   mod+s |     0     0     0  |     0 |
     row%|   0.0   0.0   0.0  |   0.0 |  X² =   5.7
--------------------------------------+  df =   2
   TOTAL |    35     3     2  |    40 |   p = 0.058
     row%|  87.5   7.5   5.0  | 100.0 | Gam =  0.52
--------------------------------------+   p = 0.159

      some strata have been deleted here

--------------------------------------+
21  None |    10     3     2  |    15 |
     row%|  66.7  20.0  13.3  | 100.0 |
    Mild |     3     2     0  |     5 |
     row%|  60.0  40.0   0.0  | 100.0 |
   mod+s |     0     1     0  |     1 |
     row%|   0.0 100.0   0.0  | 100.0 |  X² =   3.9
--------------------------------------+  df =   4
   TOTAL |    13     6     2  |    21 |   p = 0.423
     row%|  61.9  28.6   9.5  | 100.0 | Gam =  0.22
--------------------------------------+   p = 0.288
    some strata have been deleted here
--------------------------------------+
```

Figure 26. The two strata of the DB|CT table where C = 1. The remaining strata have been deleted.

Analysis of multidimensional contingency tables 2: Marginal and conditional homogeneity.

If you have repeated ordinal or nominal measurements or if you have comparable measurements of different phenomena on the same nominal or ordinal scale you can test both marginal and conditional homogeneity. Two commands result lead to analyses of homogeneity:

HOMOGENEITY <T>	Tests homogeneity of the two variables of interest in the current hypotheses if these variables have the same number of categories and the same scale type. The T option produces tables in the same way as for the TEST command.
CMH variables	Results in tests of marginal homogeneity if the list of variables include all variables summarized in the table. If only a subset of variables is included in the list of variables, CMH results in tests of conditional homogeneity of the variables given the remaining variables and in tests of marginal association between the remaining variables and the repeated measurements.

Technical details concerning marginal and conditional homogeneity will be discussed elsewhere[13]. The purpose of this section is simply to point out that these facilities are available. Notice also, that we are still developing the CMH analysis. The appearance of output from this procedure will probably change before it is completed.

Figure 27 below shows the analysis of marginal homogeneity for a two-way tables produced by the HOMOGENEITY command while Figure 28 shows part of the analysis of conditional homogeneity in a three-way table.

[13] A technical report on "Marginal and conditional homogeneity" is under preparation.

```
D and B are assumed to be associated

     +STATUS 1
     | | D:STATUS 3         |
     B |  None  Mild mod+s | TOTAL |
-------+-------------------+-------+
  None |    44     7     6 |    57 |
Expecte|  44.1   7.7   6.2 |  58.0 |
  Mild |    13    12     4 |    29 |
Expecte|  12.0  12.1   3.8 |  28.0 |
 mod+s |     2     8     4 |    14 |
Expecte|   1.9   8.2   3.9 |  14.0 |
-----------------------------------+
 TOTAL |    59    27    14 |   100 |
Expecte|  58.0  28.0  14.0 | 100.0 |
-----------------------------------+
   R/C | Marginal freqs    | Total |
-------+-------------------+-------+
   Row |    57    29    14 |   100 |
   row%|  57.0  29.0  14.0 | 100.0 |
   Col |    59    27    14 |   100 |
   row%|  59.0  27.0  14.0 | 100.0 |  X² =   0.2
-----------------------------------+  df =   2
 TOTAL |   116    56    28 |   200 |   p = 0.923
   row%|  58.0  28.0  14.0 | 100.0 | Gam = -0.03
-----------------------------------+   p = 0.377

------------------------------------------------
Analysis of homogeneity - dependent variables
------------------------------------------------
** Global results ***
------------------------------------------------
Homogeneity deviance: G² =     0.2 df =    2 p =  0.919
                      X² =     0.2 df =    2 p =  0.923
Sign test: N(+) = 17   N(tie) = 60   N(-) =    23
             X² =    0.9 p =  0.343
          Gamma = -0.030 p =  0.377
------------------------------------------------
```

Figure 27. Test of marginal homogeneity of UKU side-effects at the first (B) and third (D) measurement after treatment.

Comments on Figure 27:

Two tables are shown. First, the two-way table of B and D and second, a two-way table with the two margins to compared.

Three principally different sets of test statistics are calculated, none of which give evidence against marginal homogeneity:

The first set of test statistics is compares the observed margins in the second table directly without fit of any specific model, but evaluates significance under the assumption that the association between measurements is as observed in the first table. The two test statistics here is Everitt's (1977) chi squared statistic and a γ coefficient measuring the degree of association in the second table. Agresti (1984, p. 208-209) suggests a Mann-Whitney test instead of the γ coefficient. We notice that the two statistics are equivalent for comparison of two repeated measurements, but prefer the γ coefficient because it can be generalized both to more than two measurements and used in connection with partial γ coefficients for analysis of conditional homogeneity.

Second a deviance, G^2, between the observed table of count and a fitted table under the assumption of marginal homogeneity. The first table of Figure 27 shows both the observed and fitted table. The estimate of fitted values and the deviance based on this estimate is discussed in Kreiner (2001).

Finally a chi squared sign test is also reported.

Comments on Figure 28:

Tables comparable to the tables of Figure 27 are produced but not here. The separate test statistics are calculated and summarized as global chi squared statistics and partial γ coefficients. In addition to the global statistics, stratified statistics are also presented in the same way as for tests of conditional independence. In the example here, where there was only one conditioning variable, the stratified statistics are the same as the separate test statistics, which would have been reported with each subtable, if they had been included in Figure 28.

There is no evidence at all against conditional homogeneity.

Note, that there is no difference between the homogeneity deviance and Everitt's X^2 statistic in these examples. This is not always the case, but one may conjecture of course, that the two statistics are asymptotically equivalent.

```
Conditioning variables:

+-------------+
|  T TREATMEN |
|-------------|
|  1    Drug1 |
|  2    Drug2 |
+-------------+

------------------------------------------------
Analysis of homogeneity - dependent variables
------------------------------------------------
** Global results ***
------------------------------------------------
Homogeneity deviance: G² =     0.5 df =    4 p =  0.974
                       Q =     0.5 df =    4 p =  0.975
Sign test: N(+) = 17   N(tie) = 60   N(-) =    23
             X² =    0.9 p =  0.343
          Gamma = -0.039 p =  0.356
------------------------------------------------

  +----------------------------------+
  |                                  |
  | Marginal homogeneity in T-strata |
  |                                  |
  +----------------------------------+
                                            Sign test
T df    G²  p       X²  p    Gamma  p     N+  N-   p
----------------------------------------------------------
1  2   0.4 0.827   0.4 0.830 -0.03 0.427   7   9  0.617
2  2   0.1 0.945   0.1 0.946 -0.04 0.371  10  14  0.414
----------------------------------------------------------
```

Figure 28. Test of condition homogeneity of UKU side-effects at the first (B) and third (D) measurement after treatment given treatment.

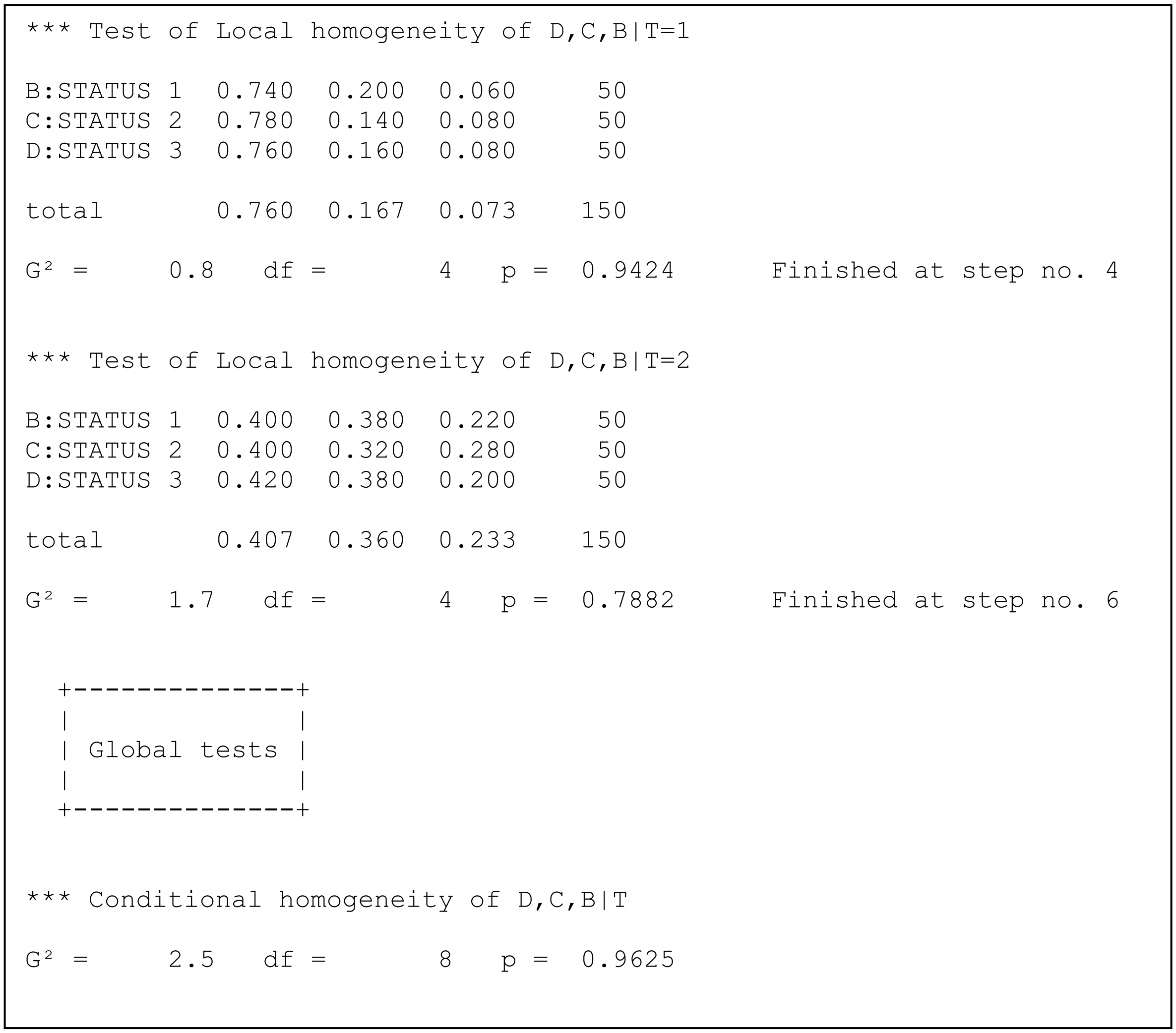
We return to the DCBT table discussed in the previous section for an illustration of the analysis of homogeneity of more than two repeated measurements.

There are three repeated measurements in the table and one additional covariate. The command, CMH DCB, will therefore result in an analysis of conditional homogeneity. We show only part of the results here. Some parts of the analysis by DIGRAM is still experimental and other parts require more information than it is possible to give in a short guided tour. Results relating to the basic test of conditional homogeneity appear as in Figure 29.

Comments on Figure 29:
The test statistics presented here are the same homogeneity deviances as in Figure 28. Several types of γ coefficients will eventually be implemented.

The marginal frequencies for the different treatments appear to be very different. Whether or not there is a significant difference between treatments with respect to side effects is however another matter. One can, of course, test marginal difference for each of the three measurements at a time, but the tests cannot be summarized into one simple global test statistic because of the correlation between measurements. It is however possible to fit models for the complete DCBT table assuming both conditional homogeneity and marginal independence of side effects and treatment. It is therefore also possible to test marginal independence of treatment and side effects given marginal homogeneity.

Tests like that is actually already being tested right now. Output from the CMH procedure automatically reports the results from these tests, and you are of course free to try them out (at your own responsibility of course).



```
*** Test of Local homogeneity of D,C,B|T=1

B:STATUS 1  0.740  0.200  0.060     50
C:STATUS 2  0.780  0.140  0.080     50
D:STATUS 3  0.760  0.160  0.080     50

total       0.760  0.167  0.073    150

G² =     0.8   df =      4   p =  0.9424      Finished at step no. 4


*** Test of Local homogeneity of D,C,B|T=2

B:STATUS 1  0.400  0.380  0.220     50
C:STATUS 2  0.400  0.320  0.280     50
D:STATUS 3  0.420  0.380  0.200     50

total       0.407  0.360  0.233    150

G² =     1.7   df =      4   p =  0.7882      Finished at step no. 6


  +--------------+
  |              |
  | Global tests |
  |              |
  +--------------+


*** Conditional homogeneity of D,C,B|T

G² =     2.5   df =      8   p =  0.9625
```

Figure 29. ***CMH DCB:*** *Tests of conditional homogeneity of three measurements of UKU side-effects given treatment.*

Analysis of multidimensional contingency tables 3: Fitting loglinear models.

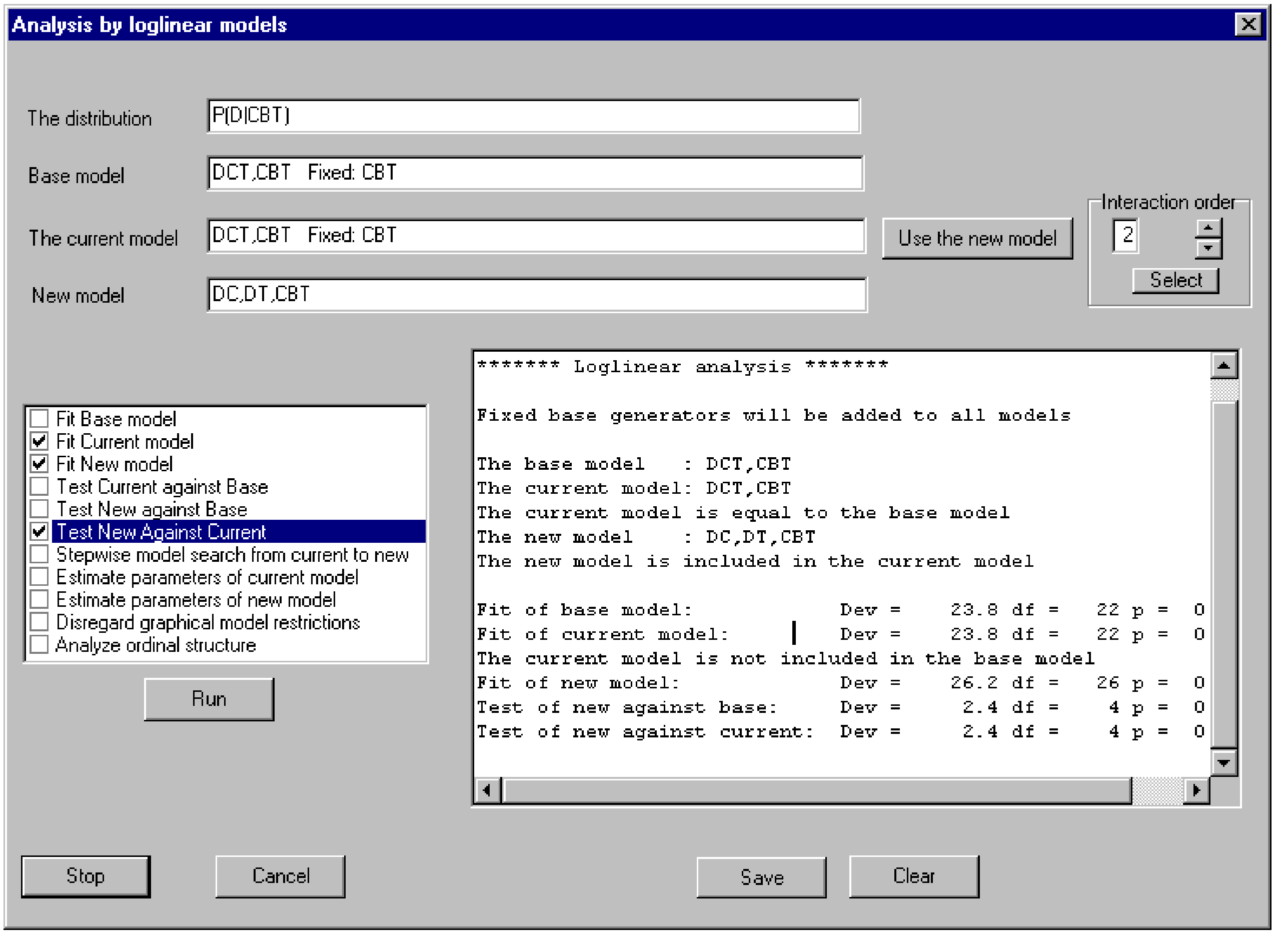
The current version of DIGRAM has some very limited facilities for fitting general hierarchical loglinear models. They will in time be expanded, but until that happens you are perhaps better advised to go elsewhere for this kind of analysis.

What you can do with this version is:

1) You can calculate the deviance of the current marginal base for the table.
2) You can calculate the deviance of any loglinear hierarchical model and test this model against both the base model and any other model.
3) For ordinal data you can compare observed γ coefficients with expected coefficient calculated for the currently fitted model.

Analysis by loglinear models is executed through the dialogue window shown in Figure 29. The command to activate this dialogue is

LOGLIN Activates the loglinear dialogue and transfers model information from the marginal loglinear model for the current table to the dialogue.



The distribution: P(D|CBT)

Base model: DCT,CBT Fixed: CBT

The current model: DCT,CBT Fixed: CBT

New model: DC,DT,CBT

Interaction order: 2

☐ Fit Base model
☑ Fit Current model
☑ Fit New model
☐ Test Current against Base
☐ Test New against Base
☑ Test New Against Current
☐ Stepwise model search from current to new
☐ Estimate parameters of current model
☐ Estimate parameters of new model
☐ Disregard graphical model restrictions
☐ Analyze ordinal structure

```
******* Loglinear analysis *******

Fixed base generators will be added to all models

The base model   : DCT,CBT
The current model: DCT,CBT
The current model is equal to the base model
The new model    : DC,DT,CBT
The new model is included in the current model

Fit of base model:             Dev =    23.8 df =   22 p =   0
Fit of current model:          Dev =    23.8 df =   22 p =   0
The current model is not included in the base model
Fit of new model:              Dev =    26.2 df =   26 p =   0
Test of new against base:      Dev =     2.4 df =    4 p =   0
Test of new against current:   Dev =     2.4 df =    4 p =   0
```

*Figure 29. **LOGLIN** activates the dialogue window**.** The output window of the loglinear dialogue shows the results of fitting the base model, the current model and a new model.*

You have three models to work with in the loglinear environment:

When the dialogue is activated DIGRAM transfers information on the marginal loglinear model for the current table to *the base model* of the dialogue. This model defines the connection between the loglinear analysis and the current graphical project model. It cannot be changed at any point of the analysis. The base model has certain fixed generators that cannot be analyzed as long as you want to keep the connection between the loglinear analysis and the graphical project model intact, because the graphical model is not parametric collapsible onto the current table with respect to the parameters associated with these generators.

During the analysis you look at *new models* and designate some of these as *the current model* that you – at a given point of time during the analysis – consider adequate. New models may always be evaluated against both the base model and the current model.

Defining, displaying and analyzing graphical models

Graphical models are defined, displayed and analyzed in the Graph window (Figure 9). Most of the commands controlling the definition and analysis of models may also be executed for the project model in the DIGRAM window.

The following types of interaction graphs can be accessed in the graph window[14]:

- The project model.
- Subgraphs of the project model
- The marginal model associated with the current table.
- IRT graphs of the loglinear graphical Rasch models defined in the GRM dialogue described below.
- Ad hoc graphs defined by yourself. Ad hoc graphs may and may not have any connection to data.

[14] Additional types of graphs, e.g. graphs for Markov Chain models or graphs for Repeated measurements, may eventually be implemented.

The interaction graph for the project model is the default.

Three types of commands are available in the graph window:

Commands redefining graphs:

ADD var1 var2

Adds an edge/arrow between var1 and var2 to the graph being shown in the graph window.

DELETE var1 var2

Delete the edge/arrow between var1 and var2 in the graph being shown in the graph window.

FIX var1 var2

Fixes the edge/arrow between var1 and var2 in the graph being shown in the graph window. Fixed edges between variables will are shown with thick lines in the graph and will not be removed by any of DIGRAM's model selection procedures.

PREVENT var1 var2

Prevents inclusion of an edge/arrow between var1 and var2 in the graph by DIGRAM's model selection procedures.

NEW <status>

Initiates the graph with connections between variables defined by the status in the following way:
Status 1 : all variables are unconnected.
Status 2: all variables are connected by unfixed edges.

XPLANATORY

Fixes edges between all variables in the final recursive block. These variables will be treated as purely explanatory variables during the analysis. It will be assumed that there is no interest in an analysis of the association between these variables.

SAVE

Saves the project model on the project's GRF file.

READ G Reads the project model from the project's GRF file.

Commands for analyses of graphs[15]:

SEPARATE pairs of variables Identifies the smallest subsets of variables separation two variables in the sense that all *indirect* paths between the variables has to go through at least on of the variables in the separating sets. Separating creates so-called separation hypotheses.

REDUCE pairs of variables Identifies the smallest component of the model with the connection between the variables in the interior of the component.

PATHS var1 var2 Identifies all paths with no shortcuts between two variables.

MODEL variables Determines the marginal model for the list of variables.

RELEVANCE var 1 var2 var3 var4 Examines the relevance of the relationship between var1 and var2 for the analysis of var3 and var4.

Output from these commands will appear in the output area in the GRAPH window, but may be saved in the output area of the DIGRAM window. The results from an analysis of relevance is shown and discussed below.

Output from the SEPARATE and DECOMPOSE commands may be accessed as hypotheses in the DIGRAM window, if the graph being analyzed was the interaction graph associated with the project model.

The purpose of the RELEVANCE command is to determine whether or not the status of the association between two variables influence the way another relationship ought to be

[15] The commands described in this subsection may be invoked from the DIGRAM window where they will always be understood as commands relating to the project model. The only exception from this rule is the MODEL command that has a different meaning in the DIGRAM window.

analyzed. Figure 30 shows that the analysis of the long term effect of treatment (T) on side effects (D) depends on whether or not it is assumed that side effects at the initial measurement after treatment (B) influences side effects at the final measurement. Parts of the model checking discussed below (CHECK DT) is grounded on analyses of relevance.

```
----------------------------------
Analysis of relevance of DB for DT
DB is currently excluded
----------------------------------

Separators for DT:

DB excluded: C
DB included: CB

The core of the DT-problem:

DB excluded: DCT
DB included: DCBT
```

Figure 30. ***RELEVANCE DB DT:*** *D and T are separated by C if DB is not in the model and separated by C and B if it is in the model. In this case the separators define the smallest irreducible components, the core, in which the relationship between D and T may be analyzed*

Commands controlling the appearance of graphs:

PLOT variables

HORIZONTAL variables

VERTICAL variables

The commands are always aimed at the graph being shown in the graph window. The following commands may be executed in the DIGRAM window, where they will always be taken as directed at the project graph.

Generating tables and hypotheses by analysis of graphs

Separation and reducibility defines hypotheses of conditional independence in marginal tables that are true if the variables are conditional independent in the complete project model. If the two variables are associated in the complete model they will also be related in the marginal model. The project model is however parametrically collapsible onto the marginal model with respect to the interaction parameters describing the association between the variables. Marginal tables and models defined by separation or reduction may therefore be of interest irrespective of whether the two variables are independent or not.

Separation and reducibility hypotheses for the project graph are saved for by the program and can therefore be accessed by the program.

Two commands will be of use here:

SHOW H Displays the current list of hypotheses defined by separation in or reduction of the project graph.

CHOOSE <no.> Creates the table in which the selected hypothesis may be tested. Tabulation proceeds exactly as after the TABULATE command.

Testing model based hypotheses

You do not have to create a table to test a hypothesis. The TEST command interprets two variables as parameters as a request for a test of conditional independence given the basic structure defined by the current project model:

TEST VAR1 VAR2 Tests conditional independence of Var1 and Var2 given the separators of these variables in the project model.

A test of long term treatment effect of UKU side effects for the model given in Figure 23 is shown below.

```
****  Summary of test results  ****

NSIM = 1000 tables generated for exact p-values

--------------------------------------------------------------------------------------
                         p-values                                   p-values
Hypothesis       X²  df asymp exact               Gamma asymp exact                nsim
--------------------------------------------------------------------------------------
 1:D&T|C        6.6   6 0.363 0.375 (0.337-0.415)  0.48 0.016 0.014 (0.007-0.027) 1000
--------------------------------------------------------------------------------------
```

Figure 31. ***TEST DT:*** *C separates D and T in the graph in Figure 23. The hypothesis to be tested is therefore* $D \perp\!\!\!\perp T \mid C$

Note, that you can only obtain global test results from the test command used in this way, and that the DCT table will not be available for further analysis after the test shown in Figure 31. No table will be created

Model selection 1: Initial screening

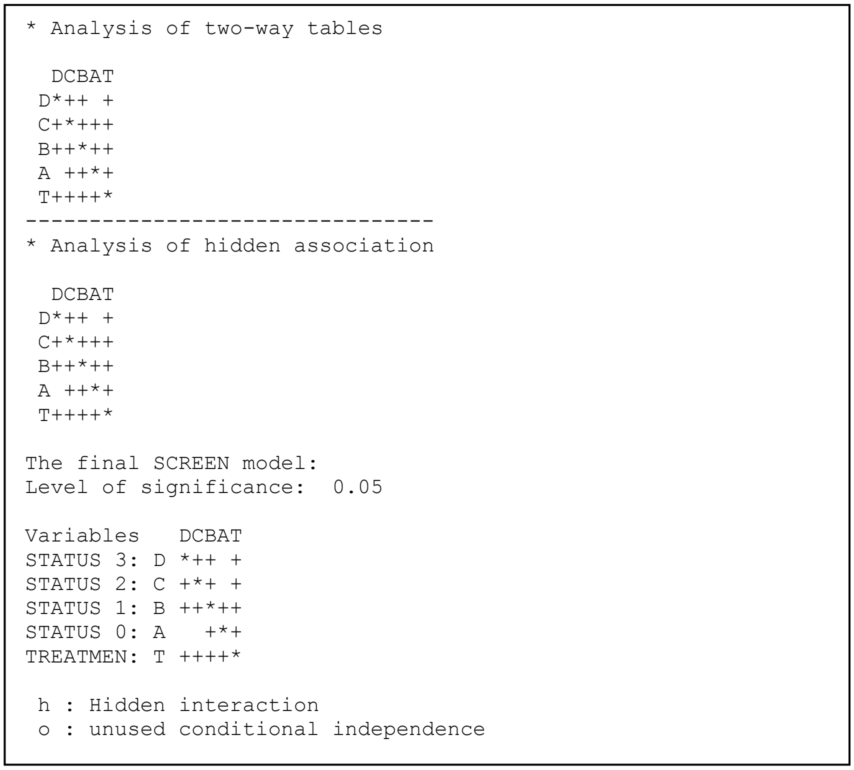
Model searching in DIGRAM is generally based on a two-step procedure. The purpose of the first step is to identify an initial model that may be used as a convenient starting point for the actual model search. DIGRAM refers to this initial step as screening. The basic example is the screening of data for an initial graphical project model, but the approach is used also for screening for an initial loglinear graphical Rasch model in connection with DIGRAM's item analysis discussed below.

The initial screening for a graphical model generates a graphical model after analysis of all two-way tables and some three-way tables with project variables.

To start the analysis enter

SCREEN	Generates an initial graphical model by analysis of two- and three-way tables.

The screening of the UKU data and the resulting model is shown in Figures 32 and 33.



```
* Analysis of two-way tables

  DCBAT
 D*++ +
 C+*+++
 B++*++
 A ++*+
 T++++*
--------------------------------
* Analysis of hidden association

  DCBAT
 D*++ +
 C+*+++
 B++*++
 A ++*+
 T++++*

The final SCREEN model:
Level of significance:  0.05

Variables   DCBAT
STATUS 3: D *++ +
STATUS 2: C +*+ +
STATUS 1: B ++*++
STATUS 0: A   +*+
TREATMEN: T ++++*

 h : Hidden interaction
 o : unused conditional independence
```

Figure 32. ***SCREEN:*** *Analysis of two- and three-way tables*

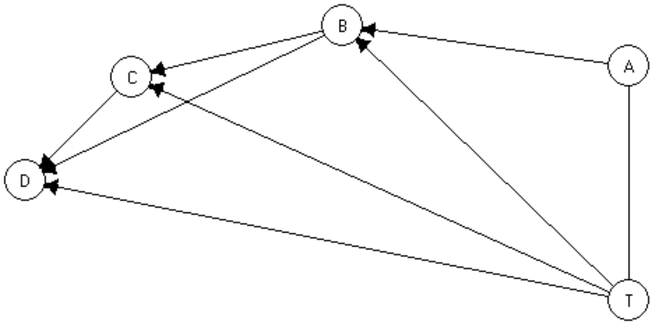


Figure 33. The initial screen model

Some remarks on DIGRAM's screening for an initial model:

Details of this procedure can be found in Kreiner (1986). What is of special importance here is to remember that screening does not add up to a proper statistical analysis of your data and that the results of screening never should be reported as such. The sole purpose of the screening is to generate a model that may save us some time looking for a plausible model using proper techniques for model search. For this reason output containing test statistics obtained during screening has been disabled in this version of DIGRAM. You only get the summary of the results shown in Figure 32 and the suggested model seen in Figure 33.

Some comments on what has happened in Figure 32 may nevertheless be in order.

Screening consists of three steps:

1) First an analysis of all two-way tables. The Results are summarized in the first adjacency matrix in Figure 32 where a '+' indicates that a significant evidence of marginally association has been found. It is seen that there was no significant association between A and D.
2) Second a check that insignificant marginal results do not hide significant conditional association. In this case the AD association was examined in all three-way tables containing A and D and one other variable. No hidden association was indicated.
3) Finally the association between variables that seemed to be associated according to the first two steps are reanalyzed in three-way tables containing the variables in question and. If variables are found to be conditionally independent in just one table, the '+' indicating association will be removed. In this case we see that not only A and D, but also C and D are conditionally independent[16].

[16] It is easy of course, but too time-consuming to create the three-way tables analyzed during the screening. In this way it is found that A and C seems to be conditionally given both B and T

Finally, interpreting the last adjacency matrix of Figure 32 as a adjacency matrix of a graphical model we arrive at the model shown in Figure 33. You should compare this model to the model in Figure 23. The screening model is somewhat more complicated than the model in Figure 23, which is to be expected, but it is actually quite close. The only difference is the association between D and B suggested by the screening model, but not included in the other model.

The closeness of the screening model to a final plausible model is not only a question that has been confirmed, but is a consequence of the somewhat primitive approach used during screening. Apart from random error, analyses of three-way tables will result in a model replicating the part of the structure of the true model, that are defined as strings or trees of either separate variables or larger cliques that are only connected by single variables, because the separation properties for these parts of the models define hypotheses of conditional independence in three-way tables. Analyses of two-way tables will in the same way identify completely disconnected components of the model. Screening will therefore at least get some of the structure right from the very beginning.

The default critical level for the statistical test executed during screening is 0.05. This choice was made deliberately even though it is obvious that a critical level this high in most cases will result in a fairly large number of type II errors. The screen model will therefore be too complex with more edges than really is needed.

If it turns out that the screen model is too complex to be practical you can use one of the following two commands to generate a less complex model:

SCREEN critical level The critical level must be given as an integer, A, with the critical level for the statistical tests calculated during screening will be equal to A/1000.

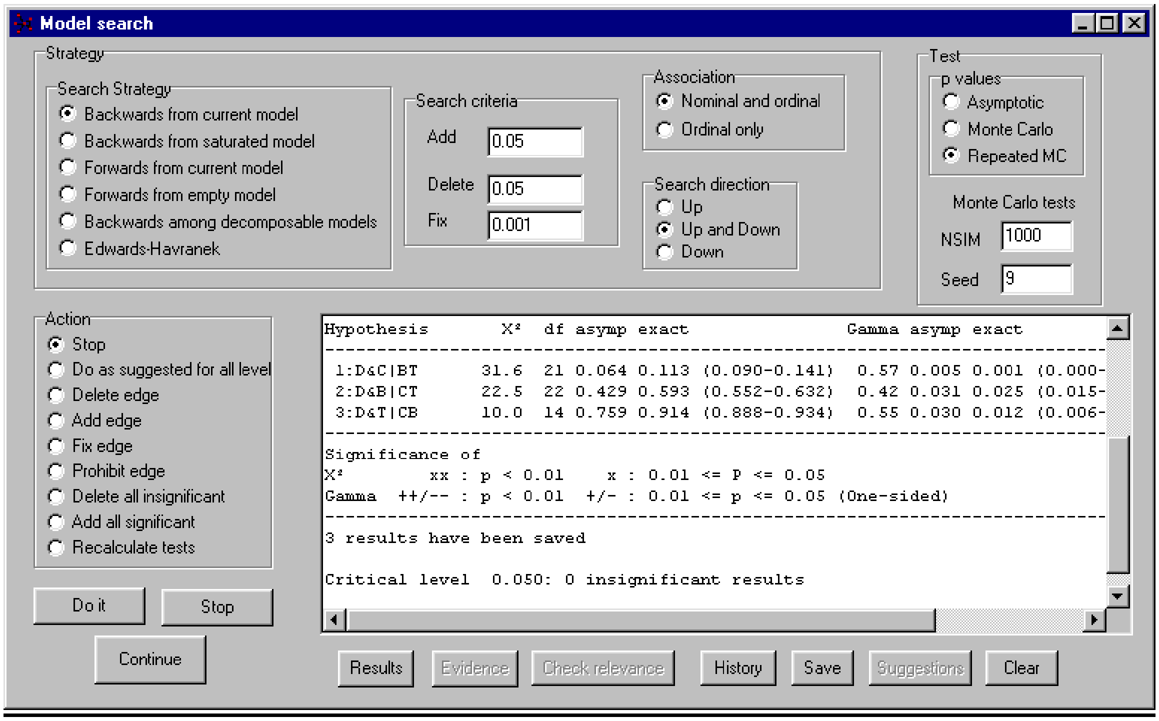
SCREEN B A Bonferroni corrected 5 % level will be used during screening.

Model selection 2: Stepwise model search

Taking the screen model (or any other model you might prefer) as the starting point we can proceed to the next and more serious problem of searching for models.

Model searching has been relegated to it's own dialogue shown in Figure 34. To activate this dialogue enter

MODELSEARCH variables Initiates a search for a model structure connecting the variables in the variable list to the remaining variables of the model.



Model search

Strategy

Search Strategy
- ● Backwards from current model
- ○ Backwards from saturated model
- ○ Forwards from current model
- ○ Forwards from empty model
- ○ Backwards among decomposable models
- ○ Edwards-Havranek

Search criteria
- Add 0.05
- Delete 0.05
- Fix 0.001

Association
- ● Nominal and ordinal
- ○ Ordinal only

Search direction
- ○ Up
- ● Up and Down
- ○ Down

Test

p values
- ○ Asymptotic
- ○ Monte Carlo
- ● Repeated MC

Monte Carlo tests
- NSIM 1000
- Seed 9

Action
- ● Stop
- ○ Do as suggested for all level
- ○ Delete edge
- ○ Add edge
- ○ Fix edge
- ○ Prohibit edge
- ○ Delete all insignificant
- ○ Add all significant
- ○ Recalculate tests

```
Hypothesis        X²  df asymp exact                     Gamma asymp exact
--------------------------------------------------------------------------------
 1:D&C|BT       31.6  21 0.064 0.113 (0.090-0.141)  0.57 0.005 0.001 (0.000-
 2:D&B|CT       22.5  22 0.429 0.593 (0.552-0.632)  0.42 0.031 0.025 (0.015-
 3:D&T|CB       10.0  14 0.759 0.914 (0.888-0.934)  0.55 0.030 0.012 (0.006-
--------------------------------------------------------------------------------
Significance of
X²         xx : p < 0.01    x : 0.01 <= P <= 0.05
Gamma  ++/-- : p < 0.01  +/- : 0.01 <= p <= 0.05 (One-sided)
--------------------------------------------------------------------------------
3 results have been saved

Critical level  0.050: 0 insignificant results
```

Figure 34. ***MODELSEARCH D:*** *Search for the structure connecting D to the rest of the UKU variables. The default strategy is backwards model search, implying tests of conditional independence of D and the variables connected to D in the current model.*

Comments on DIGRAM's model search:

First notice, that automatic model searching has been removed from this version of DIGRAM. The reason for this is that model searching of any kind is a problematic activity. We may not be able to avoid it, but there is no reason to make model selection more arbitrary than absolutely necessary. Instead of giving the program and some more or less arbitrary criteria the responsibility for finding an adequate model, we feel that the researcher should have full control, but also full responsibility over what goes on.

During model search you will either remove or add edges to the model. To help you decide what to do DIGRAM calculates different kinds of statistics the significance and suggests that the least significant edge should be removed or the most significant added. The p-value of the test statistics are however not the best criteria for deciding what to do. For ordinal variables where monotonous relationships are to be expected you should look at the size of the γ coefficients measuring the strength of association and use this instead of the size of the p-values to guide you. First of all you should be guided by substantive arguments rather than purely formal criteria. If several edges could be removed from the model, removing the one with least substantive support – the one that you would have the hardest time to explain if it was kept in the model – should be eliminated first.

There are two problems with informal strategies like that. The first is that the informal nature of the strategy prevents a formal testing of the strategy. It is hard to set up simulation experiments to test the qualities of substance matter driven model searching compared to other strategies. The second problem is that such a strategy will be prone to errors reflecting misconceptions and prejudices on behalf of the one controlling the analysis.

To avoid the second pitfall we suggest that you distinguish between primary and secondary problems. Primary problems are those that are especially important for you to solve in a correct way. Secondary problems on the other hand are of no particular interest (for the current study) apart from the fact that a correct solution of a secondary problem will increase the chances of a correct solution to the primary problems. In the UKU example the relationships between the different measurements of side effects constitute the primary problem while the associations among the repeated measurements of side effects is of secondary importance.

If you are able to do this you are well advised to leave edges relating to primary problems alone as long as possible. Let the model search take care of the secondary structure of the model first and don not remove or add primary edges until you feel that you can’t get any further with the secondary. Knowing the nature of the UKU measurements we would of course expect that the risk of side effects on one time depend on the presence of side effects at the previous measurement. We would also acknowledge the possibility that the risk of side effects at the current measurement would depend on the presence of earlier measurements, but here our expectations would be a bit less confident. Irrespective of the size of assorted p-values we would remove the edge between a measurement at one time and the very first measurements before we removed the edge between the current measurement and the one immediately before.

In the UKU example substance matter considerations outlines an almost complete strategy leaving us with no arbitrary choices among insignificant test results when we try to eliminate edges in the model. We cannot however expect things to be this simple in other applications. Decisions during model search will in practice often be prone to a certain degree of arbitrariness.

To help you in these cases you may examine the consequences of making specific choices for the evidence suggesting other choices. In DIGRAM this is referred to as an analysis of *relevance*. If the analysis suggest the two edges should be added to the model, the analysis of relevance examines the evidence suggesting that the second edge should be included conditionally given that the first one was included and visa versa.

After these comments we can describe what you must do to help DIGRAM find an adequate model. The agenda you have to consider at each step contains the following items:

Strategy

The two basic strategies are backwards and forwards model search, where backwards eliminates edges while forwards adds edges to the model. You can of course change back and forth between these strategies during models search. In most cases each step starts with the current model, but you may decide to go backwards from a saturated model or forwards from an empty model.

Backwards model search is default. If you want to start with a Forwards model search you can use the FORWARD command instead:

FORWARD variables	Does the same as “MODEL variables” except that the initial strategy is forwards instead of backwards

For completeness, the following command has also been included. It is however redundant:

BACKWARD variables	Does the same as “MODEL variables”.

The last two options on the strategy agenda have not been implemented yet.

Search criteria

DIGRAM uses critical levels for test statistics as search criteria. Remember that these criteria are only used for suggestions. DIGRAM will do nothing on it’s own based on criteria like those.

In addition to criteria for adding or deleting edges there is an additional criteria for fixing edges in the model if the evidence against conditional independence appear to be very strong during the search for a model. The effect of fixing an edge is that this edge will not be examined by the model search procedure preserving both consistence of conclusions and saving time spent on model searching.

Examination of either ordinal structure only or of both nominal and ordinal structure

If substance matter considerations tell you that relationships between variables should be monotonous you should at disregard the χ^2 tests during model search to avoid type I errors that will prolong the analysis and may lead to meaningless results. If you are concerned about overlooking something by not considering all the tests, you might consider starting looking only for monotonous associations and then reintroducing the χ^2 tests in the final steps.

Exact or asymptotic p-values

Exact p-values are generally to be preferred to asymptotic ones, but may be quite expensive in terms of the time you need to use looking for models. To reduce time you may to begin with consider exact tests based on a fairly small number of random tables or you may use the so-called repeated Monte Carlo test, which will stop, when it is (almost) certain, that the result will not be significant. Invoking the EXA command before you start the search for models means that the model search dialogue will assume that that exact tests should be used. You may of course change your mind at any time while you search for the model.

Repeated Monte Carlo tests are Monte Carlo tests that stops if the risk of obtaining a significant result at a 0.05 level becomes small (less than 0.001). You can select repeated Monte Carlo tests to save time instead of proper Monte Carlo tests during model search or before model search starts. The command is

REPEATED <nsim <seed>>	Selects repeated Monte Carlo tests. Nsim = 1000 and seed = 9 are the default Values.

Repeated Monte Carlo tests will also accept asymptotic p-values as evidence of insignificance if the asymptotic p-values are much larger than the critical level assigned for the test ($p > 0.20$)[17].

Repeated Monte Carlo tests are also used in connection with model checking as discussed below.

[17] This is justified because exact p-values typically are larger than asymptotic p-values (Kreiner, 1987).

Action

After each step of the model search procedure some action has to take place. Clicking on the “Do it” button executes the selected action, while “Continue” executes the action and continues the search for the model.

Clicking on the “Stop” button terminates the model search. If a new model has been found you will be asked whether or not you actually want to use it.

Dealing with output during model search.

Output during model search appears in the small output field in the model search dialog, Figure 34. The contents of this window will be cleared at the beginning of each step. If you want to save it in order to document what went on along the way to the final model, you therefore have to save the output on the DIGRAM output window.

Output produced during model search consists of test results in the same format as shown in Figure 31 plus a short summary and some suggestions based on the search criteria described above.

Seven buttons relating to the output a located below the output filed. These buttons and their function are described below.

Button	Comments
Results	Redisplays the test results obtained during the current step of the model search process.
Evidence	Shows the insignificant test results during backwards search and the insignificant test results during forward search. This button is disabled when there are no significant test results.
Relevance	Initiates an analysis of relevance if test results suggest that more than one edge or arrow should change status. This button is disabled if test results suggest that no more than one edge should change status.
History	Summarizes what has happened during model search. The same kind of summary will appear automatically in the DIGRAM output window at the end of the model search.
Save	Saves the current output in the DIGRAM output window.
Suggestions	Repeats the suggestions based on the current test results.
Clear	Clears the output window.

Model checking

We may distinguish between two different approaches seem to be considered for testing whether or not a specific model gives an adequate description of data:

a) Global test or fit statistics evaluating the overall fit between model and data. calculating deviances as it for instance seem to be the established approach for generalized linear models is one example of model checking by global statistics. Fit statistics based on information theory (e.g. AIC or BIC) are other examples of global model checking. Global model checking is sometimes enhanced by careful scrutiny of residuals identifying cases with unusual (according to the model) outcomes on the variables of the model.

b) Local evaluation of hypotheses derived as necessary – but not necessarily sufficient – consequences of the model.

It is our experience that global checks of loglinear and graphical models for high-dimensional contingency tables often are close to being both untrustworthy and useless. It is not unusual that p-values for deviances comparing loglinear models to the saturated model very often are to close to 1 to be comfortable. This problem reflects two problems. First, that standard tests are tests with almost no power accepting models that are much too simple, and second, that there are problems with the approximation of the distribution of the deviance by the asymptotic χ^2 distribution. Kreiner (1987) compares asymptotic p-values to Monte Carlo estimates of exact p-values for a range of very simple models showing that asymptotic p-values definitely is not to be relied on.

Another problem with the global approach to model testing is that the global tests are destructive tests in the sense that they may tell that there is something wrong with the model without providing any information on exactly what the problem is. Unless one wants to abandon the analysis if the model is rejected, one has to start looking for specific types of departures from the model. The most direct way to do this is to examine some specific properties of the model in question, which is exactly what happens during local model checking.

Finally, global criteria for fit of loglinear models only treats variables as nominal variables even though the vast majority of variables comprising contingency tables are however either dichotomous or ordinal. Violation of model assumptions because of monotonous relationships between variables are therefore in many cases conceivable, but standard approaches will have a hard time identifying this kind of problems.

For this reason model checking in DIGRAM is strictly local. Instead of bothering with the unreliable global results DIGRAM proceeds directly to tests of conditional independence implied by the model, because conditional independence is the formative properties of these models. This is done in two different ways:

CHECK Results in tests of all separation hypotheses implied by the model.

CHECK var1 var2 Tests separation hypotheses for all variables of Relevance for the analysis of the relationship between variables var1 and var2.

We refer to the first approach as a *local model check* and to the second as a *relevance check.*

Figure 35 below illustrates the check of the model in Figure 23.

```
Check of conditional independence assumptions

3 Hypotheses:

HYPOTHESIS  1:  D & B  |  C T
HYPOTHESIS  2:  D & A  |  C T
HYPOTHESIS  3:  C & A  |  B T

****  Summary of test results  ****

NSIM = 1000 tables generated for exact p-values

--------------------------------------------------------------------------------------
                         p-values                        p-values
Hypothesis       X²  df asymp exact               Gamma asymp exact               nsim
--------------------------------------------------------------------------------------
 1:D&B|CT      22.5  22 0.429 0.593 (0.552-0.632)  0.42 0.031 0.025 (0.015-0.041) 1000 +
 2:D&A|CT      18.4  14 0.188 0.172 (0.124-0.234)  0.30 0.124 0.110 (0.072-0.165)  308
 3:C&A|BT      16.1  15 0.374                     -0.07 0.397
--------------------------------------------------------------------------------------
Significance of
X²        xx : p < 0.01    x : 0.01 <= P <= 0.05
Gamma  ++/-- : p < 0.01  +/- : 0.01 <= p <= 0.05 (One-sided)
--------------------------------------------------------------------------------------
```

Figure 35. ***CHECK:*** *Tests of all separation hypotheses implied by the model in Figure 23. Repeated Monte Carlo tests are used shortening the time used to generate random tables for the second hypothesis. Monte Carlo tests were abandoned for the third hypothesis because the asymptotic p-values were clearly insignificant (p > 0.20).*

Figure 35 discloses some evidence against the model in Figure 23. We note however, that p-values for the γ coefficients are one-sided, whereas two-sided p-values in most cases would have been appropriate for model generated hypotheses. Also, the fact that three hypotheses are tested weighs against relying too strongly on weakly significant test results. That γ coefficients suggesting strong positive association between D and B speaks on the other hand for taking the evidence seriously.

All in all, the results of model checking in Figure 35 cannot be said to be conclusive[18].

[18] We notice for the record that the deviance for the DCT,CBAT model for the conditional distribution of D given C,B,A and T is equal to 49.0. Due to zeros in the sufficient margins of this model it is necessary to reduce the degrees of freedom from 96 to 46 giving a p-value of 0.35. The deviance of the CBT,BAT

```
Relevance check for D&T


3 Hypotheses:

HYPOTHESIS  1:  D & B  |  C T
HYPOTHESIS  2:  D & A  |  C T
HYPOTHESIS  3:  D & C  |  T

The first 2 hypotheses must be accepted


****  Summary of test results  ****

NSIM = 1000 tables generated for exact p-values

---------------------------------------------------------------------------------------
                        p-values                                   p-values
Hypothesis       X²  df asymp exact              Gamma asymp exact               nsim
---------------------------------------------------------------------------------------
 1:D&B|CT      22.5  22 0.429 0.593 (0.552-0.632)  0.42 0.031 0.025 (0.015-0.041) 1000    +
 2:D&A|CT      18.4  14 0.188 0.172 (0.124-0.234)  0.30 0.124 0.110 (0.072-0.165)  308
 3:D&C|T       22.0   8 0.005 0.013 (0.006-0.026)  0.59 0.000 0.000 (0.000-0.007) 1000 x ++
---------------------------------------------------------------------------------------
Significance of
X²        xx : p < 0.01    x : 0.01 <= P <= 0.05
Gamma  ++/-- : p < 0.01  +/- : 0.01 <= p <= 0.05 (One-sided)
---------------------------------------------------------------------------------------
```

Figure 36. ***CHECK DT:*** *Three relationships are important for the way the association between variables D and T has to be analyzed.*

Comments on Figure 36:

D and B, and D and A, have to be conditionally independent for the collapsibility properties relating to D and T has to apply. If one of these assumptions is incorrect, the estimate of the association between D and T will be confounded. The model also assumes that D and C are related. If this is not the case collapsibility onto a smaller marginal then the one implied by the model will be possible. Estimation of the DT-relationship may not be confounded, but the standard error will be inflated.

Some comments on the relationship between model searching and model checking:

model for the conditional distribution of C given B,A and T is equal to 16.0. Degrees of freedom has to be reduced from 24 to 15 resulting in a p-value equal to 0.38.

It should be obvious to anyone that the demarcation line between model checking and model search is not very clear. In fact the procedure for local model checking is nothing less than a forward step in the strategy for model search implemented in DIGRAM. This is no coincidence of course, but it emphasizes that strict model checking is impossible.

The circularity problems that this creates may be seen from two different angles. From one viewpoint we may argue that model searching proceed along the following strategy:

Initialization:

Select an initial model. The model selected at any given point of time is referred to as the current model.

Step:

Check the adequacy of the current model. This check has to be fairly comprehensive. If the model is adequate we stop. Otherwise we

- generate a set of candidate models,
- perform a less comprehensive check each of these models,
- select on of these models as the current model,
- repeat the step.

From the point of view of the model checking we start with the ideal situation where we – or the researcher responsible for designing the study and collecting the data – are able to construct a plausible model for the data based strictly on subject matter considerations:

Initialization:

Construct a *plausible* model.

Model check:

Examine the adequacy of the model. Proceed to the final part of the analysis if the model fits the data.

Model search:

Search for a better model. Return to check the model when you have found a new model.

The only difference between the two approaches is the nature of the initial model. Model searching simply starts with a convenient model while model checking and improvement starts with a model that we actually expected to believe in. Apart from the rare cases where we hit the mark the first time, the end result is a model partly based on speculation and partly on what is found in data.

Describing relationships

We assume that one of the purposes of doing an analysis by graphical models is to find out how to control for confounding and effect modification of a limited number of specific relationships. Unless one want to condition with all remaining variables one has to determine how the complete model collapses onto specific relationships. This is done by identifying all minimal sets of separators of the pair of variables in question leading to marginal tables where the association parameters of interest are the same as in the model for the complete table. The marginal models for such tables are loglinear where the question of effect modification is a question of whether or not the two variables of interest are part of the same higher order interactions.

To obtain all available information on a specific relationship enter

DESCRIBE Var1 Var2 Analyses the relationship between Var1 and Var2

Information obtained by this command includes

- The marginal two-way table (Figure 37).
- Test of all separation hypotheses including
  - Information on the loglinear structure of the marginal tables in which separation hypotheses may be tested. This information includes information

on potential effect modificators of the relationship between Var1 and Var2 (Figure 38)

- Local test results (Figure 39).

- Analysis of homogeneity of local γ coefficients (Figure 40).

- Fit of a loglinear model that assumes that there are no higher order interaction involving Var1 and Var2 (partial relationship) (Figure 41).

- Standardized parameters of the two-way interactions between Var1 and Var2 assuming no higher order interaction (Figure 42).

- Fit of a loglinear model assuming conditional independence between Var1 and Var2 and calculation of the expected marginal γ coefficient under the assumption of conditional independence (Figure 40).

- A summary in epidemiological terms concerning confounders and effect modificators of the Var1 – Var2 relationship (Figure 44).

```
Table 1. The DT distribution.

     +TREATMEN
     | | D:STATUS 3         |
     T |  None  Mild mod+s | TOTAL |
-------+-------------------+-------+
 Drug1 |    38     8     4 |    50 |
   row%|  76.0  16.0   8.0 | 100.0 |
 Drug2 |    21    19    10 |    50 |
   row%|  42.0  38.0  20.0 | 100.0 |  X² =  12.0
-----------------------------------+  df =    2
 TOTAL |    59    27    14 |   100 |   p = 0.003
   row%|  59.0  27.0  14.0 | 100.0 | Gam =  0.57
-----------------------------------+   p = 0.000
```

Figure 37. ***DESCRIBE DT:*** *The marginal relationship between D and T. Side effect seems to be more serious for Drug 2.*

```
Separation hypothesis:

HYPOTHESIS  1:  D & T  |  C

  +----------------------+
  |                      |
  | Marginal model: D|CT |
  |                      |
  +----------------------+

The marginal model is graphical

Cliques of the marginal graph: DCT
Fixed interactions           : CT
Collapsibility:                Strong.

Estimable parameters         : D,DC,DT,DCT


The DT interaction may be modified by C
```

Figure 38. ***DESCRIBE DT:*** *D and T are separated by C. The Marginal DCT model is saturated. The relationship between D and T may be modified by C. Finally the (fixed) relationship between C and T cannot be analyzed here.*

```
****  Summary of results  ****

NSIM = 1000 tables generated for exact p-values

---------------------------------------------------------------------------------------
                         p-values             p-values
Hypothesis       X²  df asymp exact 99% conf.int. Gamma asymp exact nsim 99% conf.int.
---------------------------------------------------------------------------------------
 1:D&T|C        6.6   6 0.363 0.375 0.337 - 0.415  0.48 0.016 0.014 0.007 - 0.027 1000
---------------------------------------------------------------------------------------
-----------------------------------------------------------
** Local testresults for strata defined by STATUS 2 (C) **
                          p-values              p-values
 C: STATUS 2   X²    df asympt  exact  Gamma asympt  exact
-----------------------------------------------------------
 1:    None   6.03    2 0.0491 0.0390   0.59 0.0169 0.0110
 2:    Mild   0.10    2 0.9514 1.0000   0.06 0.4416 0.3450
 3:mod+stro   0.44    2 0.8040 1.0000   0.24 0.2905 0.2010
-----------------------------------------------------------
```

Figure 39. ***DESCRIBE DT:*** *Local tests of conditional independence of D and T*

What you see in Figures 38 and 39 is of course the same results that you could get out of tabulating and testing as discussed at the very beginning of the guided tour. We assume however that the DESCRIBE command is to be used at the end of the analysis, where the model is ready and where some conclusions concerning specific relationships has to be formulated. Figures 38 and 39 tell us that the marginal association between D and T is a somewhat confounded expression of the true conditional relationship between the two variables. The marginal and partial γ coefficients suggest however that the degree of confounding is minor and perhaps inconsequential.

The local γ coefficients are somewhat different form each other suggesting that the long term direct effect of treatment on side effects are modified by side effects at the previous measurement. The effect of treatment is strong and significant if there was no side effects at the previous measurement, but weaker and insignificant if side effects were present.

The two different conclusions – almost no confounding or effect modification – are very different and not compatible. The analysis is therefore not over until a choice has been made between these conclusions. The rest of the description of relationships attempts to solve this problem.

Effect modification is best represented as higher order interaction in loglinear models: Effects are modified if higher order interactions are present and not modified if there is no higher order interaction. From this point of view, the question of whether or effect are modified should be a simple one, requiring nothing but standard tests of hypotheses of vanishing higher order interaction. We want however to be a little more careful here. Standard techniques for analysis of multidimensional tables by loglinear models disregard the ordinal nature of variables and the fact that relationships often are monotonous. The power of tests for higher order interactions is therefore less than impressive.

Instead we suggest an analysis focusing first on the local γ coefficients and only second on the loglinear structure. The first question we want to ask therefore is whether or not the γ coefficients in Figure 39 provides sufficient evidence to conclude that the effect of treatment is stronger when side effects were not present at the previous measurement than when side effects were present. This question is answered in Figure 40.

Figure 40 first present an alternative partial γ coefficient calculated as a weighted sum of the local γ coefficients[19].

The next part of Figure 40 shows the local γ coefficient with corresponding standard errors. Below this table a χ^2 test of homogeneity of γ coefficient is reported. The differences between the local γ coefficients are after all not sufficient evidence against homogeneity of the local γ coefficients.

Finally results from a stepwise multiple comparison analysis are reported. Each step compares adjacent γ coefficients and collapses the least significantly different coefficients until either all coefficients have been collapsed or the remaining coefficients are significantly different at a 1 % level.

In the first step here γ coefficients for situation where side effects were present at the previous measurements are collapsed and replaced with a weighted mean of 0.15. In the final step this value is compared to the 0.59 values observed if no side effects were present at the earlier measurement. The difference is not significant. The conclusion therefore remains the same: Local γ coefficients seem to be homogenous.

The analysis of homogeneity of the γ coefficients in Figure 40 suggests that C does not modify the effect of T on D. We have to be a little careful here. The γ coefficients are non-parametric rank correlation coefficients. Unless both variables are dichotomous

[19] If the hypothesis of homogenous γ coefficients is correct the weighted coefficient is actually more precise than the ordinary partial coefficient.

results concerning homogeneity of coefficients cannot automatically be interpreted as evidence of no higher order interactions in the loglinear DCT model.

In addition to looking at γ coefficients we therefore also have to examine how well a model with no higher order interactions fits the data. This is done in Figure 41. The deviance also provide no arguments against partial association ($p = 0.65$).

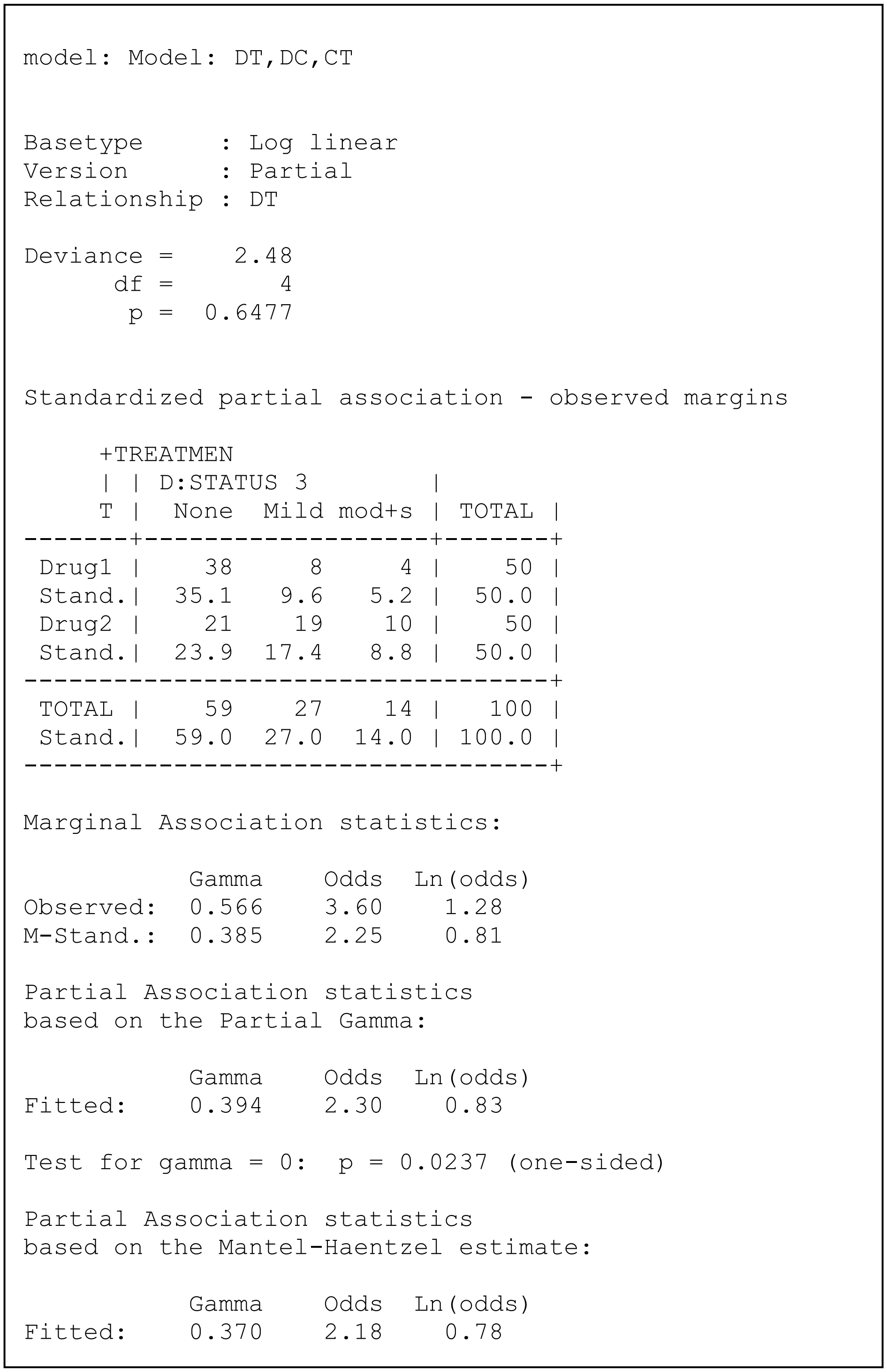
The question of how best to present the parameters of loglinear models has never had a satisfactory solution. The table in Figure 41 is perhaps an unconventional solution to this problem. Instead of presenting parameters summarizing to zero across all levels of variables or parameters where parameters for certain reference categories have been set to zero, we present *standardized*[20] parameters based on the assumption that it will be of interest to compare the conditional association between variables to the marginal in order to evaluate the degree of confounding implied by the model. The table shows both the observed marginal association between the two variables and a raked table with the same row and column margins as the observed table but with the same degree of association – the same loglinear parameters - as found in the fitted conditional association between D and T.

Fitting a table of parameters to the same margins of the observed table has the implication that statistics calculated on the marginal table can also be calculated for the fitted table. Figure 41 thus presents and compares γ-coefficient for both the marginal and the standardized partial association. The standardized partial γ coefficient is much smaller than the marginal, suggesting a high degree of confounding. The hypothesis of conditional independence may also be tested using the standardized partial coefficient as a test statistic. The evidence against conditional independence is significant. Finally Figure 41 presents results based on a Mantel-Haenszel procedure rather than the partial γ coefficients. These results will be discussed in a later section on Gamma coefficients.

[20] We refer to Agresti (1990) for a discussion of standardized and raked tables.

```
+------------------------------------------------+
|                                                |
| Analysis of homogeneity of Gamma coefficients |
|                                                |
+------------------------------------------------+


Gamma =  0.4813   s.d. =  0.1689

Possible higher order interaction:

  C: STATUS 2

+----------------------------------+
|                                  |
| Gamma coefficients in C-strata |
|                                  |
+----------------------------------+

Least square estimate:  Gamma =  0.4458 s.d.  =  0.1650

C: STATUS 2  Gamma variance     sd   weight  residual
-----------------------------------------------------
 1:           0.59   0.0402   0.2005  0.677   1.253
 2:           0.06   0.1604   0.4005  0.170  -1.060
 3:           0.24   0.1778   0.4216  0.153  -0.522
-----------------------------------------------------

Test for partial association: X² =    1.7 df = 2 p =  0.434


Analysis of collapsibility across 3 ordinal categories

Collapse of                                mean  sd    p
-----------------------------------------------------------
1&2:                                       0.48 0.179 0.237
2&3:                                       0.15 0.290 0.751

*** Collapsed: 2 & 3 ***

1&23:                                      0.45 0.165 0.210

*** Collapsed: 1 & 23 ***

All groups have been collapsed
```

Figure 40. ***DESCRIBE DT:*** *Analysis of homogeneity of local γ coefficients*



```
model: Model: DT,DC,CT


Basetype     : Log linear
Version      : Partial
Relationship : DT

Deviance =    2.48
      df =       4
       p =  0.6477


Standardized partial association - observed margins

     +TREATMEN
     | | D:STATUS 3        |
     T |  None  Mild mod+s | TOTAL |
-------+-------------------+-------+
 Drug1 |    38     8     4 |    50 |
 Stand.|  35.1   9.6   5.2 |  50.0 |
 Drug2 |    21    19    10 |    50 |
 Stand.|  23.9  17.4   8.8 |  50.0 |
-----------------------------------+
 TOTAL |    59    27    14 |   100 |
 Stand.|  59.0  27.0  14.0 | 100.0 |
-----------------------------------+

Marginal Association statistics:

           Gamma    Odds  Ln(odds)
Observed:  0.566    3.60    1.28
M-Stand.:  0.385    2.25    0.81

Partial Association statistics
based on the Partial Gamma:

           Gamma    Odds  Ln(odds)
Fitted:    0.394    2.30    0.83

Test for gamma = 0:  p = 0.0237 (one-sided)

Partial Association statistics
based on the Mantel-Haentzel estimate:

           Gamma    Odds  Ln(odds)
Fitted:    0.370    2.18    0.78
```

Figure 41. ***DESCRIBE DT:*** *Analysis of partial association*

```
model: Model: DC,CT


Basetype     : Log linear
Version      : Conditional independ
Relationship : DT

Deviance =     6.31
      df =        6
       p =  0.3895


Fitted marginal assuming conditional
independence

     +TREATMEN
     | | D:STATUS 3         |
     T |  None  Mild mod+s | TOTAL |
-------+-------------------+-------+
 Drug1 |    38     8     4 |    50 |
 Fitted|  34.0  10.9   5.1 |  50.0 |
 Drug2 |    21    19    10 |    50 |
 Fitted|  25.0  16.1   8.9 |  50.0 |
-----------------------------------+
 TOTAL |    59    27    14 |   100 |
 Fitted|  59.0  27.0  14.0 | 100.0 |
-----------------------------------+

Row frequencies

     +TREATMEN
     | | D:STATUS 3         |
     T |  None  Mild mod+s | TOTAL |
-------+-------------------+-------+
 Drug1 |    38     8     4 |    50 |
   row%|  76.0  16.0   8.0 | 100.0 |
 Fitted|  68.1  21.7  10.2 | 100.0 |
 Drug2 |    21    19    10 |    50 |
   row%|  42.0  38.0  20.0 | 100.0 |
 Fitted|  49.9  32.3  17.8 | 100.0 |
-----------------------------------+
 TOTAL |    59    27    14 |   100 |
   row%|  59.0  27.0  14.0 | 100.0 |
 Fitted|  59.0  27.0  14.0 | 100.0 |
-----------------------------------+

Marginal Association statistics:

           Gamma  s.e.    Odds  Ln(Odds)
Observed:  0.566 0.135    3.60    1.28
Fitted:    0.327 0.166    1.97    0.68

Test for observed=fitted :  p =0.0754
```

Figure 42. ***DESCRIBE DT:*** *Analysis of indirect effect*

Figure 42 presents results from fitting of a loglinear model, where it is assumed that D and T are conditionally independent. Apart from demonstrating the lack of power of the deviance, which is not able to detect the association between D and T, the main purpose of showing the results here is to give an idea about the size of the indirect effect of T on D. The two tables in Figure 42 compares the observed marginal distribution of D and T to the fitted marginal distribution under the assumption of no direct effect. The tables show both absolute and relative marginal frequencies and marginal γ coefficients for both the observed and fitted tables. The fitted γ value suggests that there is considerably indirect effect of about the same order of size as the estimate of the direct effect presented in Figure 41. Finally a test comparing observed and fitted γ coefficients are not significant. The test is however two-sided where a one-sided test assuming that the observed γ had to be larger than the fitted γ for the indirect effect.

Finally Figure 43 summarizes the results in terms of confounding and effect modification. C was a potential effect modificator, but no evidence turned up that this was the case. The partial γ was a little smaller than the marginal γ, but estimates of γ coefficients fitting a model with no higher order interactions involving D and T suggested a somewhat smaller value.

```
Summary of analysis of conditional relationship between
STATUS 3 and TREATMEN

 C:STATUS 2     Potential modificator - no evidence


Summary statistics

Marginal Gamma (all cases)         =  0.57   n =    100
Marginal Gamma (missing excluded) =  0.57   n =    100
Partial Gamma                      =  0.48  df =      6
```

Figure 43. ***DESCRIBE DT:*** *Summary of results*

Analysis by Markov chains

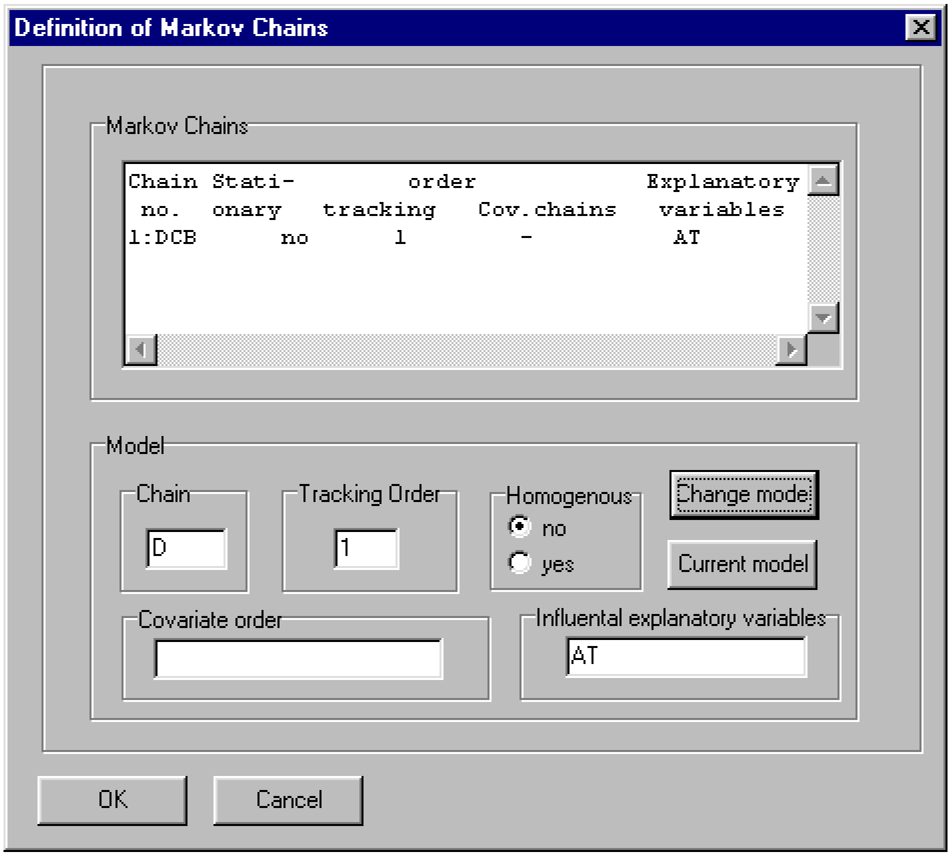
We are currently working on implementation of methods for analysis of multivariate Markov Chains. At the moment three commands are available. The first is used both to set up and modify Markov Chain models, while the last two are used to analyze the transition probabilities of the model.

MARKOV variables	Initiates the model in terms of repeated response variables and explanatory covariates.
MARKOV	Lets you modify an existing model.
MTABLES variable	Creates a table of transitions and the hypotheses needed to test that the model can be simplified. This table can be treated in exactly the same way as the other tables in DIGRAM.
MTEST variable	Tests all hypotheses of conditional independence implying that the model may be simplified.

The UKU example discussed in this notes is an obvious candidate for a Markov Chain model. We ignore the presence of side effects prior to treatment and regard treatment as an explanatory variable.

To initiate the Markov Chain model we write "MARKOV DCB". DIGRAM checks that the three variables are posited in three different recursive levels have the same number of categories and the same scale type (nominal or ordinal). If these requirements were fulfilled, all other variables in the same recursive blocks as D, C and B would have been ignored[21] while all variable from variables in earlier blocks will be treated as explanatory variables. The Markov Chain dialog, Figure 44, illustrates this set up.

[21] This of course is particular simple in this example as there are only one variable in each block. If we had three other variables in the blocks, they could either be included in the Markov structure or ignored. Assume that the project has to different repeated measurements in each block, D,G <- C,F <- B,E <-A,T, then MARKOV DGCFBE would define a model with two chains, the DCB chain as before and the GFE chain, which we might treat as a chain of time dependent covariates for the DCB chain. MARKOV DCB however defines the same model as in the UKU example.



```
Chain Stati-        order                  Explanatory
 no.  onary   tracking   Cov.chains    variables
1:DCB     no      1          -          AT
```

Chain: D

Tracking Order: 1

Homogenous: ● no ○ yes

Covariate order:

Influental explanatory variables: AT

Figure 44. ***MARKOV DCB:*** *A Markov chain model with A and T as explanatory variables has been constructed. The tracking order has been set to 1 here.*

Comments on Figure 44:

To modify the model you must select a chain of repeated measurements by entering one of variables in the chain field. Clicking the "Current model" button places the following information on the selected chain into the fields of the dialogue.

- *Tracking order*: The number of prior measurements on which the current measurement is assumed to depend.
- *Covariate order*: The number of current and prior measurements of time dependent covariates (if any) that the current measurement is assumed to depend on.
- *Homogeneity*: The chain is homogenous if transition probabilities do not depend on time.

- *Influental explanatory variables*: The explanatory variables on which the current chain is assumed to depend.

To modify the model you must first change the relevant information and click on the "Change model" button. In the current example tracking order was initially set to two (the maximum value with three measurement), but changed to one.

To proceed with the analysis you must leave the Markov Chain Dialog and then enter either MTAB or MTEST[22]. Figure 45 shows the result of TEST after MTAB.

If the table of transitions is more than 8-dimensional you have to use MTEST D instead of MTAB D. The results will of using MTEST D here is of course the same results as in Figure 45, but the table will not be available for further analyses.

[22] If the model is a model for more than one parallel chains the call of MTAB or MTEST must include a variable from the chain you want to analyze.

```
Markov chain table for chain no. 1 will be created
Trackorder = 1
Number of covariate chains = 0
Explanatory variables: AT
5 variables have been selected: DCAT#
---------------------------------

The Markov chain table:

D - STATUS 3 Dim = 3
C - STATUS 2 Dim = 3
A - STATUS 0 Dim = 3
T - TREATMEN Dim = 2
# -     Time Dim = 2

4 Hypotheses:

HYPOTHESIS  1:  D & C  |  A T #
HYPOTHESIS  2:  D & A  |  C T #
HYPOTHESIS  3:  D & T  |  C A #
HYPOTHESIS  4:  D & #  |  C A T


****  Summary of test results  ****

NSIM = 1000 tables generated for exact p-values

--------------------------------------------------------------------------------------
                        p-values                              p-values
Hypothesis       X²  df asymp exact               Gamma asymp exact                nsim
--------------------------------------------------------------------------------------
 1:D&C|AT#     76.4  29 0.000 0.000 (0.000-0.007)  0.66 0.000 0.000 (0.000-0.007) 1000
 2:D&A|CT#     34.5  29 0.220 0.327 (0.290-0.366)  0.15 0.217 0.249 (0.216-0.286) 1000
 3:D&T|CA#     23.9  19 0.200 0.290 (0.255-0.328)  0.40 0.015 0.009 (0.004-0.021) 1000
 4:D&#|CAT     15.2  19 0.710 0.885 (0.856-0.908)  0.08 0.330 0.333 (0.296-0.372) 1000
--------------------------------------------------------------------------------------
```

Figure 45: ***MTAB D and TEST*[23]*:*** *MTAB creates a table of transitions where the current state is referred to by the label of the latest variable. '#' is used as the label of time, if the chain is not supposed to be homogenous.*

[23] Monte Carlo estimates of exact p-values are used, although the conditional distribution given marginas are not exactly hypergeometrical.

Item analysis

DIGRAM fits a wide range of graphical and loglinear Rasch models for dichotomous and polytomous (ordinal) items. The details of DIGRAM's item analysis will be described elsewhere[24].

You define items scores and exogenous variables for item analysis by the following three commands:

ITEMS variables	Selects items and calculates a summary score
EXOGENOUS variables	Selects exogenous variables
CUT cut-points	Defines score intervals

Once things have been set up for item analysis, there are several commands that you may use. We only use one here

GRM	Activates a dialog where you may define and work With graphical and loglinear Rasch models

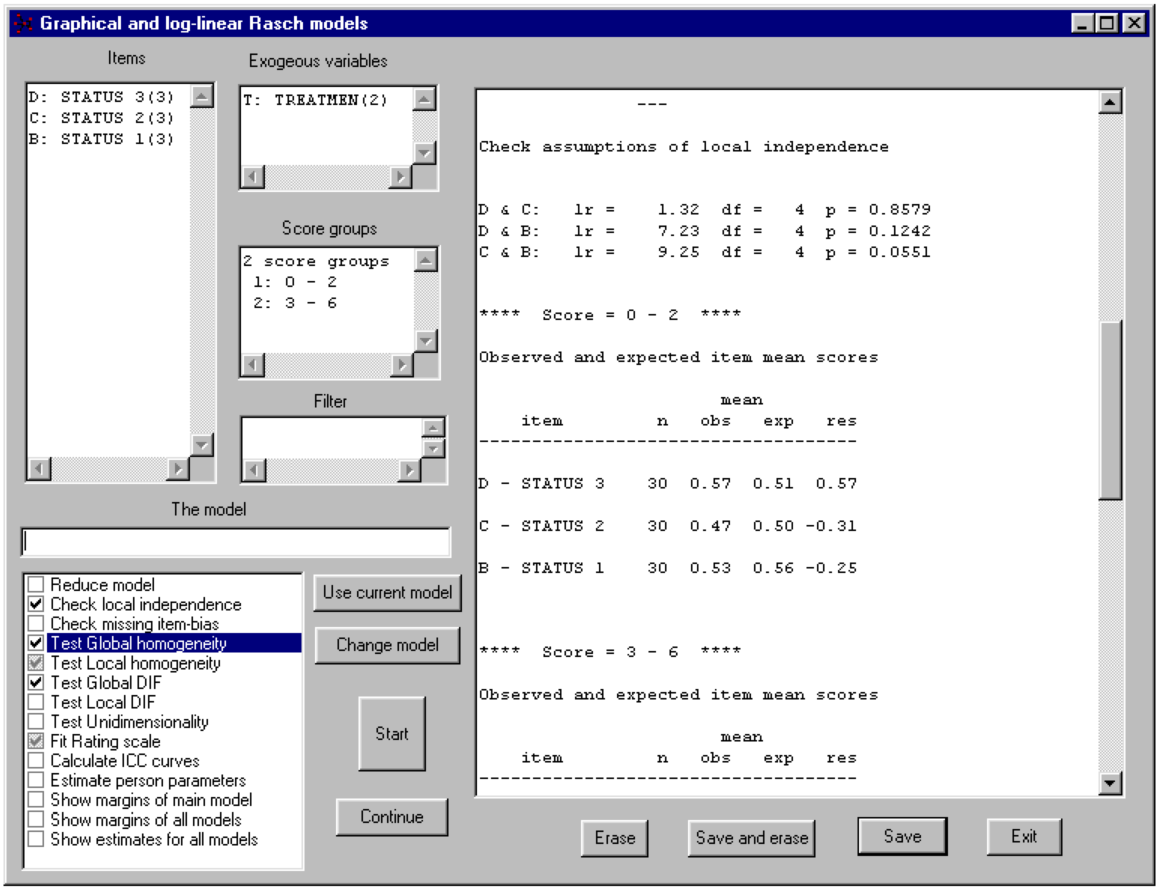
The GRM dialog is shown in Figure 46. The model that have been fitted here is a Rasch model assuming the side effects measured at different times are conditionally independent given a latent random effect representing susceptibility. We notice first that there are no evidence against local independence and second, that the test of global DIF[25] discloses no evidence of differential effect of treatment on frequency of side effects. The presence of side effects thus seems to be more a question of population heterogeneity and less a problem of autocorrelation.

Using a Rasch model for the UKU side effects is equivalent to a random effects model where you assume that the random effect is the same for all measurements of side effects.

[24] A preliminary version of the guide to DIGRAM's item analysis is included with the current release of SCD.

[25] Differential item functioning.

Figure 46 below show you the GRM dialogue and the test results indicating that such a model seems to give an adequate description of the variation of side effects after treatment.



Items:

```
D: STATUS 3(3)
C: STATUS 2(3)
B: STATUS 1(3)
```

Exogeous variables:

```
T: TREATMEN(2)
```

Score groups:

```
2 score groups
 1: 0 - 2
 2: 3 - 6
```

- ☐ Reduce model
- ☑ Check local independence
- ☐ Check missing item-bias
- ☑ Test Global homogeneity
- ☑ Test Local homogeneity
- ☑ Test Global DIF
- ☐ Test Local DIF
- ☐ Test Unidimensionality
- ☑ Fit Rating scale
- ☐ Calculate ICC curves
- ☐ Estimate person parameters
- ☐ Show margins of main model
- ☐ Show margins of all models
- ☐ Show estimates for all models

```
                  ---

Check assumptions of local independence


D & C:   lr =    1.32  df =   4  p = 0.8579
D & B:   lr =    7.23  df =   4  p = 0.1242
C & B:   lr =    9.25  df =   4  p = 0.0551


****  Score = 0 - 2  ****

Observed and expected item mean scores

                      mean
    item          n   obs   exp   res
------------------------------------

D - STATUS 3     30  0.57  0.51  0.57

C - STATUS 2     30  0.47  0.50 -0.31

B - STATUS 1     30  0.53  0.56 -0.25



****  Score = 3 - 6  ****

Observed and expected item mean scores

                      mean
    item          n   obs   exp   res
------------------------------------
```

Figure 46: ***ITEMS DCB, CUT 2, EXO T, GRM:*** *Items are selected, the score partitioned in two score groups, one exogenous variable (T) selected, and an analysis by graphical and loglinear Rasch models selected.*

Note, that you can only use DIGRAM to test the adequacy of the model. To estimate and analyze the random effect you have to export the data to other programs. For additional information on DIGRAM's item analysis see Kreiner (200?).

Gamma coefficients

One of the main features setting DIGRAM apart from most other programs for analysis of multidimensional contingency tables is the possibility for analysis of ordinal categorical data. These analyses are tied to the use of marginal and partial γ coefficients. To appreciate the scope and limitations of methods based on these coefficients one must of course know how they are defined. The section here is intended partly as a short introduction to γ coefficients for those who are not already familiar with γ coefficients and partly as a presentation of some new ideas about how these methods may be used.

The first thing to notice is that the γ coefficient is a non-parametric rank correlation. One may actually argue that γ is nothing but Kendall's τ in disguise. It differs somewhat in the way ties are treated, but is otherwise based on the same ideas. To calculate a γ coefficient for evaluation of the association between two ordinal variables, X and Y, we perform all possible pairwise comparisons of persons counting both cases and concordance and cases of discordance:

Let (x_i, y_i) and (x_j, y_j) be outcomes on X and Y for two different persons. Comparison of these outcomes lead to

concordance	if	$(x_i-x_j) \cdot (y_i-y_j) > 0$
discordance	if	$(x_i-x_j) \cdot (y_i-y_j) < 0$
tie	if	$(x_i-x_j) \cdot (y_i-y_j) = 0$

Goodman and Kruskall (1955) defines γ as

$$\gamma = \frac{C - D}{C + D}$$

where

C = the number of comparisons resulting in concordance
D = the number of comparisons resulting in discordance

and where the number of ties are disregarded.

γ can be interpreted as an estimate of a difference between two conditional probabilities for possible outcomes of comparisons between two persons,

$$\pi_{C|C\vee D} = P(\text{Concordance}|\text{Concordance or Discordance})$$

$$\pi_{D|C\vee D} = P(\text{Concordance}|\text{Concordance or Discordance})$$

$$\gamma \approx \pi_{C|C\vee D} - \pi_{D|C\vee D}$$

Davis (1967) extends Goodman and Kruskall's γ to a partial rank correlation in the following way.

Let X and Y be as before, but include a third variable, Z, and consider the problem of evaluation the conditional association between X and Y. Z may be a scalar or a vector.

Instead of comparing all persons as before, we now only compare persons having the same values on Z. This leads to a new definition of concordance and discordance

concordance	if	$(x_i-x_j)\cdot(y_i-y_j) > 0$ and $z_i = z_j$
discordance	if	$(x_i-x_j)\cdot(y_i-y_j) < 0$ and $z_i = z_j$
tie	if	$(x_i-x_j)\cdot(y_i-y_j) = 0$ and $z_i = z_j$
no comparison	if	and $z_i \neq z_j$

The definition of the γ coefficient is the same as before

$$\gamma = \frac{C_p - D_p}{C_p + D_p}$$

C_p = the number of comparisons with $z_i = z_j$ resulting in concordance

D_P = the number of comparisons with $z_i = z_j$ resulting in discordance

It is easy to see that the partial γ coefficient is a weighted mean of γ coefficients calculated separately for different values of Z and that the partial γ coefficient also can be interpreted as the difference between two conditional probabilities. We refer to Agresti (1984) for further discussion of these coefficients.

To calculate partial γ coefficient for all relationships between dichotomous and/or ordinal variables you can invoke the following command:

Gamma Produces a matrix of partial gamma coefficients for all pairs of ordinal or binary variables. Conditioning will always be with respect to minimal sets separators in the current project model. If more than one minimal separator sets exist, the gamma value presented will be the mean partial gamma coefficient for all sets.

Note that the estimated γ coefficients are partial coefficient estimated under the current model. Figure 47 shows the Gamma coefficients for the model shown in Figure 23. The output includes the same kind of model information that you would get from a SHOW G command to remind you about the conditions under which the coefficients were calculated.

The “Toggle gamma values on/off” in the Graph window becomes visible when γ coefficients have been calculated. If you press this button γ coefficients will be included in the graph for all existing edges or arrows. Note, that γ coefficients for non-existing connections between variables will not be show even though they have been calculated and. Also, γ coefficients will not be shown for nominal variables with more than two categories.

Figure 48 shows the same model as in Figure 23 with the γ coefficients of Figure 47 included.

```
Project graph

Variables   DCBAT
STATUS 3: D *+  +
STATUS 2: C +*+ +
STATUS 1: B  +*++
STATUS 0: A   +*+
TREATMEN: T ++++*
------------------------------
Log-linear generators/cliques:
LEVEL 1: (DCT),(CBAT)
LEVEL 2: (CBT),(BAT)
LEVEL 3: (BAT)
LEVEL 4: (AT)
------------------------------

Partial Gamma coefficients calculated under the above model

             D      C      B      A      T

D:STATUS 3   .      0.588  0.418  0.301  0.481
C:STATUS 2  0.588   .      0.631 -0.068  0.470
B:STATUS 1  0.418  0.631   .      0.603  0.520
A:STATUS 0  0.301 -0.068  0.603   .      0.492
T:TREATMEN  0.481  0.470  0.520  0.492   .
```

*Figure 47. **GAMMA:** Partial γ coefficients for the model in Figure 23.*

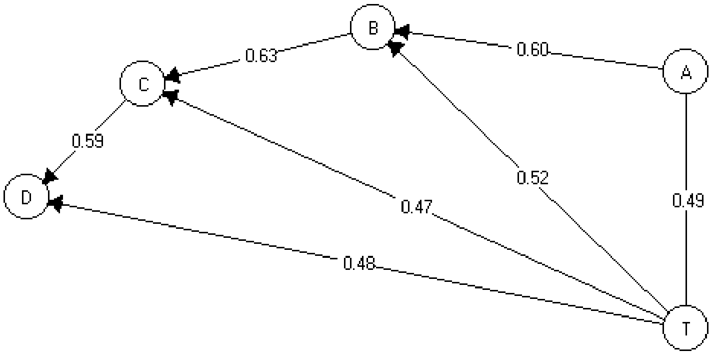


Figure 48. Interaction graph with gamma values toggled on.

γ coefficients are used throughout DIGRAM for two different reasons. First, γ coefficients used as test statistics for tests of (conditional) independence of dichotomous and ordinal variables have a much stronger power against alternatives assuming monotonous relationships between variables than standard tests assuming that all variables are nominal. Second, γ coefficients are informative descriptive measures – correlation coefficients – summarizing trends in a much more manageable way than observed tables of counts or tables of parameters of logistic and loglinear models.

The problem with γ coefficients is that they are *non-parametric* measures of association. This means that comparison of γ coefficients calculated under different conditions not always will be meaningful in the same was as comparisons of parameters of loglinear and logistic models will be. The fact for instance that γ coefficients describing the strength of association between two variables, A and B, are the same for men and women does not automatically imply that there is no higher order interaction between Sex and A and B. To make matters even more complex the opposite is also true: A difference between γ coefficients for men and women does not imply that there has to a higher order interaction between Sex and A and B.

Restrictions on γ coefficient do not in general define the type of loglinear models that we are using for analysis in DIGRAM.

There is, however, one exception to this dismal situation. For two dichotomous variables an analysis based on partial γ coefficients will be formally equivalent to a so-called Mantel-Haenszel analysis of odds ratio values defining proper loglinear and/or logit models.

To see this consider a simple 2×2 table of counts,

a	b
c	d

The odds-ratio, ω, for association in such a table is given by the cross-product ratio,

$$\omega = \frac{ad}{bc}$$

The γ coefficient on the other hand is given by

$$\gamma = \frac{ad - bc}{ad + bc}$$

which is well-known as Yule's Q. It is easy to see, that γ may be rewritten as a function of ω and visa versa:

$$\gamma = \frac{ad - bc}{ad + bc} = \frac{(\frac{ad}{bc} - 1)}{(\frac{ad}{bc} + 1)} = \frac{\omega - 1}{\omega + 1}$$

$$\omega = \frac{1 + \gamma}{1 - \gamma}$$

The logit value $\beta = \ln(\omega)$ is approximately equal to $2 \cdot \gamma$ for moderate values of β.

If we assume that γ and ω values are constant across different levels of one or more stratifying variables define a loglinear model with no higher order interaction involving the two dichotomous variables. The common odds ratio value may be estimated by the so-called Mantel-Haenszel estimator, ω_{MH}, which is a weighted mean of odds ratio values calculated in each strata, while the partial γ coefficient, γ_{par} – a weighted mean of γ coefficients in the different strata – estimates the common γ value.

The weights used for these two statistics do not result in estimates being one-to-one functions of each other. Both estimates are, however, consistent and the results should therefore be fairly close to each other:

$$\gamma_{par} \sim \frac{\omega_{MH} - 1}{\omega_{MH} + 1}$$

If you want to compare partial γ coefficients to Mantel-Haenszel estimates for binary variables you can use the following command. Remember that the relationship between Mantel-Haenszel estimates and partial γ coefficients require that there is no higher order interaction. If the estimates are different it suggests that higher order interaction – effect modification – is at work.

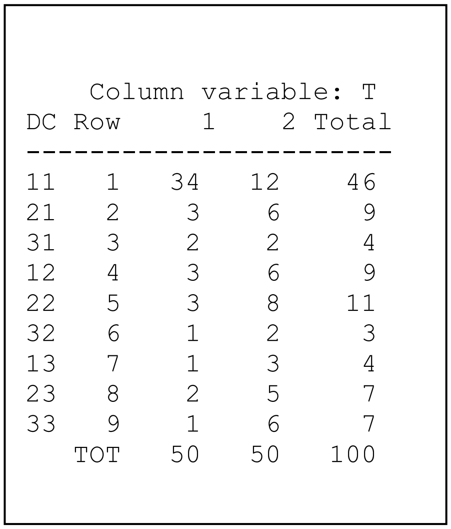
MANTEL-HAENSZEL var1 var2	Calculates both Mantel –Haenszel estimates and partial gamma coefficient for binary variables.

For ordinal variables with more than two categories the situation is more complicated. Instead of working directly with observed partial γ coefficients we suggest that an analysis of partial γ coefficients calculated for tables fitted to specific loglinear models might be helpful. You can find traces of this idea in some of the results from the DESCRIBE procedure in the current version of DIGRAM and it will be developed further as part of the procedure for analysis by loglinear models under development.

Analysis of category collapsibility in multidimensional contingency tables

This section gives a brief introduction to the problem of category collapsibility and how it may be analyzed by DIGRAM. A more systematic discussion of the whys and hows of analysis of category collapsibility is provided by Kreiner and Gundelach (200?).

The table in Figure 48 shows the distribution of UKU side effects (C and D) at the second and third measurement for the two different drugs. Note that the table is organized as a two-way table with drugs in the columns and with rows defined by a stacked variable combining side effects at the two different measurements.



```
    Column variable: T
DC Row     1    2 Total
-----------------------
11   1    34   12    46
21   2     3    6     9
31   3     2    2     4
12   4     3    6     9
22   5     3    8    11
32   6     1    2     3
13   7     1    3     4
23   8     2    5     7
33   9     1    6     7
   TOT    50   50   100
```

Figure 48. UKU side effects for two different drugs

We say that a subset of row categories is collapsible if the conditional distribution of the column variable is the same for all categories within the subset. It is immediately seen that rows 2, 4, and 6 are collapsible in Figure 48. Collapsibility of column categories may be defined in the same way as for row categories. In the example used here collapsibility of the two column categories implies that side effects are independent of drug.

The analysis of category collapsibility performed by DIGRAM is an automatic stepwise procedure where each step consist of

1) pairwise comparison of row categories
2) pairwise comparison of column categories
3) creation of a new table where the least significant row or column categories are collapsed if the Boferroni corrected p-value is larger than 0.05.

The analysis may be invoked by two different commands:

COLLAPS column variable row variables Stepwise analysis of category collapsibility. Both column and row categories will be collapsed.

MCA Stepwise analysis of category collapsibility. Collapse of row or column collapsibility may be deselected.

Figures 49 – 52 show the result of the analysis of the table in Figure 48 following a COLLAPS TDC command. The difference between the MCA command and the COLLAPS command will be described at the end of this section.

The test results motivating collapse of categories are shown in Figure 49. Rows 2,4, and 6 are collapsed in the first two steps. The third step collapses rows 5 and 8. The fourth step collapses row 7 with row 5+8. Rows 2+4+6 are collapsed with rows 5+7+8 in fifth step. Step 6 and 7 adds first row 3 and second row 9 to the collapsed row creating one row combining all rows from 2 to 9. Finally, the last step rejects collapsibility of row no. 1 and rows 2-9. Row no. 1 contains persons without side effects. The end result therefore is the 2×2 table shown in Figure 50 summarizing the difference between drugs to a question of whether or not there are side effects at all.

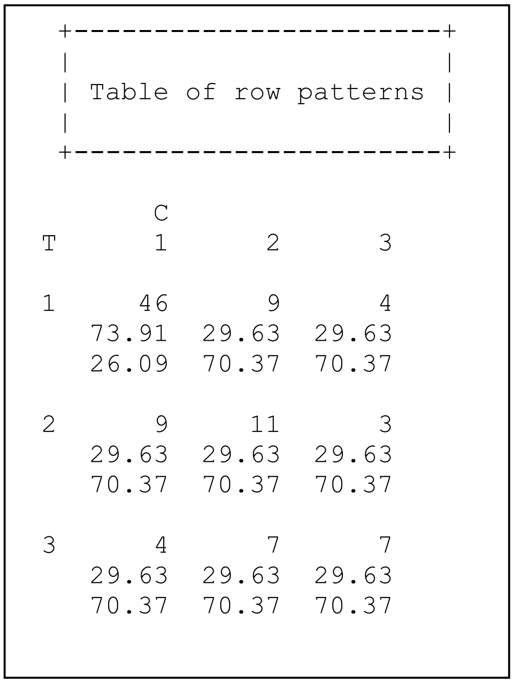
```
  +---------------------------------------------------------------+
  |                                                               |
  | Analysis of collapsibility of categories by unrestricted MCA |
  |                                                               |
  +---------------------------------------------------------------+

    lr df      p   gamma p(2-sided)
  -----------------------------------------------
  -0.0  1 1.0000   0.000 1.0000 rows  :  2 and 4
   0.0  1 1.0000   0.000 1.0000 rows  :  2 4 and 6
   0.0  1 0.9522  -0.032 0.9523 rows  :  5 and 8
   0.0  1 0.9095   0.071 0.9083 rows  :  5 8 and 7
   0.2  1 0.6653   0.143 0.6651 rows  :  2 4 6 and 5 7 8
   0.6  1 0.4320  -0.395 0.4697 rows  :  2 4 5 6 7 8 and 3
   1.0  1 0.3116   0.475 0.2641 rows  :  2 3 4 5 6 7 8 and 9
  20.2  1 0.0000   0.741 0.0000 rows  :  1 and 2 3 4 5 6 7 8 9

no further collaps:  Bonferoni adjusted pmax = 0.0000


Test results for all remaining rows and columns


    lr df      p   gamma p(2-sided)
-----------------------------------------------
  20.2  1 0.0000   0.741 0.0000 rows   : 1 and 2 3 4 5 6 7 8 9
```

Figure 49. Stepwise collapse of the rows of the table in Figure 48

```
Collapsed row distributions (sorted):

 Row     1     2       n    Rows included

   2  29.6  70.4      54    2 3 4 5 6 7 8 9
   1  73.9  26.1      46    1

Collapsed column distributions (sorted):

 Row     1     2            Rows included

   2  32.0  76.0            2 3 4 5 6 7 8 9
   1  68.0  24.0            1
```

Figure 50. The collapsed table.

The result of the analysis is finally summarized in a table, Figure 51, containing the row variables where each cell contains the number of cases and the distribution of the column variable. It is not useful in this particular case, but is often useful when the results are more complicated than those shown in Figure 50.



```
 +-----------------------+
 |                       |
 | Table of row patterns |
 |                       |
 +-----------------------+


        C
T       1      2      3

1      46      9      4
    73.91  29.63  29.63
    26.09  70.37  70.37

2       9     11      3
    29.63  29.63  29.63
    70.37  70.37  70.37

3       4      7      7
    29.63  29.63  29.63
    70.37  70.37  70.37
```

Figure 51. Row patterns after collapse

The analysis illustrated above is based on unrestricted comparisons of rows where each row category is compared with all other categories. If one or more of the variables incl-uded in the rows is an ordinal variable the analysis will be repeated utilizing a strategy for restricted comparisons of nearest neighbors.

The row categories are defined and ordered in the setup below:

	D		
C	1	2	3
1	**1**	**2**	**3**
2	**4**	**5**	**6**
3	**7**	**8**	**9**

Both row variables are ordinal variables. The strategy for restricted comparisons only permits comparisons of rows 1 & 2, 1 & 4, 2 & 3, 2 & 5, 3 & 6, 4 & 5, 4 & 7, 5 & 6, 5 & 8, 6 & 9, 7 & 8 and 8 & 9. If C had been nominal instead of ordinal comparisons would have been permitted for rows 1 & 7, 2 & 5 and 3 & 9 as well.

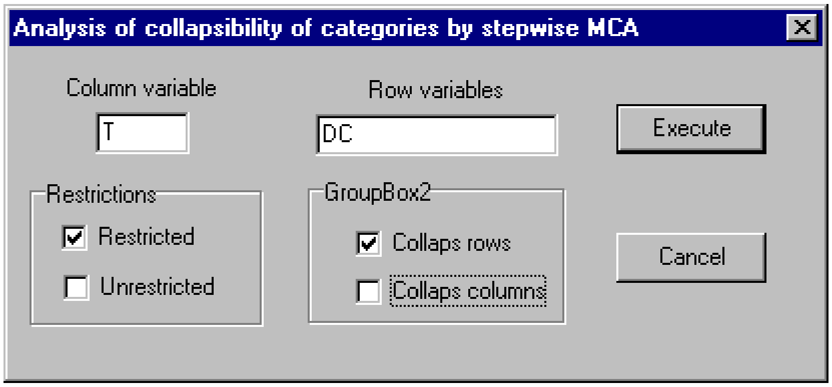
Figure 52 below summarizes the results of the restricted analysis. The end result is in this case the same as for the unrestricted analysis, but the steps are somewhat different.

Rows 5 and 8 are collapsed first. The nearest neighbors of the collapsed (5+8) is rows 2, 4, 6, 7, and 9. In the second step all these rows are consequently compared to (5+8). Figure 52 shows that row (5+8) and row 7 are collapsed in the second step.

```
    lr df      p   gamma p(2-sided)
-----------------------------------------------
   0.0  1 0.9522  -0.032 0.9523 rows  :  5 and 8
   0.0  1 0.9095   0.071 0.9083 rows  :  5 8 and 7
   0.0  1 0.8290  -0.143 0.8343 rows  :  5 7 8 and 6
   0.1  1 0.7651   0.125 0.7686 rows  :  2 and 5 6 7 8
   0.1  1 0.8210  -0.091 0.8234 rows  :  2 5 6 7 8 and 4
   0.6  1 0.4320  -0.395 0.4697 rows  :  2 4 5 6 7 8 and 3
   1.0  1 0.3116   0.475 0.2641 rows  :  2 3 4 5 6 7 8 and 9
  20.2  1 0.0000   0.741 0.0000 rows  :  1 and 2 3 4 5 6 7 8 9

no further collaps:  Bonferoni adjusted pmax = 0.0000
```

Figure 52. Restricted stepwise MCA

The MCA command gives you a little more freedom to determine how the analysis of category collapsibility should proceed. Figure 53 shows the dialogue form shown after the MCA command where you must enter the column and row variables and decide whether or not both column and rows should be compared and whether or not you want an restricted and/or unrestricted analysis.


Analysis of collapsibility of categories by stepwise MCA

Column variable: T

Row variables: DC

Restrictions
☑ Restricted
☐ Unrestricted

GroupBox2
☑ Collaps rows
☐ Collaps columns

Figure 53. The MCA dialogue

It may be argued that the stepwise analysis of category collapsibility illustrated above is an example of the most shameless type of exploratory analysis conceivable. I do not unreservedly subscribe to this point of view, but I nevertheless think that it is fair to issue some cautious remarks concerning the use of this procedure. The analysis is exploratory. It is meant to simplify the description of fairly complex relationships and it should only be used when other evidence, e.g. significant tests for higher order interaction, imply that relationships actually are complex. It is not meant as a procedure for digging out evidence of complex higher order interactions. The sometimes astronomical number of comparisons to be made and the large number of cells rows with fairly few cases, implies an uncomfortably high risk of making a large number of type II errors along the way.

As I see it, analysis of category collapsibility is a descriptive analysis that you may use to describe the results of the analysis. It may however also be useful in the initial data analysis when you have to decide how many categories you need for the project variables. If two or more categories of a variable, A, can be collapsed in all two-way tables including A and one of the other project variables then it suggests that A should be redefined for the project by collapse of these categories.

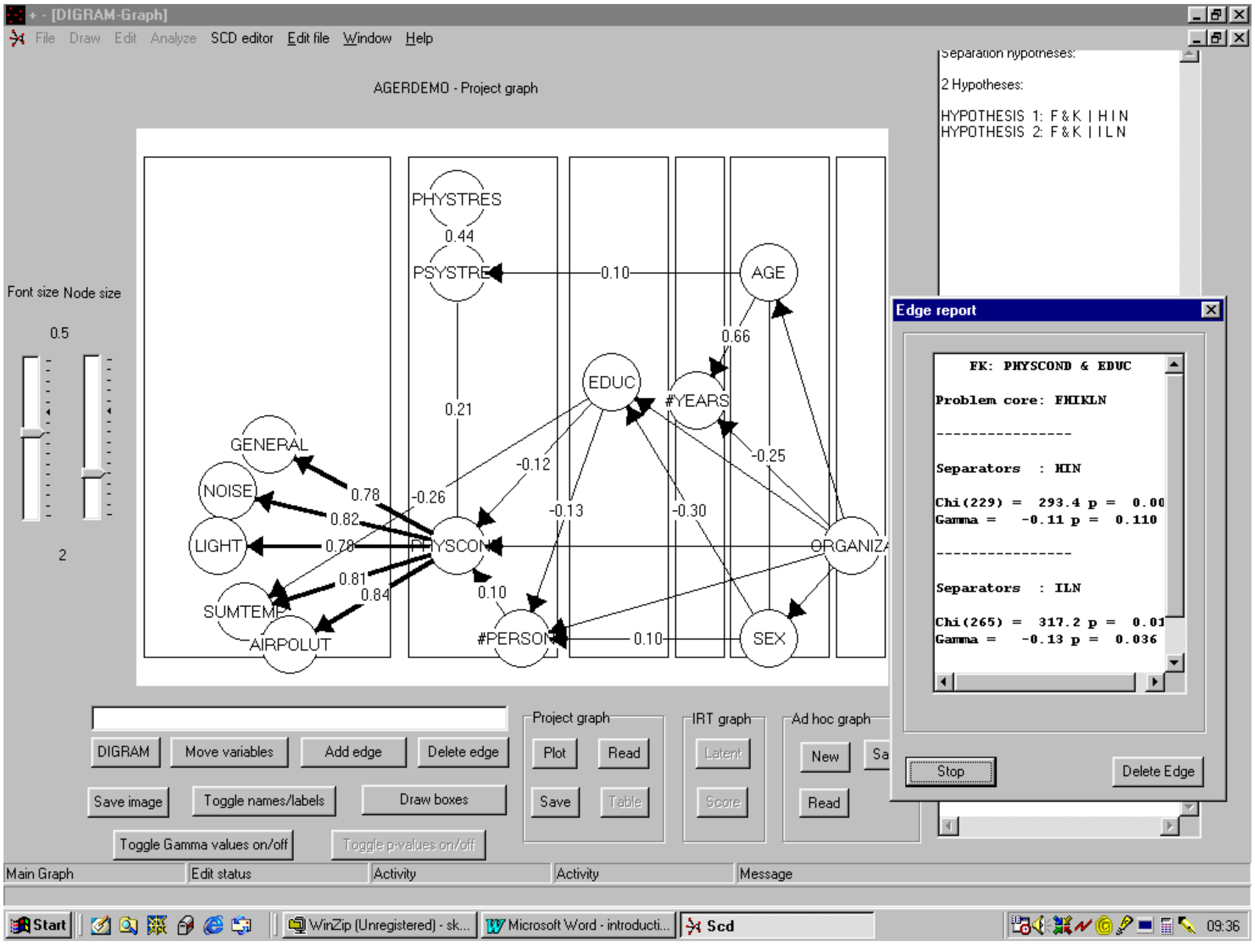
To see whether this is possible you may use the COLLAPS command with reference to nothing but A:

COLLAPS variable	Analysis of collapse of the categories of the variable in all possible two-way tables containing the variable.

The output of following this command contains of test results comparing all pairs of categories of the variable followed be output from the stepwise analysis described above.

Appendix 1: The Graph editor

The appearance of the graph shown in Figure 9 may be edited in several different ways using the buttons and the track bar of the graph form. Figure A.1 below thus shows the same graph as Figure 9 with boxes drawn around the recursive blocks, with variable names rather than variable labels and with the with the position of some variables changed to fit the boxes, with γ coefficients for all edges of the model. Node sizes have also been adjusted the better to accommodate the variable names. Finally a right click on the edge between the PHYSCOND and EDUC variables produced the edge report shown to the right of the graph.



```
Separation hypotheses:

2 Hypotheses:

HYPOTHESIS 1: F & K | H I N
HYPOTHESIS 2: F & K | I L N
```

```
     FK: PHYSCOND & EDUC

Problem core: FHIKLN

----------------

Separators  : HIN

Chi(229) =  293.4 p =  0.00
Gamma =   -0.11 p =  0.110

----------------

Separators  : ILN

Chi(265) =  317.2 p =  0.01
Gamma =   -0.13 p =  0.036
```

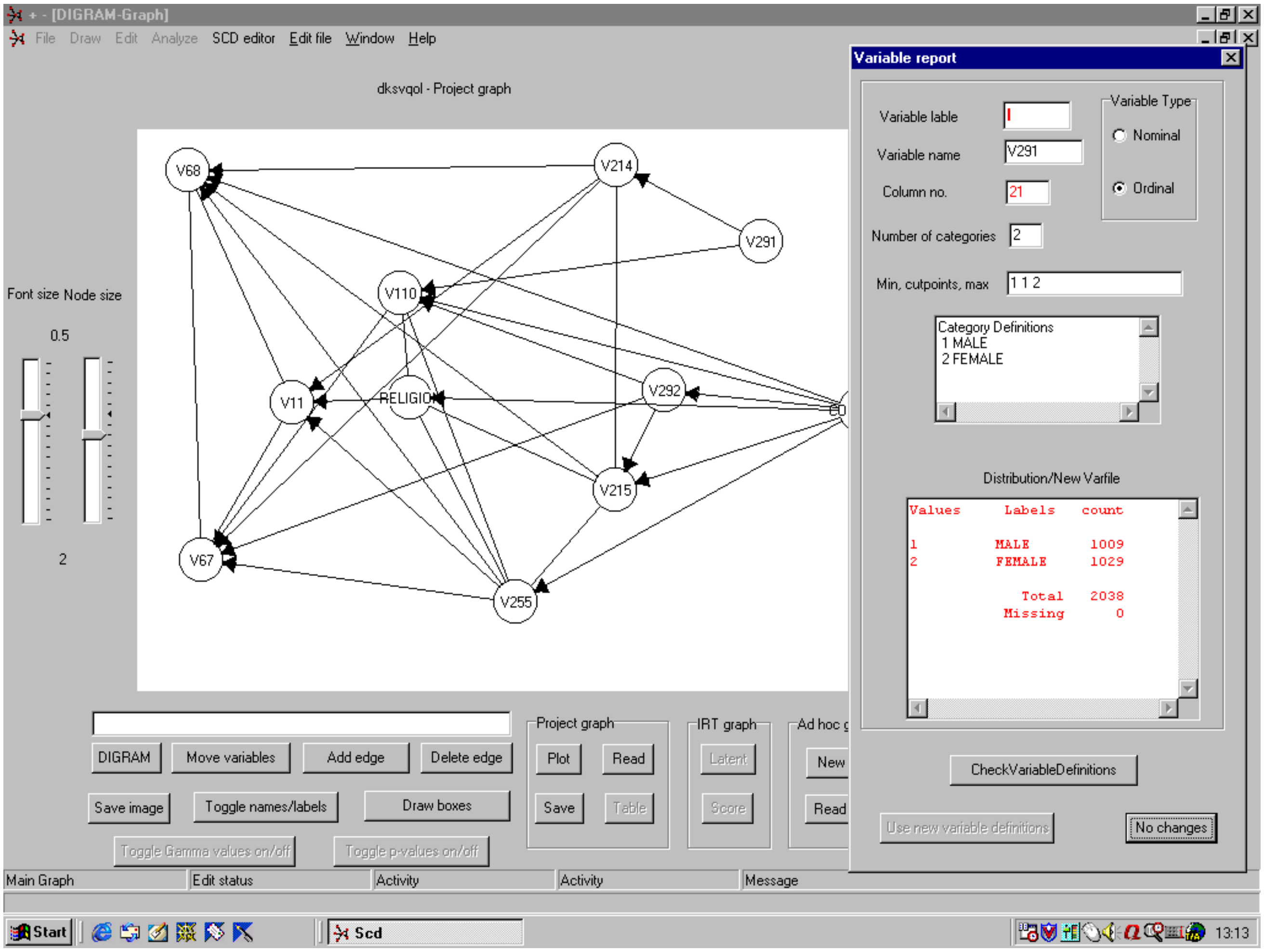
Figure A1. An edited version of the independence graph shown in Figure 9

Three buttons permit you to move nodes and delete or add edges. Having clicked one of these buttons you must first click on one node and then either click on a new position of this node or click on another node where you either want to add or delete a connection. The color of the first node selected changes to remind you what you are doing. A node to be moved becomes black. A node attached to an edge that you want to include in the graph turns green while a node connected to an edge that you want to delete turns red.

The two track bars to the left of the graph may be used to change the size of the nodes and the size of the font used for variable labels and names.

Rightclicking on a variable produces the variable report form shown in Figure A.2 showing both the definition and the marginal distribution of the variable.The information written in black may be edited. In the example in Figure A.2 V291 is changed to Sex. Click on the "Check variable definitions" button to make sure that the edited information is acceptable. If this is the case the contents of the new VAR file will be shown instead of the distribution of the variable and the "Use new variable definitions" button will be enabled. Click on this button to implement the edited variable definitions. The VAR and/or CAT files will be overwritten depending on the changes you have suggested. New SYS and TAB files will be generated if you have changed the number of categories or the cutpoints defining categories.

WARNING: This procedure has not been properly tested yet. You are therefore advised to make a copy of the variable definitions before you try to use this.



Variable lable

Variable name V291

Column no. 21

Number of categories 2

Min, cutpoints, max 1 1 2

Variable Type
○ Nominal
● Ordinal

Category Definitions
1 MALE
2 FEMALE

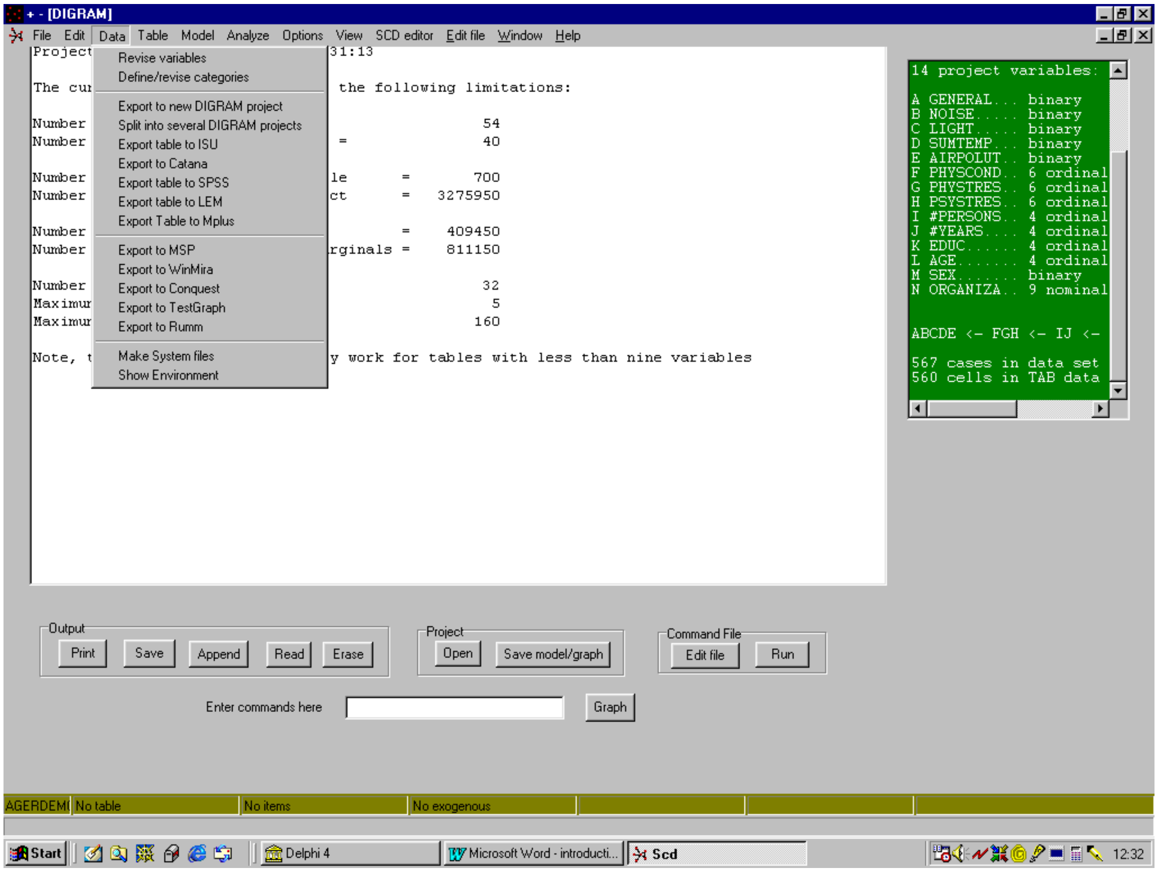
```
Values      Labels    count

1           MALE       1009
2           FEMALE     1029

               Total   2038
            Missing       0
```

Figure A.2 Variable report after Rightclicking on V291

Appendix B: Coexisting with other programs

Instead of spending time on development of software for methods that already exists in other programs it makes more sense simply to move to these programs for the analysis. To make the process of moving between easier you may use DIGRAM to generate files that these programs may read.

Click on the Data menu to get an idea about the programs currently being supported in this way by DIGRAM. We are not quite ready with everything here. Some of these export functions have not been tested yet and the list may eventually be expanded, but Figure B1 nevertheless should give you an idea about what we are aiming at. DIGRAM will issue a short report on the files created for the other programs.



```
Project                                  31:13

The cur                                   the following limitations:

Number                                                        54
Number                                    =                   40

Number                                   le       =          700
Number                                   ct       =      3275950

Number                                            =       409450
Number                                   rginals  =       811150

Number                                                        32
Maximur                                                        5
Maximur                                                      160

Note, t                                  y work for tables with less than nine variables
```

```
14 project variables:

A GENERAL...  binary
B NOISE.....  binary
C LIGHT.....  binary
D SUMTEMP...  binary
E AIRPOLUT..  binary
F PHYSCOND..  6 ordinal
G PHYSTRES..  6 ordinal
H PSYSTRES..  6 ordinal
I #PERSONS..  4 ordinal
J #YEARS....  4 ordinal
K EDUC......  4 ordinal
L AGE.......  4 ordinal
M SEX.......  binary
N ORGANIZA..  9 nominal

ABCDE <- FGH <- IJ <-

567 cases in data set
560 cells in TAB data
```

Figure B1. List of export items on the Data menu

References

Agresti. A. (1984). *Analysis of Ordinal Categorical data.* New York: John Wiley & Sons.

Agresti, A. (1990). *Categorical Data Analysis.* New York: John Wiley & Sons.

Cox, D. and Wermuth, N. (1996). *Multivariate Dependencies. Models, analysis and interpretation.* London: Chapman and Hall.

Darroch, J.N., Lauritzen, S.L. and Speed, T.P. (1980). Markov fields and log-linear models for contingency tables. *Annals of Statistics,* **11**, 724-38

Edwards, D. and Kreiner, S. (1983). The Analysis of contingency tables graphical models. *Biometrika,* **70**, 553 – 565.

Edwards, D. (1995,2000). *Introduction to Graphical Modelling*. New York: Springer

Everitt, B.S. (1977). *The Analysis of contingency tables.* London: Chapman and Hall.

Kreiner, S. (1987). Analysis of multidimensional contingency tables by exact conditional tests: techniques and strategies. *Scandinavian Journal of Statistics,* **14**, 97-112.

Kreiner, S. (1996). An informal introduction to graphical modelling. In Knudsen, H.C. and Thornicroft, G. (1996): *Mental Health Service Evaluation.* Cambridge University Press, 156 – 175.

Kreiner, S. (200?a) *Item analysis*. DIGRAM user guides.

Kreiner, S. (200?b). *Analysis of homogeneity of repeated and similar measurements.* DIGRAM user guides.

Kreiner, S. (200?c). *Multidimensional Markov Chains.* DIGRAM user guides.

Kreiner, S. & Gundelach, P. (200?d). *Analysis of category collapsibility in multidimensional contingency tables.* DIGRAM user guides.

Lauritzen, S. L. (1996). *Graphical Models.* Oxford: Clarendon Press.

Wermuth, N. (1993). Association Structures with Few variables: Characteristics and Examples. In Dean, K. (1993): *Population Health Research. Linking Theory and Methods.* London. Sage Publications.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics.* Chichester: John Wiley & Sons.

Whittaker, J. (1993). Graphical Interaction Models: a New Approach for Statistical Modelling. In Dean, K. (1993): *Population Health Research. Linking Theory and Methods*. London. Sage Publications.