STRUCTURE

A General Purpose Network Analysis Program
Providing Sociometric Indices, Cliques,
Structural and Role Equivalence, Density Tables,
Contagion, Autonomy, Power and Equilibria
in Multiple Network Systems

Version 4.2

REFERENCE MANUAL

Columbia University
New York, NY 10027

Version 4.1, Columbia University, 1989
Version 4.0, Columbia University, 1989
Version 3.2, Columbia University, 1987
Version 3.1, Columbia University, 1986
Version 3.0, Columbia University, 1984
Version 2.0, University of California at Berkeley, 1981
Version 1.0, University of Chicago, 1975

STRUCTURE Version 4.2 contains 12,551 lines of source code compiled for IBM microcomputers and compatibles on equipment obtained with consulting revenues to Ronald Burt, grants to Ronald Burt from the National Science Foundation Sociology Program, and gifts from the International Business Machines Corporation to Columbia University's AURORA and BOREALIS projects. Source code was compiled with Microsoft FORTRAN and Benjamin R. Strong Jr.'s FORTRAN Toolkits (available from IMPULSE Engineering). The source code and manuals were written by Ronald Burt, except the *Reference Manual* Network Models section written by Thomas Schøtt, with consulting services or donated contributions from the following people on the indicated sections of the code; Russell Bernard and Peter Killworth (rank order sociometric data), Miguel Guilarte (cluster analysis, debugging, eigenvectors, random numbers and spatial display), Xia Fang (debugging, power and t-tests), Stephen Johnson (cluster analysis), Nan Lin (path distances), Peter Marsden (eigenvectors), Mark Mizruchi (reflected prominence), Ronan van Rossem (triad pattern frequencies), Seymour Spilerman (eigenvectors), and Tetsuji Uchiyama (debugging and density tables).

CONTENTS

APPENDICES

This *Reference Manual* has three sections. Following some brief introductory remarks here, the first section, COMMANDS, is an annotated alphabetical list of the seventeen program commands with which you tell STRUCTURE how to perform an analysis. The second section, NETWORK DATA, describes kinds of data used in network analyses and the alternative strategies available in STRUCTURE for deriving network relations from raw data. The third section, NETWORK MODELS, describes the results you can obtain with the analytical procedures provided in STRUCTURE. You can use the Table of Contents to locate sections of the *Reference Manual* relevant to your interests, or locate specific topics through the index at the end of the *Manual*.

Before getting into that material, however, let me give you a sense of perspective. What is it you can expect to get out of a network analysis? How is that different from what you would get if you just carried out the usual survey type analysis? I'll sketch answers to these questions, refer you elsewhere for more complete answers, then move immediately to running STRUCTURE so you can begin doing network analyses — and so learn the answers for yourself. Experienced network analysts might prefer to skip to **Running the Program** on page 14.

NETWORK ANALYSIS

The central tenet of network analysis is that the causal motor behind what people feel, believe and do lies in the patterns of relations between actors in a situation, as opposed to the attributes of the individual actors. Your race, age, sex, and so on, matter less than the pattern of relations that position you in social structure. Actors here could be animals, people, groups, organizations, geographic regions, traffic lights, or any object capable of being linked by a relationship to other objects. Under the central tenet, any explanation of beliefs and behaviors requires an analysis of how actors are connected to one another in the situation where beliefs or behaviors are observed. Network analysis is a collection of descriptive and predictive models for doing just that.

For inexpensive, practical introduction, see *Applied Network Analysis*, edited by Burt and Minor (1983), and Knoke and Kuklinski's (1982) book *Network Analysis*. *Toward a Structural Theory of Action* (Burt, 1982), provides a more formal review of models. If you can read German, Pappi's (1987) book, *Methoden der Netzwerkanalyse*, is a superb collection of review articles on network methodology. The journal, *Social Networks*, is a valuable source for keeping up to date with contemporary issues in network methodology. The success of network analysis as a component in theoretical social science is most evident from network articles in core disciplinary journals, notably the *American Journal of Sociology*. Diverse substantive applications have been assembled in *Social Networks*, edited by Leinhardt (1977), *Social Structure and Network Analysis*, edited by Marsden and Lin (1983), *Social Structures*, edited by Wellman and Berkowitz (1988), and several books in Granovetter's Structural Analysis Series from Cambridge University Press.

Analysis starts with data sufficient to define: a system of actors and one or more networks of relations among the actors, where the variables measuring relations within each network are comparable in metric and content, and each relation variable measures the strength of a relation from one actor to another. Marsden's (1990) chapter in the *Annual Review of Sociology* is the most succinct review of network data. For more details, see Chapters 1 through 8 of *Applied Network Analysis*. In addition to Chapters 7 and 8 on measuring relations in terms of joint affiliations, see Breiger's (1974) succinct article on the duality of persons and groups. Burt (1988a) reviews alternative measures of relations derived from sociometric choice data. The article also illustrates Monte Carlo network analysis with target image matrices. If you have rank order sociometric data, see Killworth and Bernard (1974). The model used in STRUCTURE to derive relations from ranks is taken from the program that Killworth and Bernard developed and describe in their article. Kinds of data for a network analysis are covered in the Network Data section of this *Manual*.

Yes, But What Is the Substance of Network Analysis?

Contemporary network analysis spans five principles; cohesion, equivalence, prominence, range and brokerage. These define broadly distinct kinds of insights you can get from a network analysis. They are laid out in the table across pages 8 and 9. Each principle involves a structural intuition — an implicit causal motor behind beliefs and behaviors — and theoretical models that aggregate observed relations into structural images and causal variables. As with any intellectual sorting, one can quibble about the categories. However, these span the variety of network models most often used in contemporary research at the same time that they highlight intrinsic qualities of network analysis that make it distinct from other analytical strategies. Substantively, models within the five principles describe three phenomena; boundaries, bonds (underlying shared beliefs) and behaviors.

Cohesion and equivalence are principles for defining social boundaries. In the principles table, squares indicate actors and boxes are drawn around actors to be grouped together as the same kind of actor. As you can see in the table, cohesion draws boundaries around actors tied to one another by strong direct and indirect relationships within cliques. The alternative criterion is equivalence. Equivalence draws boundaries around actors who share a similar pattern of direct and indirect, present and absent, relations with other actors. Their similar relations may or may not include strong relations with one another. In the equivalence column of the table, a box surrounds two actors who are equivalent. They have no direct contact with one another, but similar relations with the same other actors in the system. The others with whom equivalent actors are similarly related can be defined at two levels. Actors are *structurally equivalent* to the extent that they have similar relations with every other individual actor. Actors are *role equivalent* to the extent that they have similar relations with similar kinds of actors. Drawing boundaries by equivalence criteria is the basis for operationalizing status/role-sets in network analysis, sometimes discussed as position detection to distinguish it from the clique detection.

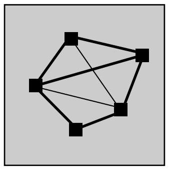
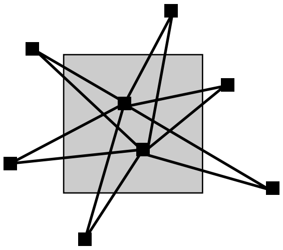
The Cohesion and Equivalence sections in this *Manual* are an introduction to detecting cliques, positions, roles and blockmodels. The CLIQUES command provides popular models for clique analysis. The POSITIONS command provides popular models, plus some more sophisticated models, for structural and role equivalence analysis. These sections are a stand-alone introduction for someone without network training, but references are provided for more detailed reading.

Cohesion and equivalence are the principles defining bonds — relations of interpersonal influence and contagion. Research here concerns the strength of, and social mechanism behind, contagion effects. The first concern asks how much responses on a criterion variable are determined by social contagion. The second concern asks how the observed contagion effect was created. Contagion by cohesion arises from people discussing opinions in strong, socializing relationships, and so coming to a shared opinion. Contagion by equivalence arises from people in the same structural position playing one another's roles and so coming to a shared opinion.

A contagion analysis is requested with the CONTAGION command, in combination with a CLIQUES or POSITIONS command defining bonds. A wide range of alternative contagion processes and output options are available with these commands as discussed in the Contagion section of this *Manual*. This introduction to contagion analysis can stand alone, however, I defer to a recent book for detailed discussion and illustration of substantive and methodological points (Burt, 1992b).

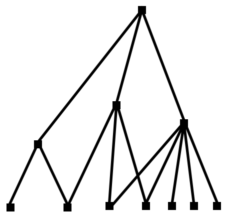
Principles of

BOUNDARIES, BONDS,

	 **COHESION**	 **EQUIVALENCE**
STRUCTURAL INTUTION	Similar actors are tied together by socializing bonds of interaction through which they come to share beliefs and behavioral tendencies. Causal force lies in the strength of communication ties.	Similar actors have similar patterns of relations with others. Pattern similarity defines social boundaries around reference groups and feelings of relative deprivation, creating homogeneous beliefs and behavioral tendencies among equivalent actors. Causal force lies in role playing within a shared frame of reference
AGGREGATION IN A THEORETICAL MODEL TO GUIDE EMPIRICAL RESEARCH	Sum relations or chains of relations to measure social integration. Primary groups are identified as cliques and cross-cutting relations provide social integration (Luce & Perry 1949; Festinger et al. 1950; Coleman et al. 1957; Flament 1963; Harary, Norman & Cartwright 1965; Alba 1973; Alba & Kadushin 1976; Blau 1977)	Sum weighted differences between corresponding relations in two patterns to measure distance (typically Euclidean) between the patterns. Structural and role equivalent actors are separated by zero distance (Lorrain & White 1971; White, Breiger & Boorman 1976; Burt 1976, 1977, 1982, 1987, 1992b; Winship & Mandel 1983; White & Reitz 1983; Hummell & Sodeur 1987; as well as a great many of the articles in the journal *Social Networks*)

Network Analysis

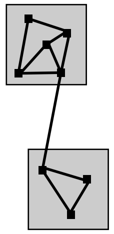
AND BEHAVIORS



PROMINENCE

Prominence comes from being the object of relations from prominent contacts. Causal force lies in the direct and indirect system demand for relations with an individual actor.

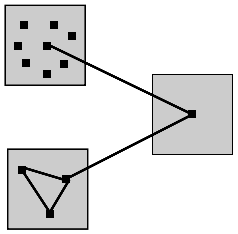
Sum relations weighted by the prominence of source (typically eigenvector). Concepts of power, prestige, centrality, etc. are measured by network prominence (Katz 1953; Hubbell 1965; Coleman 1966, 1972; Bonacich 1972, 1987; Marsden 1984; Mizruchi et al. 1986)



RANGE

Communication between kinds of actors occurs over bridge (typically weak) relations. Bridge relations are a social resource. Causal force lies in relational access to valued resources.

Sum relations to measure contact with other kinds of actors. Rules are vague, typically ad hoc to specific research projects. Defines network range (access to diverse kinds of social resources) reflecting the strength of weak ties (Granovetter 1973; Berkman & Syme 1979; Lin et al. 1981)



BROKERAGE

Holes in social structure are entrepreneurial opportunities to promote and take advantage of competition between contacts. Causal force lies in contact with disorganized others.

Sum relations weighted by their number and disorganization, where, ceteris paribus, number of contacts increases holes and density of ties among contacts decreases holes. Defines structural autonomy; relative ability to negotiate relations to personal advantage (Freeman 1977; Cook & Emerson 1978; Burt 1980, 1982, 1992a)

The remaining three network principles predict behavior by measuring an actor's power or freedom to act. Prominence models place individuals on the vertical axis of organization; leaders at the top, the led at the bottom. The causal mechanism is demand. To the extent that an individual is the object of relations from many others, who are in turn the object of many relations, there is high demand for the actor and she can use that demand to push others into carrying out her wishes.

The POWER command provides popular prominence models, including the eigenvector models associated with Bonacich (1972) and Coleman (1972), and Mizruchi et al.'s (1986) distinction between reflected and derived prominence. In addition, Coleman's exchange model is used to measure the stability of an observed network; under independence (as in Coleman's original model), and allowing substitutions between actors (Burt, 1979). The Power section of this *Manual* can stand alone as an introduction for someone without network training.

Range and brokerage models measure a different action potential; the extent to which individuals have opportunities to avoid or negotiate control efforts by others. The simplest range measure is a sum of an actor's relations. The more relations you have, the more access you have to social resources. More sophisticated models search for bridging ties. In the range column of the principles table, for example, the one link between boxes is the key to each group of actors having access to the other. Recent developments have extended this theme to brokerage. Under this principle, an individual actor has the freedom to pursue his own interests to the extent that his relations connect him with people who are disorganized and so can be played off against one another, freeing him of their constraint. In the brokerage column of the principles table, the actor alone in a box has high negotiating flexibility in his relation with the disorganized actors in the top box and less flexibility with the completely interconnected actors in the bottom box.

The AUTONOMY command provides measures of size, density and brokerage opportunities in an individual's network. The measures are described in the Autonomy section of this *Manual*, keyed to detailed discussion and illustration in a recent book on the role of network size, density and brokerage in competitive arenas (Burt, 1992a).

So What?

Network analysis adds two kinds of value to social science research. It adds theoretical clarity and stronger, more cumulative, research results.

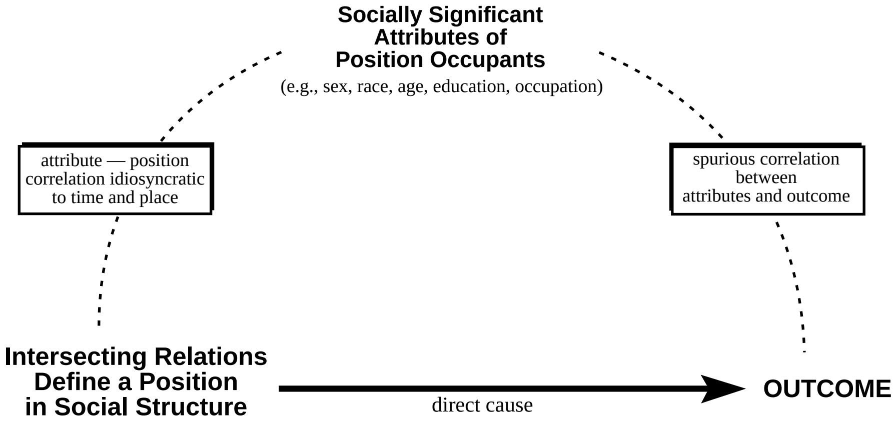
In network analysis, causal force originates in the structure of relations that define a situation. Theoretical clarity comes from having to be explicit about how patterns of relations combine to generate a causal force. It isn't enough to say that socially similar people influence one another — you must define how relations bring people close so that interpersonal influence occurs. It isn't enough to say that people high in prestige have more power than people low in prestige — you must define how patterns of relations create prestige and power differences between people. It isn't enough to say that women have lower salaries than comparable men because women are assigned to weaker positions — you must define how relation patterns determine the strength responsible for higher salaries. Like research in more developed sciences, network analysis in social science shifts the focus of research attention from *what* to *how*; from a search for factors that matter to a search for how factors that matter have their effect.

Stronger, more cumulative research results result from having to be explicit about the manner in which relations aggregate to create causal force. This is a by-product of rejecting attributes as causal variables. Consider the diagram at the top of page 13. At the same time that causal force originates in the structure of relations, the positions of individuals in social structure is correlated with physical attributes such as age, sex, race, and so on. Some positions contain a high proportion of young, highly educated females, others contain a high proportion of old, poorly educated males. The result is a correlation between attributes and outcomes — and the possibility of predicting outcomes from social structure or attributes.

In everyday conversation and presumption, we use familiarity with the link between social structure and attributes to interpret what specific individuals are likely to do and say. The existing map of physical attributes into social structural positions provides an ethnomethodological calculus for understanding social life. This colloquial understanding is the substance of social prejudice, but more, it is where social science begins. In the early stages of work on a phenomenon, physical attributes are often used to describe variation. Attributes are easy to see and familiar to one's audience. Their ostensible effects are accordingly easy to communicate. In sociology, income differences are often correlated with age, sex, race and education to assess how much these attributes contribute to income differences. In anthropology, initial fieldwork on a society depends on attributes — sex, costume, age, wealth, etc. — to describe life in the society.

As in everyday life, when physical attributes are used as causal variables in social science, they are typically used as a surrogate for social structural data. For example, sociologists do not search for sex differences so that they can attribute the differences to biology. Where sex differences in income are detected, they are used to identify kinds of positions to which women have been assigned under prejudices accepted in a male society. But as Linton (1936) emphasized in his caution to anthropologists, people tend to forget the causal importance of relations and focus on correlates of more easily detected and communicated physical attributes known to be socially significant.

This inelegance becomes a problem when attempts are made to generalize attribute based descriptions across societies or over time. The problem is that


Socially Significant
Attributes of
Position Occupants
(e.g., sex, race, age, education, occupation)
attribute — position correlation idiosyncratic to time and place
spurious correlation between attributes and outcome
Intersecting Relations Define a Position in Social Structure
direct cause
OUTCOME

The Network Perspective

the connection between physical attributes and social structure changes over time and across societies. How frequently it changes and how much it changes is an empirical question. The point is that the connection is not causal. It is a correlation, determined when a society is observed. *Network analysis generates stronger, more cumulative research results than predictions from attributes because it grounds explanation in the bedrock relation patterns responsible for certain attributes being socially significant.* For example, sex differences in income between American men and women must have been greater at the turn of the century than now because men were then less constrained by law to employ men and women as comparable workers. Then, women could be assigned more easily to low autonomy positions regardless of their work qualifications. However, the structural conditions creating job autonomy and so higher income have their effect now as then. Research results on the structural effects generalize over time. Research results on the physical attributes associated with the process are journalistic descriptions of a situation in a society at a moment in time. They only generalize to other societies or moments in time in which the same connection occurs between attributes and social structure.

I've developed this point elsewhere (Burt, 1992a:Chap. 5), but I can summarize it here for a research audience with the analogy to stochastic regressors. Physical attributes are stochastic regressors in the sense that differences in sex, race, age, and so on are imperfect indicators of underlying structural variables directly causing outcome differences. You can estimate the effect directly by using network variables to predict outcome variables. Or, you can estimate the effect indirectly by using physical attributes to predict the outcome, in which case the estimated effect is a compound coefficient with an upper limit equal to the product of the correlation between attributes and the outcome times the correlation between attributes and social structure. As with stochastic regressors generally, the indirect estimation yields results biased toward zero — increasingly so as the connection weakens between social structure and the physical attributes selected for study.

RUNNING THE PROGRAM

To run the program, enter the following command (see README.DOC on the program disk for details on hardware requirements and installation):

```
STRUC
```

After you press a key to start the program, you'll be asked to name a command file. The default is INPUT and an example file named INPUT is included on the program disk (a simple analysis of cliques, structural equivalence and a density table). Type in a file name or press the Enter key to choose the default. Guided by the commands, the program reads data from an input file, computes, prints to an output file and writes results (if requested) to a data output file. As the program runs, it displays the commands defining the analysis and the specific task and actor being processed. When the analysis is finished, there will be a printout file, INPUT.PRT, on the default drive. Use the BROWSE utility on the program disk to take a quick look at the printout file. Enter: `BROWSE INPUT.PRT` and use the cursor keys to move through the file. Press the ESCape key to exit. Print the file as you do other text files; using a word processing program, or the PRINT command in DOS. Or, use the utility program provided with STRUCTURE to give you control over printing output files (See Chapter 6 of the ASSISTANT *Command Booklet*).

The program disk contains the program (STRUC.EXE), a utility program for looking at output (BROWSE.COM), a utility program for editing joint involvement data (JEDIT.EXE, see pages 82ff), a research assistant to help with input-output tasks (ASSIST.EXE, described in a separate *Command Booklet*), and example data files. I recommend that you put the program files in your machine's root directory or its equivalent so you can run analyses without the programs in your work area (see README.DOC on the program disk). The file EXAMPLES contains diverse analyses. Other files are named for their substance or the commands they illustrate: AUTON, CO-CITE, CONTAG, CONTAG-B to CONTAG-F, ELITES, EQUI, GROUPS, GSS, INTERLOK, MARKET, MEDICAL, MONTE, NETWORK and POWER.

Moving to your own data, a description and analysis of network structure in a study population involves up to six steps: (1) Define an input data file. The input data can be direct measures of relations, group affiliations or participation in events, binary sociometric choice data, rank order sociometric data, or the program can generate Monte Carlo data within preconfigured population structures. (2) Define a command file (see box on page 17). The example command and data files on the program disk complete these first two steps. (3) Iterate to the preferred structural image by studying the effects of alternative analysis options. (4) For the final analysis, write it's network variables to an output data file to merge with nonnetwork data. Many options are provided in the program for writing results to an output data file. (5) Use data analysis software such as SYSTAT or SPSS to study associations among the network variables, and between the network and nonnetwork variables. (6) Iterate through steps 3 to 5 as necessary.

Note the role of command files. Network analysis involves many decisions between analysis options. When your paper comes back from the journal for revision, you will have forgotten which decisions you made. The command file is your reminder memo of the steps from raw data to final results. Store it with the text of your paper. Further, the command file syntax makes it easy to try alternative analytical decisions in the context of the many decisions required in a network analysis (versus re-creating the analysis with a series of menu choices). The presumption here is that there is no one best way to measure or analyze relations across study populations. All that is best is to make it easy to compare combinations of analysis alternatives. How is analysis changed if I make relations symmetric (just change column 24 of the NETWORK command and re-run the analysis), or if I limit everyone to equal amounts of interaction by making relations row or column stochastic (just change column 32 of the NETWORK command and re-run the analysis), or if I limit indirect contact to one intermediary (just change columns 26-28 of the NETWORK command and re-run the analysis). The ideal software solution will combine a menu syntax that preserves a command file (crudely provided by ASSISTANT) with diverse analytical models and large system analysis capabilities (STRUCTURE with virtual memory). The tasks are separate at the moment to provide software solutions for the broad diversity of DOS-box research facilities.

The command file is an ASCII text file created elsewhere with a program like XTree or SideKick, a word-processor in text mode — or you can use the menu-driven ASSISTANT program on the program disk. I typically create an initial command file with ASSISTANT, then move among alternative analyses by changing options in the command file with the editor in SideKick. Problems encountered when first running the program are often input format errors. Check for consistency between commands, format statements, and input data. Running files through the ASSISTANT program is a quick, easy way to check for errors.

COMMANDS

STRUCTURE recognizes three required commands, eight optional data definition commands, and six procedure commands. Five of the procedure commands span the principles of contemporary network analysis and a sixth provides Monte Carlo studies of the others. The three required commands are DATA, NETWORK and ANALYZE. The DATA command tells the program where to find input data. The NETWORK command tells the program the size of the system under study, the type of network data input, and requests basic network indices. The ANALYZE command tells the program to begin the analysis. For example, the example command file in the shaded box requests a clique and structural equivalence analysis of a 25 person network where direct measures of relations between the people are contained on the default drive in a file named MYDATA on disk drive C (example data files are presented in the notes on the DATA command on pages 28-30, and in the Network Data section, pages 53ff). All you have to do is enter a command word such as CLIQUES or POSITIONS and the rest is handled by defaults. If you can spell it, you can do it. To modify the default analysis, use an editor or the ASSISTANT program to enter the appropriate codes following the command word.

```
DATA      C:mydata
NETWORK     25
CLIQUES
POSITIONS
ANALYZE
```

The seventeen STRUCTURE commands are listed in the following pages in alphabetical order with summary notes on each command and its parameters. Commands can appear in a command file in any order — with two exceptions; the NETWORK command must appear before an ID CODES command or any procedure command, and the ANALYZE command must be the last command defining an analysis. Once assembled for an analysis, commands are processed in the following order: data definition, MONTE, NETWORK, CLIQUES, POSITIONS, CONTAGION, AUTONOMY, and POWER.

ANALYZE

. . . defines the end of commands defining an analysis. Every analysis must end with an ANALYZE command. There are no command parameters.

For example, the following command file finds all indirect connections and prints relations in a network derived from sociometric choice data among 50 people:

```
       DATA       E:\MYPATH\MYDATA
       NETWORK     50 C                     1
       ANALYZE
```

and the following command file requests three analyses:

```
       DATA       E:\MYPATH\MYDATA
       NETWORK     30 C
       CLIQUES
       POSITIONS
       ANALYZE
       DATA       E:\MYPATH\RANK.DAT
       NETWORK     75 R
       POSITIONS
       AUTONOMY
       ANALYZE
       DATA       E:\MYPATH\NET-J
       NETWORK     85 J
       POSITIONS
       POWER
       ANALYZE
```

AUTONOMY

. . . requests an analysis of structural holes in an individual's network (see the AUTON and MARKET example command files). Here are the command parameters, where 0 marks the default options:

COLUMNS		PARAMETERS
12		**define ego** (actors whose contacts are to be studied)
	0	analyze each actor as ego
	1	first actor in system is ego
	2	last actor in system is ego
	3	select actors to be ego — IF SELECTED, then one or more lines of binary data in (80I1) format follow the AUTONOMY command to specify which actors are to be studied for their network range and structural autonomy
14		**define contacts** (actors to include in ego’s network)
	0	contacts are anyone connected with ego
	1	contacts are only the actors receiving relations from ego
16		**define extent to which actors are unique**
	0	everyone is unique
	1	read oligopoly data from input data file
18		**printout**
	0	ego results
	1	also dyad results
20		**write to output data file**
	0	no
	1	ego indices
	2	dyad indices
	3	hole signature results
	4	matrix of constraint relations (c_{ij})

AUTONOMY (continued)

For example, here is an example command file that requests a default analysis of network range and structural autonomy in a 5 person network:

```
DATA       E:AUTON.DAT
NETWORK      7 D
AUTONOMY
ANALYZE
```

and here is the same analysis with oligopoly data read from the input data file and the last individual selected for analysis:

```
DATA       E:AUTON.DAT
NETWORK      7 D
AUTONOMY   3   1
0000001
ANALYZE
```

and this is the data file AUTON.DAT for the above analyses (used in the AUTON example analysis file):

```
(7F1.0)
 100111
0 01000
00 0000
000 000
0000 01
00000 1
011100
(7F4.3)                          this line isn't read by first command file
1.00.125.1251.00.1251.00.250    this line isn't read by first command file
```

C

. . . indicates a line of user comments, to be ignored by the program. The C must appear in the first column and be followed by a blank. There are no command parameters.

COLUMNS	**PARAMETERS**
3-80	**any string of letters or numbers**

For example, this is useful for editing command files that request multiple analyses with variations on the same network concept. Here is a command file requesting various equivalence analyses:

```
DATA     E:\MYPATH\MYDATA
NETWORK    50 C
C      structural equivalence is defined by
C      raw relation patterns
POSITIONS
ANALYZE
DATA     E:\MYPATH\MYDATA
NETWORK    50 C
C      structural equivalence is defined by
C      z-score relation patterns
POSITIONS 3
ANALYZE
DATA     E:\MYPATH\MYDATA
NETWORK    50 C
C      role equivalence, defined by raw triad frequencies
POSITIONS 4
ANALYZE
```

CLIQUES

. . . requests a cohesion analysis, detecting cliques and describing the relative strength of relations (see the GROUPS and ELITES example command files). Here are the command parameters (0 marks the default options unless otherwise noted):

COLUMNS		PARAMETERS
12	**cohesion between I and J**	
	0	minimum strength relation between I and J in any network (basis for detecting strong component cliques)
	1	maximum strength relation between I and j in any network (basis for detecting weak component cliques)
	2	overlap between I's social circle across networks and J's socia circle across networks
	3	read cohesion data from input file
14	**clustering to identify cliques**	
	0,1	yes, no
16	**tree diagram** (only printed if cluster analysis is requested)	
	0,1	0 for 80 column, 1 for 132 column, typewriter display
	2,3	2 for 80 column, 3 for 132 column, graphic display
18	**print cohesion data**	
	0,1	no, yes
20	**write to output data file**	
	0	no
	1	square matrix of cohesion distances
	2	lower diagonal of matrix
	3	distances by dyad
	4	DIST.DAT file

To detect strong components, define cohesion by the minimum strength relation in any network between two people (0 in column 12). People clustered together are in reciprocal contact with one another in each network defining the system. For example, here is a file that requests strong components in a network derived from choice data among 10 people:

```
       DATA       NET-C
       NETWORK     10 C           9
       CLIQUES
       ANALYZE
```

To detect weak components, define cohesion by the maximum strength relation in any network between two people (1 in column 12). People clustered together can contact one another in at least one network, but the contact could be asymmetric and could occur in only one network. For example, here is a file that requests weak components in a network derived from choice data among 100 people (with the lower diagonal of the cohesion data matrix written to the data output file for use with a scaling program):

```
       DATA       E:\MYPATH\MYDATA2
       NETWORK    100 C          99
       CLIQUES    1       2
       ANALYZE
```

Alternatively, overlapping social circles can be used to define cohesion in weak component cliques (by putting a 2 in column 12). Overlapping social circles measure cohesion in terms of mutual social affiliations. The more affiliations two individuals have in common, the more their contact with one another is reinforced by their relations with others. Specifically, given two people I and J connected in any network, a list of everyone connected in any network to person I, and a list of everyone connected in any network to person J, the overlap in their social circles is the number of people on both lists divided by the total number of people on either list. Cohesion by overlapping social circles is set to 0 if I is excluded from J's social circle and J is excluded from I's social circle — that is to say, if there is no direct or indirect contact between I and J in any network. The range of social circles can be controlled if relations are derived from sociometric choice data. The size of the social circle around each person is determined by the maximum path distance requested. For example, a 1 in column 28 of the NETWORK command would restrict social circles to direct contacts, or a 2 would restrict social circles to direct and one step indirect contacts. To carry out clique analyses like the large scale analyses published by Kadushin and his colleagues, force choices to be symmetric and restrict social circles to one step indirect contacts (by putting a 1 in column 24 and 2 in column 28 of the NETWORK command).

CONTAGION

. . . requests a social contagion analysis to estimate interpersonal influence as a function of the equivalence or the strength of ties between actors (see the CONTAG and MEDICAL example command files). Here are the command parameters (0 marks the default options unless otherwise noted):

COLUMNS	PARAMETERS
12	**define ego** (actors to study as the object of contagion processes)
0	analyze each actor as ego
1	first actor in system is ego
2	last actor in system is ego
3	select actors to be ego — IF SELECTED, then one or more lines of binary data in (80I1) format should follow the CONTAGION command to specify which actors are to be studied as the object of contagion processes.
14	**define alters** (actors to study as the source of contagion processes)
0	alters defined by equivalence (with a POSITIONS command)
1	alters defined by cohesion (with a CLIQUES command)
2	alters defined by cohesion (for ego I by z_{ijk} across networks)
16	**define network data**
0	observed data, defined by NETWORK command
1	random data, defined by MONTE & NETWORK commands
18	**define response data**
0	read from input data file
1	read from input data file, then redistribute among actors at random
2-4	random (distribution: [2] uniform, [3] nearly normal, or [4] log normal)
20	**number of response variables** (1 - 5; blank defaults to 1)

22 **some responses missing**
0,1 no, yes

23-27 **define** ν (in F5.3 format, BE SURE TO INCLUDE A DECIMAL)
Blank defaults to an automatic search with tolerance .05
and a negative fraction requests automatic search with
different tolerance (e.g., -.2 searches with .2 tolerance,
or -.005 requests automatic search with .005 tolerance).

28 **printout**
0 summary table only
1 summary table and ego results
2 summary table, ego results, and dyad results

30 **write to output data file**
0 no
1 ego results
2 ego-alter dyad results
3 estimation results (jackknife, hetero, prom, Y, Y*)
4 square matrix of network weights (w_{ij})
5 mean network weights by response dyads

32 **print estimation details** (jackknife, hetero, prom, etc.)
0,1 no, yes

If, as is usually the case, response data are to be read from the input file (0 or 1 in column 18), then put the response data in the input data file just after the network data. The order of the input is as follows (for details, see pages 149ff in the Network Models section):

FORMAT STATEMENT (use F format to define columns containing response data)
MISSING DATA RECORD (only included if some data missing (1 in column 22))
RESPONSE DATA (with each person's responses beginning on a new row)

For example, here is an example command file that requests a default analysis of contagion by structural equivalence (on one response variable, with ν to be determined by the program) in a 5 person network derived from sociometric choice data:

CONTAGION (continued)

```
DATA       E:\MYPATH\MYDATA
NETWORK      5 C
CONTAGION
ANALYZE
```

and this is the data file, MYDATA, on path E:\MYPATH containing the initial sociometric choice data followed by each person's response on the criterion variable suspected of being generated by social contagion:

```
(5I1)
 1000
0 001
10 10
011 0
1000
(F2.1)
.9
1.
3.
4.
.5
```

The following file focuses the analysis on contagion's effect on 20 people (persons 5 through 14, and 25 through 34) within a broader network of 50 people, with ν set to 2.0 and detailed jackknife results requested to assess the estimated contagion effect:

```
DATA       CHOICES
NETWORK     50 C
CONTAGION  3 0 0 0 1 0 2.0 0 0 1
00001111111111000000000011111111110000000000000000
ANALYZE
```

IN SUM, The following table distinguishes the six classes of contagion analyses available. The usual analysis will be a class A analysis, using observed network data to predict observed response (0 in columns 16 and 18, and illustrated in the CONTAG and MEDICAL example command files on diskette). The remaining classes of contagion analyses are kinds of Monte Carlo runs for studying the properties of contagion effects and assessing the magnitude of specific observed effects.

	DEFINE RESPONSE DATA (column 18)		
DEFINE NETWORK DATA (column 16)	OBSERVED 0 in 18	SCRAMBLED 1 in 18	RANDOM 2,3,4 in 18
OBSERVED 0 in 16	A	B	C
RANDOM 1 in 16	D	E	F

In class B analyses (illustrated in the CONTAG-B example file), the observed responses are randomly mixed among the people. This has the effect of preserving the observed network structure and the observed distribution of responses across repeated trials while eliminating the contagion effect. The random mix of responses across people yields an estimated contagion effect, so repeated trials generate a sampling distribution of the contagion effect under the null hypothesis of no effect. Estimates can be written to the output data file to describe the distribution of estimates. Class B analyses are therefore useful to create a precise sampling distribution for a specific contagion effect observed in a specific network rather than relying on a theoretical sampling distribution which can be unknown or inappropriate to the network under study.

Classes C and F analyses are for studying the behavior of contagion with random response data distributed in various ways in either an observed network or a random network with fixed structural properties imposed through a Monte Carlo network design analysis. These classes of analyses are illustrated in the CONTAG-C and CONTAG-F example files.

Finally, classes D and E analyses are for studying the behavior of contagion effects with a known response distribution in a random network. These classes of analyses are illustrated in the CONTAG-D and CONTAG-E example files. Class D analyses in particular are useful because a fixed level of contagion can be set by linking the observed response data to a target image matrix and then studying how well the known contagion effect can be recovered with varying numbers of sociometric choices or varying levels of stratification within the network.

DATA

. . . defines the input data file. Every analysis must include a DATA command. The only exceptions are Monte Carlo analyses that do not require any externally defined data. There is one command parameter:

COLUMNS	**PARAMETERS**
12-80	**drive:\path\file name of data input file**

See the Network Data section of this *Manual* for detailed discussion of data files. In brief, data files contain a format statement followed by a square matrix of network data with each row of the matrix beginning on a new line (except for data on joint involvements, see page 30). Different kinds of network data require different kinds of format statements. Direct measures of relations are read in F (real) format. Sociometric choice and rank order data are read in I (integer) format. Data on joint involvement in events or affiliations are read in I and A (character) format. For example, the following command file reads direct measures of network relations from MYDATA on path MYPATH on the E drive:

```
DATA       E:\MYPATH\MYDATA
NETWORK      5 D
ANALYZE
```

and here is the MYDATA data file (each row beginning on a new line):

```
(5F6.3)
       0.431 0.671 0.000 0.000
 0.325       0.912 0.000 0.000
 0.111 0.232       0.000 0.000
 0.000 0.000 0.000       0.438
 0.000 0.427 0.000 0.875
```

The following command file reads sociometric choice data from CHOICE.DAT on the B drive:

```
DATA       B:CHOICE.DAT
NETWORK      5 C
ANALYZE
```

and here is the CHOICE.DAT data file (each row beginning on a new line):

```
(5I1)
 1100
1 100
11 00
000 1
0001
```

The following command file reads sociometric rank order data from RANK.NET on the D drive:

```
DATA       D:RANK.NET
NETWORK      5 R
ANALYZE
```

and here is the RANK.NET data file (each row beginning on a new line):

```
(5I2)
   1 2 3 4
 2   1 4 3
 1 2   3 4
 4 3 2   1
 2 4 3 1
```

And finally, the following command file reads joint involvement data from NET-J on the C drive:

```
DATA       C:NET-J
NETWORK        J
ANALYZE
```

and the next page lists the format statement and network data in the NET-J data file (with each event beginning on a new line):

DATA (continued)

```
(I2,12(1X,A4)/2X,2(1X,A4))
 3 Evel Laur Thre
 3 Evel Laur Bren
 4 Evel Thre Bren Char
 6 Evel Laur Thre Bren Char Fran
 8 Evel Laur Thre Bren Char Fran Elea Ruth
14 Evel Laur Thre Bren Fran Elea Pear Ruth Vern Myra Kath Sylv
   Hele Doro
 8 Evel Laur Thre Bren Char Elea Pear Nora
10 Laur Thre Bren Char Elea Ruth Vern Sylv Nora Hele
12 Evel Thre Pear Ruth Vern Myra Kath Sylv Nora Doro Oliv Flor
 5 Myra Kath Sylv Nora Hele
 4 Nora Hele Oliv Flor
 6 Vern Myra Kath Sylv Nora Hele
 3 Kath Sylv Nora
 3 Kath Sylv Nora
END NETWORK
```

Note that the format statement is alphanumeric with an integer defining the number of actors involved in each event (e.g., 3 people are involved in the last event; Kath, Sylv and Nora). Note also that the last entry in the list of events is "END NETWORK" to indicate the end of the events or affiliations defining the network. Events can be weighted to capture the greater importance of involvement in one event relative to others (column 20 of the NETWORK command). The diagonal element z_{jj} will be the sum of weights for events in which actor j is involved. The relationship between i and j, z_{ij}, will be the sum of weights for events in which actors i and j are both involved. In addition to weighting events by the number of actors they contain, you can define weights directly in the data file. The weight for an event is a real number and precedes the number of participants in the event. For example, here are the first three lines of the above data file with participation in the first event defined to be two and a half times more important than participation in the second:

```
(F3.1,I2,12(1X,A4)/2X,2(1X,A4))
2.5 3 Evel Laur Thre
1.0 3 Evel Laur Bren
```

More examples are given in the Network Data section of this *Manual*, beginning on page 53.

DEFAULTS

. . . redefines program defaults. If there is no DEFAULTS command in an analysis, the variables listed below are assigned their default values. Here are the command parameters:

COLUMNS	PARAMETERS
12	**random number generator cycles performed per number** (blank defaults to 2)
14-16	**maximum number of iterations to perform in computing eigenvectors or transforming a network to be row and column stochastic** (blank defaults to 50)
18-22	**maximum difference allowed between iterated values of variables at convergence** (read in F5.5 format; blank defaults to .0004)
30	**kind of display monitor** (blank defaults to program's best guess)
1	color display
2	monochrome display (color codes below are ignored)
3	COMPAQ dual display
31	**title color** (0-8; blank defaults to 2 on color displays)
32	**subtitle color** (0-8; blank defaults to 3 on color displays)
33	**background color** (0-8; blank defaults to 1 on color displays)

For example, the following command file requests greater precision in the iterative computations used to compute the eigenvector measures of prominence:

```
TITLE      More Precise Calculations
DEFAULTS     120 .0001
DATA       NETWORK.DAT
NETWORK    100 D
POWER
ANALYZE
```

DEFAULTS (continued)

and the following command file allows less precision given the small size of the network and possible difficulties in reaching convergence:

```
       TITLE      Less Precise Calculations
       DEFAULTS         .0400
       DATA       ELITES.DAT
       NETWORK     52 C
       POWER
       ANALYZE
```

and the following command file eliminates the background color (if the program is being run on a machine with a color monitor):

```
       TITLE      Default Level of Precision & No Background Color
       DEFAULTS                       230
       DATA       ELITES.DAT
       NETWORK     52 C
       POWER
       ANALYZE
```

ID CODES

. . . defines alphanumeric identification codes for the network elements. If there is no ID CODES command in an analysis, network elements are given sequential integer identification codes (1, 2, 3, ...). There are no command parameters. The command is followed by one or more lines of identification codes in 10A8 format sufficient to identify every element in the network.

For example, the following command file concerns a network among 6 people; Sam, Joan, John, William, Gert, and Zeke:

```
DATA       E:\MYPATH\MYDATA
NETWORK      6 D
ID CODES
     SAM    JOAN    JOHN WILLIAM    GERT    ZEKE
CLIQUES
ANALYZE
```

and the following command file concerns two networks among 100 people where people are identified by sequential integers in the printout and written data:

```
DATA       E:\MYPATH\CHOICES
NETWORK    100 C 2
CLIQUES
POSITIONS
ANALYZE
```

IFORMAT

. . . defines the format of integer matrices in the printed or written output. There is only one parameter:

COLUMNS	PARAMETERS
12-80	**any I format statement** (defaults to “(1X,25I4)”)

For example, the following file is included as one of the examples on diskette:

```
TITLE      Rank Order Data from Newcomb-Nordlie
DATA       NET-R
NETWORK     17 R                 1 2
ANALYZE
```

and the following is a portion of the catij path distance output showing the default (1X,25I4) format (the preliminary blank space is to avoid carriage control commands):

```
    0   2   3   2   3   3   2   1   2   1   2   3   1   1   1   1   1
    2   0   1   1   1   3   2   4   2   1   1   1   2   3   2   2   1
    2   2   0   2   3   1   2   2   2   1   3   1   2   3   1   2   1
    3   1   1   0   1   1   4   2   2   1   1   1   3   3   1   1   1
    2   1   1   1   0   2   2   3   1   1   3   2   1   2   2   2   2
```

The distances here do not require four digits, so the output format can be redefined as in the following file:

```
TITLE      Rank Order Data from Newcomb-Nordlie
IFORMAT    (1X,17I2)
DATA       NET-R
NETWORK     17 R                 1 2
ANALYZE
```

which generates the above data in the following form:

```
  0 2 3 2 3 3 2 1 2 1 2 3 1 1 1 1 1
  2 0 1 1 1 3 2 4 2 1 1 1 2 3 2 2 1
  2 2 0 2 3 1 2 2 2 1 3 1 2 3 1 2 1
  3 1 1 0 1 1 4 2 2 1 1 1 3 3 1 1 1
  2 1 1 1 0 2 2 3 1 1 3 2 1 2 2 2 2
```

Alternatively, a network of joint involvements across many events might require more than four digits to display and that could be provided by writing the data as I5 or I6 variables with the IFORMAT command.

MONTE

. . . requests a Monte Carlo network analysis (see the MONTE example command file). Here are the command parameters (0 marks the default options unless otherwise noted):

COLUMNS	PARAMETERS
12	**network data probability distribution**
0	uniform
1	nearly normal
2	log normal
13-16	**number of trials to run** (1-9999; blank defaults to 1)
17-20	**initial seed number** (integer up to four digits, blank defaults to a screen prompt)
21-22	**number of random choices per chooser** defaults to 1 for sociometric choice data (1 choice from each person) defaults to 2 for joint involvement data (2 actors chosen per event) defaults to N-1 for sociometric rank order data (each relation ranked)
24	**write choice data**
0,1	no, yes
26	**print the density table realized in each trial for the target image matrix**
0,1	no, yes
27-28	**number of positions in target image matrix**
0	no target image matrix
29-73	**number of occupants in each position** (in 15I3 format)
0-40	if all blank, then equal numbers of people are assigned to each position

The DESIGN option in the ASSISTANT program offers the easiest way to create a Monte Carlo network analysis command file. If you are writing a command file from scratch, be sure to include a target image matrix if column 28 is nonzero, each row beginning on a new line in 15I1 format. The following command file requests 99 trials (413 initial seed) of 2 uniform random choices from each of 100 people in a complex hierarchy of positions with ego-network indices for each trial:

```
NETWORK    100 C
MONTE         99 413 2     6 10 35 10  5 10 30
100100
010000
101000
000100
000011
101001
ANALYZE
```

and here is a command file, without a target image matrix, requesting 99 replicate joint involvement networks among 50 people with each person involved in up to 8 events (639 initial seed) and network power scores for each trial:

```
NETWORK     50 J
MONTE         99 639 8
POWER
ANALYZE
```

NETWORK

. . . defines the kind of network data to be analyzed (see NETWORK example command file). Every analysis must include a NETWORK command defining, at minimum, the number of actors in the system to be analyzed (or that relations are derived from joint involvements). Here are the command parameters (0 marks the default options unless otherwise noted):

COLUMNS	PARAMETERS
12-14	**number of actors in system** (2 or more; less than 2 terminates the program unless relations are being derived from joint involvements)
16	**kind of network data to be analyzed**
D	DIRECT measures (blank defaults to direct measures)
J	JOINT involvements or shared affiliations
R	sociometric RANK order data
C	sociometric CHOICE data
18	**number of networks** (1-9; blank defaults to 1 network)
20	**weighting for events defining joint involvement**
0	each event has equal weight
1	number of participants in event divided by maximum possible number of actors in system
2	1 divided by number of participants in event
3	event weights are input (not available with MONTE)
22	**nature of relations with self**
0	diagonal elements are arbitrary constants
1	diagonal elements measure strength of relation to self
24	**make relations symmetric**
0	no
1	yes, i-j relation equals stronger relation between I and J
2	yes, i-j relation equals weaker relation between I and J
3	transpose, relations between I and J reverse direction

26-28 **maximum chain of indirect contacts** (maximum path or catij distance)

default N-1 for sociometric data

30 **function transforming path or catij distances into relations**

0 frequency decay
1 fixed decay

32 **transform relations**

0 no
1 measure z_{ij} relative to the maximum in row I
2 measure z_{ij} relative to the maximum in column J
3 add a constant to all z_{ij} to eliminate any negative relations
4 make each network row stochastic
5 make each network column stochastic
6 make each network row and column stochastic

34 **printout**

0 summary table (nothing if D in column 12)
1 summary table and matrices of relations
2 suppress summary table and matrices

36 **write to output data file**

0 no
1 square matrix (or matrices) of final z_{ij}
2 square matrix (or matrices) of path distances
3 all relations by dyad

POSITIONS

. . . requests an analysis detecting structurally equivalent or role equivalent positions and describing the relative extent to which observed positions are equivalent (see the GROUPS, ELITES and ROLES example command files). Here is a partial list of the command parameters (0 marks the default options):

COLUMNS		**PARAMETERS**
12		**equivalence measure (options 0 - 6 are Euclidean distances)**
	0	Euclidean distance between raw relation patterns
	1	between raw relation patterns (mean difference)
	2	between deviation relation patterns (covariance)
	3	between z-score relation patterns (correlation)
	4	between raw triad patterns (role equivalence)
	5	between binary triad patterns (role equivalence)
	6	between weighted triad patterns (role equivalence — IF SELECTED, then the program will read weights in 36I2 format on the line following the POSITIONS command)
	7	read distance data from input file
14		**clustering to identify equivalent positions**
	0,1	yes, no
16		**tree diagram** (only printed if cluster analysis is requested)
	0,1	0 for 80 column, 1 for 132 column, typewriter display
	2,3	2 for 80 column, 3 for 132 column, graphic display
18		**spatial display**
	0	screen display
	1	pica 80 column printing
	2	condensed 132 column printing
	3	pica 132 column printing
	4	no spatial display

20		**print equivalence data (distances; also triads if 4- 6 in column 12)**
	0,1	no, yes
22		**write to output data file**
	0	no
	1	square matrix of equivalence distances
	2	lower diagonal of matrix
	3	distancs by dyad
	4	triad patterns (if 4, 5 or 6 in column 12)
	5	DIST.DAT for hypothesis tests with ASSISTANT

For example, here is a file that requests a default structural equivalence analysis of a network derived from data on an unknown number of actors jointly involved in some number of events (structural equivalence measured by differences between raw relation patterns, displayed in a screen-size spatial map, and equivalence detected with a cluster analysis):

```
DATA       NET-J
TITLE      Joint Involvement Data from Davis-Gardner
NETWORK        J
POSITIONS
ANALYZE
```

The following file requests a structural equivalence analysis of 2 networks of directly measured relations among 100 people, where structural equivalence is measured by differences between z-score relation patterns (with a printout of the Euclidean distances and an 80 column pica print of the spatial map):

```
DATA       F:\MYPATH\JOINT.DAT
NETWORK    100 D 2
POSITIONS  3     1 1
ANALYZE
```

The following file requests a role equivalence analysis of a network of directly measured relations among 100 people, where equivalence is measured by differences between binary triad patterns (with a printout of the Euclidean distances and an 80 column pica print of the spatial map):

POSITIONS (continued)

```
       TITLE    Role Equivalence based on binary triad patterns
       DATA      role.dat
       NETWORK   100 D
       POSITIONS  5    1 1
       ANALYZE
```

Binary triad patterns have been transformed from frequencies to binary data, 1 for each triad type in which an individual is involved and 0 for each type in which he is not involved. The following file requests a role equivalence analysis of a network of directly measured relations among 100 people, where equivalence is measured by differences between weighted triad patterns (again, with a printout of the Euclidean distances and an 80 column pica print of the spatial map):

```
TITLE    Role Equivalence based on weighted triad patterns
DATA      role.dat
NETWORK   100 D
POSITIONS  6    1 1
 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
ANALYZE
```

where the weighting eliminates the null triad (triad 1), giving equal weight to each other triad type. Weights are computed by dividing each weight by the largest in the weight vector. In this case, all nonzero weights are the same magnitude, so all triad types but the null triad will be given equal weight. Here is the same command file with a different weight vector:

```
TITLE    Role Equivalence based on weighted triad patterns
DATA      role.dat
NETWORK   100 D
POSITIONS  6    1 1
 0 0 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ANALYZE
```

In this analysis, role equivalence will be defined by the extent to which individuals are similarly involved in the six triads where ego is a broker between alters (triads 3, 5, 7, 8, 9 and 10). In other words, individuals will be equivalent to the extent that they fulfill similar broker roles. The sequential identification of triad types is printed with the distance data for any run requesting a role equivalence analysis.

The POSITIONS command also requests density tables summarizing the structure of relations among two or more positions defined across networks (see the last analyses in the GROUPS example command file). Here are the remaining POSITIONS command parameters (0 marks default unless otherwise noted):

COLUMNS	PARAMETERS
27-28	**number of positions in density table** (1 - 15)
29-31	**number of actors occupying position 1** (1 - 99 occupants)
32-34	**number of actors occupying position 2** (1 - 99 occupants)
35-37	**number of actors occupying position 3** (1 - 99 occupants)
38-73	**number occupying positions 4 up to 15** (in I3 format with 1 - 99 occupants per position)

The occupants of each position are identified by their sequential position in the network system (in 20I4 format, 1 for first, 34 for thirty–fourth, etc) with the occupants of each position beginning on a new line. For example, the following command file requests a two position density table defined across two networks where four people occupy the first position and four occupy the second position (and the remaining two people are residuals):

```
DATA       TABLE.DAT
TITLE      Example Density Table
NETWORK     10 D 2
POSITIONS                  2  4  4
   1   2   3   4
   7   8   9  10
ANALYZE
```

POSITIONS (continued)

and the following analysis, taken from the GROUPS example file, requests a four position density table:

```
TITLE      Subgroups Among Elite Methodologists
DATA       ELITEGRP.DAT
PRINT      ELITE.PRT
NETWORK     25 C          12     1 1
POSITIONS                  4  4  6  6  9
   1   2   3   4
   5   6   7   8   9  10
  11  12  13  14  15  16
  17  18  19  20  21  22  23  24  25
ANALYZE
```

POWER

. . requests an analysis of network prominence and equilibrium (see the POWER and EQUI example command files). Prominence measures and network stability results are computed under independence or with substitutions between actors. Here are the command parameters, where 0 marks the default options:

COLUMNS	PARAMETERS
12	**network equilibrium**
0	independence
1	allow substitutions
13-17	**define *v* for substitutions** (in F5.3 format, BE SURE TO INCLUDE A DECIMAL) Blank defaults to an automatic search with tolerance .05 and a negative fraction requests an automatic search with different tolerance (see page 25).
18-20	**number of alternative *v*** (I3 format) Program computes equilibrium relations for integer values of *v* from 1 to the integer entered here, and prints a summary of change in stability correlations across the values of *v*.
22	**printout**
0	prominence indices and stability results
1	also equilibrium relations
24	**write to output data file**
0	no
1	prominence indices
2	square matrix of equilibrium relations
3	equilibrium relations by dyad

POWER (continued)

For example, here is a command file that requests a default analysis of power and stability:

```
DATA       E:NETWORK
NETWORK     50 D                 1 1
POWER
ANALYZE
```

Here the analysis is extended to allow actors to substitute for one another in the search for equilibrium (ν is left blank for automatic search), and have relations written by dyad to the output data file to study the association between observed and equilibrium relations:

```
DATA       E:NETWORK
WRITE      E:POWER.CMD
NETWORK     50 D
POWER      1     000 0 3
ANALYZE
```

and here the analysis consists of looking at the correlation between observed and equilibrium relations across 20 values of ν as the substitution weights change from interdependence toward independence:

```
DATA       E:NETWORK
NETWORK     50 D
POWER      1     020 0 0
ANALYZE
```

PRINT

. . . defines the printout file. There is only one command parameter.

COLUMNS	PARAMETERS
12-80	**drive:\path\file name of printout file** (defaults to the name of the input data file with a .PRT extension added)

For example, the following command file prints to file PRINTOUT on path MYPATH on the E drive:

```
DATA       E:\MYPATH\MYDATA
PRINT      E:\MYPATH\PRINTOUT
NETWORK      5 D
ANALYZE
```

and the following command file prints to file MYDATA.PRT on path MYPATH on the E drive:

```
DATA       E:\MYPATH\MYDATA
NETWORK      5 D
ANALYZE
```

You can send the output directly to your printer by adding a print command to your command file for each analysis defining LPT1 as the printout file (no colon after LPT1). Replace LPT1 with the correct port if your printer isn't connected to the #1 line printer port.

RENDER

. . . requests an attribute rendering to describe the location of attributes in social structure (see the RENDER and AGE example command files). Here are the command parameters (0 marks the default options unless otherwise noted):

COLUMNS		PARAMETERS
12		**define networks behind attributes**
	0	defined by NETWORK command
	1	asymmetric joint involvements — IF SELECTED, then joint involvement data are read as sociometric choices in survey network data (first attribute is ego, following attributes are alters). Use the ID CODES command to define the order of attributes in the network.
13-17		**define** *v* (in F5.3 format, BE SURE TO INCLUDE A DECIMAL) Blank defaults to an automatic search with tolerance .05 and a negative fraction requests automatic search with different tolerance (e.g., -.2 searches with .2 tolerance, or -.005 requests automatic search with .005 tolerance).
18		**topological map**
	0	screen display
	1	pica 80 column printing
	2	condensed 132 column printing
	3	pica 132 columns printing
	4	no map
20		**analyze internal consistency**
	0	no, just build Blau space
	1	yes, box-plot and variance results
	2	also cluster analysis within attribute categories (80 column)
	3	also cluster analysis within attribute categories (132 column)
22		**write to output data file**
	0	no
	1	Blau space X-Y-Z coordinates
	2	equivalence tie box-plot parameters
	3	DIST.DAT matrix of equivalence densities (for UCINET)
	4	equivalence densities (M x M+1; last column is density of equivalence ties to excluded individuals)

The analysis requires a variable assigning individuals to attribute categories. The attribute variable is read from the data file after network or distance data. For example, here is the BLAU2.DAT file:

```
(I1,9A2)
8T T G1G2G3B1B2B3
5B1T B2B3B1
5B2T B1B3B2
5B3T B1B2B3
5G1T G2G3G1
5G2T G1G3G2
5G3T G1G2G3
END
(A7)
TEACHER
BOY
BOY
BOY
GIRL
GIRL
GIRL
```

which is read by the following command file (second analysis in the example file RENDER):

```
DATA       BLAU2.DAT
TITLE      Hypothetical Class on Last Day
NETWORK    007 J     1           2
ID CODES
      T       B1      B2      B3       G1      G2      G3
RENDER     1 1.0 0 2 0
ANALYZE
```

Note four things: (1) The attribute variable is input in the same manner as response data for a contagion analysis; a format statement is followed by a record for each individual, each beginning on a new line. (2) Each individual's attribute is read as an alphanumeric label like an ID code (also limited to 8 characters). Enter a blank label for any individual to be excluded from the attribute categories under study. (3) Joint involvement data here are read as survey network data (1 in column 12 of RENDER command, e.g., the first row of data in BLAU2.DAT shows that the teacher cited each of the children in the class). (4) The order of the attributes in the network is defined by listing attributes in the desired order on the ID CODES command in the command file.

RFORMAT

. . . defines format of real number matrices in the printed or written output. There is only one parameter:

COLUMNS	PARAMETERS
12-80	**any F format statement** (defaults to “(1X,15F8.3)”)

For example, the following file is included as one of the examples on diskette:

```
TITLE      Structurally Equivalent Methodology Experts
DATA       ELITES.DAT
NETWORK     52 C          13
POSITIONS
ANALYZE
```

and this is a portion of the distance output showing the default (1X,15F8.3) format (the preliminary blank space is to avoid carriage control commands):

```
    4.109
    3.165   2.658
    3.434   4.115   4.259
    3.912   1.807   3.051   3.175
    2.837   2.966   2.997   3.512   3.206
```

If this level of information is more than necessary (e.g., to shorten the printout file), the output format can be redefined as in the following file:

```
TITLE      Structurally Equivalent Methodology Experts
DATA       ELITES.DAT
RFORMAT    (1X,30F3.0)
NETWORK     52 C          13
POSITIONS
ANALYZE
```

which generates the above data in the following form:

```
 4.
 3. 3.
 3. 4. 4.
 4. 2. 3. 3.
 3. 3. 3. 4. 3.
```

Alternatively, the distance data could be preserved with greater precision by redefining the output format as in the following file:

```
TITLE      Structurally Equivalent Methodology Experts
DATA       ELITES.DAT
RFORMAT    (1X,10F10.6)
NETWORK     52 C          13
POSITIONS
ANALYZE
```

which generates the distance data in the following form:

```
  4.109089
  3.164973  2.658168
  3.434168  4.114684  4.258572
  3.911698  1.806730  3.051057  3.175215
  2.837340  2.965625  2.996818  3.511561  3.206498
```

TITLE

. . . defines a title to appear in the printout. If there is no TITLE command in an analysis, the phrase "No Title" appears in the printed and written output where the title would otherwise appear. There is only one command parameter:

COLUMNS	**PARAMETERS**
12-80	**any string of letters or numbers** (defaults to "No Title")

For example, the title `Example Analysis with Basic Sociometric Indices` will appear on printout from the following command file:

```
DATA       E:\MYPATH\NETWORK
TITLE      Example Analysis with Basic Sociometric Indices
NETWORK     85 C
ANALYZE
```

WRITE

. . . defines the data output file. There is only one command parameter:

COLUMNS	PARAMETERS
12-80	**drive:\path\file name of data output file** (defaults to the name of the input data file with a .WRT extension added)

For example, the following command file writes power scores to file INDICES on path MYPATH on the E drive:

```
DATA     E:\MYPATH\MYDATA
WRITE    E:\MYPATH\INDICES
NETWORK   100 D
POWER     0 000 0 1
ANALYZE
```

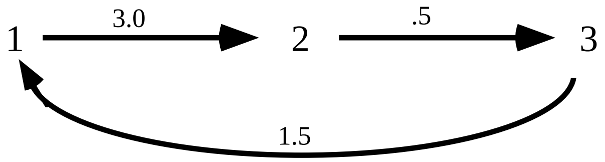
and the following command file writes them to file MYDATA.WRT on path MYPATH on the E drive:

```
DATA     E:\MYPATH\MYDATA
NETWORK   100 D
POWER     0 000 0 1
ANALYZE
```

NETWORK DATA

Much of the variation in quality between network analyses lies in the wit and insight that different individuals bring to the task of imagining how data can be defined to capture the networks that structure a system of actors. Network data are the critical epistemic link connecting reality with the abstract network models that describe social structure and predict feelings and behaviors. In addition to methodological skills in sampling and measurement — skills needed in any social science effort to obtain data on representative attributes, feelings and behaviors — the network analyst needs the skills to capture interdependence between analytical units. Marsden (1990) reviews issues in measuring network relations and Coleman's (1958) discussion of research strategies remains a key reference, both for what it says as well as the kind of methodological problems highlighted by the selection of solutions.

Network analysis starts with one or more square tables of variables where each table represents a network and each variable represents a relation within a network. For example, here is a diagram, often called a sociogram, of a network among three actors:



where arrows represent relations from one actor to another. You can see that actor 1 directs a relation of strength 3.0 to actor 2, who directs a relation of strength .5 to actor 3, who directs a relation of strength 1.5 to actor 1. The actors here could be animals, people, groups, organizations, geographic regions, traffic lights, or any object capable of being linked by a relationship to other objects. For analysis, it is convenient to represent the network data in a sociogram as a table where rows and columns correspond to actors. Under usual conventions, the above sociogram is represented by the following table:

$$\begin{array}{ccc} 0.0 & 3.0 & 0.0 \\ 0.0 & 0.0 & 0.5 \\ 1.5 & 0.0 & 0.0 \end{array}$$

where the first row contains relationships from actor 1 (nothing to himself, a 3.0 strength relation to actor 2 and no relation to actor 3), the second row contains relations from actor 2 (no relation to actor 1, no relation to herself and a .5 strength relation to actor 3), and the third row contains relations from actor 3 (a 1.5 strength relation to actor 1, no relation to actor 2 and no relation to himself). For methodological and theoretical discussions, it is convenient to represent the table more generally as a matrix of variables where variable z_{ij} measures the strength of the relation from actor i (row i) to actor j (column j):

$$\begin{array}{ccc} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{array}$$

For example, z_{23} equals .5 in the above table, measuring the relation from actor 2 to actor 3. More general still, the table can be discussed as an (N by N) matrix Z containing the N^2 variables measuring relations among N actors in a network.

Three presumptions should be noted. First, every variable is presumed to be measured. There are no randomly missing z_{ij}. However, two kinds of systematic missing data are common. Self-relations (the z_{ii}, diagonal elements in the Z matrix) are often excluded for want of comparable information on relations between and within actors. Further, where full information is not available on relations in both directions between actors (z_{ij} and z_{ji}), symmetry is imposed on whatever relationship data can be obtained (by making z_{ij} and

z_{ji} equal by definition), which means that only the variables in the lower or upper half of the matrix need to be available (3 relations in the above example; z_{21}, z_{31} and z_{32}, or N(N-1)/2 relations in general).

Second, each variable is presumed to measure the intensity, or strength of a relationship. Categorical measures are excluded. For example, you can't define z_{ij} to be 1 if actors i and j have a friendship relation, 2 if they have a kinship relation and 3 if they are co-workers. Categorical measures are handled by creating separate networks for separate kinds of relations.

Third, each variable is presumed to be comparable to the others. They should be comparable in metric. For example, the 3.0 relation from actor 1 to actor 2 in the above table is presumed to represent a relationship that is twice as strong as the 1.5 relationship from actor 3 to actor 1. Relations within the same network should also be comparable in content. If the z_{12} relation of 3.0 measures hours of advice that actor 1 has solicited from actor 2, then the strength of every other relation in the network should be measure hours of advice each actor sought from another (as opposed to dollars of economic exchange, or hours of socializing).

In sum, the data for a network analysis consist of one or more tables of relation variables defining:

- a system of N actors,
- K networks of relations among the N actors,
- where the variables measuring relations in each network are comparable in metric and content, and
- each relation variable z_{ij} measures the form (the strength or intensity) of a relation from actor i to actor j.

You have several options for doing this in STRUCTURE. The simplest is to read direct measures of z_{ij} from the input data file (with the option of later transforming the raw data to make relations more comparable or interpretable). If direct measures are not available, you can derive relations from binary sociometric choice data, rank order sociometric data, or joint involvement data.

Three STRUCTURE commands define the network data for an analysis. First, the required DATA command defines the input file containing the raw data on relations or from which relations will be derived. Second, the required NETWORK command defines the number K of networks to be defined, the number N of actors connected by relations in the networks, and the kind of network data to be analyzed (D for direct measures, C for binary sociometric choice, R for rank order sociometric data, and J for joint involvement data). Third, the optional ID CODES command assigns eight-character alphanumeric identification codes to each actor. I'll discuss the alternative kinds of network data in turn, illustrating the DATA and NETWORK commands as I go.

DIRECT MEASURES OF RELATIONS

Occasionally, you can get direct measures of relations. Excellent network data are available, for example, from input-output tables describing regional and national economies. An example is included for the American economy on the program disk. The file MARKET.DAT, named in example command file MARKET, contains an input-output table measuring the flow of dollars in 1972 between 77 sectors of the American economy. Relation z_{ij} is the millions of dollars of goods that establishments in sector i sold in 1972 to establishments in sector j.

The data are read into STRUCTURE as a square matrix of real number variables for each network. In the command file, put a D in column 16 of the

NETWORK command to indicate direct measures (or you can leave it blank; D is the default), put the number of actors in columns 12-14, and the number of networks in column 18 (or leave it blank if you have one network; 1 is the default). The first line of the input data file should contain a real number (e.g., F or E) format statement for reading each row of network data, followed by the network data for each network (all rows in network 1 followed by all rows in network 2, and so on). If you look at the MARKET.DAT example data file, you'll see that the first line is a format statement and that each row of the network begins on a new line. If you are not familiar with format statements, see the Appendix to this *Manual*, or Appendix D in the ASSISTANT *Command Booklet*. As a simple, immediate illustration, here is the input data file for the three person network on pages 53-54:

```
(3F6.1)
   0.0   3.0   0.0
   0.0   0.0   0.5
   1.5   0.0   0.0
```

which begins with a format statement defining three 6-column variables per row and each row of the network begins on a new line. The format in MARKET.DAT is slightly more complex because of its larger size.

Direct measures of relations are often obtained in the form of on-off data. For example, the sociogram at the top of the preceding page describes a simple bureaucracy where persons 1, 2, 3 and 4 report to person 5, who reports to a committee composed of persons 7, 8, 9 and 10. Person 6 is outside the chain of command. Here is the input data file for the sociogram (where "report" relation z_{ij} equals 1 if person i reports to person j and 0 otherwise):

```
(10F1.0)
0000100000
0000100000
0000100000
0000100000
0000001111
0000000000
0000000111
0000001011
0000001101
0000001110
```

And here is a command file to read the data from a file named INPUT.DAT:

```
DATA       INPUT.DAT
NETWORK    010
ANALYZE
```

Data Transformations

After the z_{ij} are read, they can be transformed. The transformed z_{ij} will then be treated as the direct measures of relations to be analyzed. The parameters in columns 22, 24 and 32 of the NETWORK command define transformations (see pages 38-39). First, you can keep or ignore diagonal elements of the network data — the z_{ii} that measure self relations. The default is to ignore them (0 or blank in column 22). A 1 in column 22 asks that z_{ii} self relations be treated like the z_{ij} relations between actors.

Second, you can keep or ignore asymmetries in the network data. This is controlled by column 24 of the NETWORK command. This parameter will be set by the program for some clique requests. The default is to retain whatever asymmetry exists in the network data. A 1 in column 24 sets z_{ij} and z_{ji} within a network to the value of whichever is stronger. This creates a relationship between two actors whenever there is any contact between them. A 2 in column 24 sets z_{ij} and z_{ji} within a network to whichever is weaker. This eliminates any relationship between two actors unless they both direct a relation to the other. A 3 in column 24 reverses z_{ij} and z_{ji} within each network. A person who was the object extensive relations before the transformation will be the source of extensive relations after the transformation.

Third, you can keep the existing metric or impose a new one on the network data. This is controlled by column 32 of the NETWORK command. If you impose symmetry on the data and request one of these metric transformations, symmetry is imposed before the metric transforma-tion is performed. A 1 or 2 in column 32 converts the network data to row or column marginal data. Each element is measured relative to the largest in the row or column:

$$z_{ij} = z_{ij}/(\text{maximum } z_{ij} \text{ in row i}),$$

or,

$$z_{ij} = z_{ij}/(\text{maximum } z_{ij} \text{ in column j}).$$

This can be a useful transformation for improving comparisons across respondents (e.g., relations defined by rating data often vary in mean and variance between respondents), or highlighting pattern in networks of low magnitude relations (e.g., Burt and Carlton, 1989). A 3 in column 32 eliminates negative data in a network by setting the weakest relation to zero:

$$z_{ij} = z_{ij} - (\text{minimum } z_{ij} \text{ in network}),$$

This is useful if power or equilibrium results will be extracted from the data. This elimination of negative relations is performed automatically before carrying out any of the remaining three transformations. A 4 or 5 or 6 in column 32 respectively converts the network data to row stochastic,

$$z_{ij} = z_{ij}/(\Sigma_j \, z_{ij}),$$

column stochastic,

$$z_{ij} = z_{ij}/(\Sigma_i \, z_{ij}),$$

or both,

$$z_{ij} = z_{ij}/(\Sigma_j \, z_{ij}) = z_{ij}/(\Sigma_i \, z_{ij}),$$

where summation is within each network. These transformations are useful for highlighting flows through networks, and eliminating differences between actors in their aggregate tendencies to respectively send, receive, or send and receive relations. For example, input coefficients in input-output analysis correspond to column stochastic network data. The transformation to simultaneous row and column stochastic data is iterative. A limit of 50 iterations will be run to carry out the transformation (or whatever limit is defined in columns 14-16 of a DEFAULTS command, see page 31). The transformation is complete when the z_{ij} in each row and column of each network sum to within .0004 of 1.0 (or whatever tolerance is defined in columns 18-22 of a DEFAULTS command to replace the .0004 default).

SOCIOMETRIC CHOICE DATA

STRUCTURE accepts binary choice data or rank order choice data. I'll focus on binary choice data since it is the more typical. Lindzey and Byrne (1968) provide a thorough review of data collection issues (Burt, 1982:22-29, for quick review with selected references). In their most familiar form, sociometric choice data are responses to name generating questions: "Who do you ___?" For example, who are your best friends? From whom do you most often get advice? With whom do you most often socialize? A respondent is expected to cite some number of people with whom he has especially strong relations of the kind requested by the name generator. In selecting secondary sociometric data or obtaining your own, it is worth noting Lindzey and Byrne's five data collection guidelines, here paraphrased for brevity: (1) System boundaries should be made clear to respondents. For example, it should be difficult for one respondent to be citing people throughout an organization while another respondent believes that he is supposed to limit his citations to immediate co-workers. (2) "Under many circumstances it is preferable to specify the number of choices to be made." An upper limit of 3 to 5 names is typically used to keep the response task simple. (3) The kind of relation being requested should be clear to respondents. For example, avoid ambiguous name generators that would allow one respondent to believe you want her to name friends while another respondent believes you want him to name the people he meets most frequently. (4) Citations should be confidential. Respondents making citations in a group setting, or aware that their citations will be known to others in their group, feel pressure to mention people in the group. (5) Questions should be gauged to the respondent's level of understanding. Rural, working class respondents should not be asked to name the individuals quintessential to their career mobility (as if anyone should be presented with such a supercilious phrase). Once obtained, the choices define a matrix of binary variables a_{ij} for each name generator — a_{ij} equal to 0, unless i cites j whereupon a_{ij} equals 1.

The data are read into STRUCTURE as a square matrix of integer variables for each network. Each row of each network begins on a new line. In the command file, a C in column 16 of the NETWORK command indicates choice data. Put the number of actors in columns 12-14, and the number of networks in column 18 (or leave it blank if you have one network; 1 is the default). Almost always, the diagonal elements are arbitrary constants. Respondents are not asked to cite themselves (column 22 of the NETWORK command is left blank). For example, here is an input data file of sociometric choices corresponding to the sociogram on page 57:

```
(10I1)
0000100000
0000100000
0000100000
0000100000
0000001111
0000000000
0000000111
0000001011
0000001101
0000001110
```

and the following command file reads the data from an input data file named NET-C:

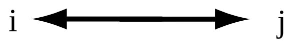
```
DATA       NET-C
IFORMAT    (1X,10I5)
RFORMAT    (1X,10F5.2)
NETWORK    010 C
ANALYZE
```

Notice that these look almost like the data and command files on page 58 in which the binary data are treated as direct measures of relations. The difference in the data file is the use of integer (I) format for sociometric choice data. See the Appendix to this *Manual*, or Appendix D in the ASSISTANT Command Booklet, if you are not familiar with format statements. The essential difference in the command file is the C in column 16 of the NETWORK command. Nonessential differences in the command file are the optional output format commands I have included to condense the printout (IFORMAT for integer output, RFORMAT for real number output). If you have multiple networks of choice data, simply add the second network below the first in the input data file, then the third, and so on, and put the number of input networks in column 18 of the NETWORK command (see the example data file NET-C and example command file INPUT on the program disk).

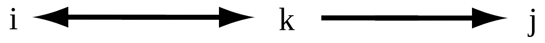
Measuring Relations

After the a_{ij} are read, they are used to derive the z_{ij} measures of relation strength. The task of deriving relations from sociometric choices is one of changing a_{ij} equal to 0 into nonzero z_{ij} where appropriate.

This is accomplished by locating indirect connections and measuring z_{ij} as a function of the length of indirect connection from i to j. The longer the indirect connection, the weaker the relation. Indirect connections are detected with path distances. The path distance from actor i to actor j, pd_{ij}, is the minimum number of choices i requires to reach j. For example, here is a sociogram in which i cites j directly so pd_{ij} equals 1:

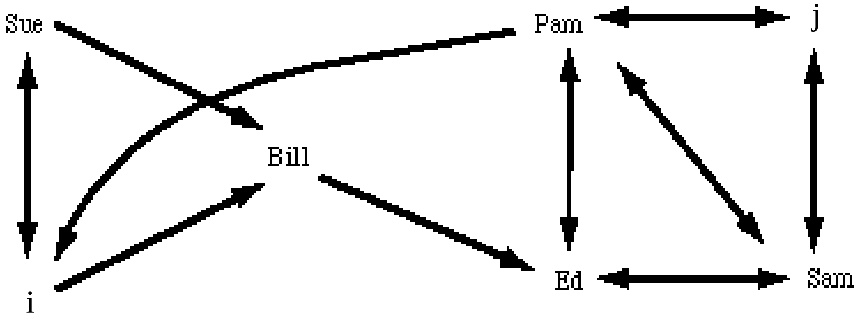


Here is a sociogram in which i doesn't cite j, but his citation to k and k's citation to j creates a two-step indirect connection from i to j so pd_{ij} equals 2:



In the sociogram at the top of the next page, i and j are on opposite sides of the network. There are two chains, each of four citations, through which i can reach j so pd_{ij} equals 4. Path distances need not be symmetric. In the sociogram, path distance from i to j is 4, but the path distance from j to i is only 2 (through j's citation to Pam and her citation to i).

STRUCTURE finds path distances by raising the input matrix of choice data to successive powers (e.g., Burt, 1982:26-27). You control the search for indirect connections with a three-digit number in columns 26-28 of the



NETWORK command. If sociometric data are input (i.e., there is a C or R in column 16 of the NETWORK command), the path distance parameter defaults to a search for all possible indirect connections in each network. If you set the parameter to a nonzero value, the search is limited to path distances equal to, or shorter than, the specified value. For example, if you put a value of 002 in columns 26-28, nonzero relations ($z_{ij} > 0$) will only occur where there is a direct sociometric citation ($pd_{ij} = 1$) or an indirect connection through one intermediary ($pd_{ij} = 2$). You have three options with STRUCTURE for deriving the z_{ij}.

Sociometric Choice as a Direct Measure

The simplest is to ignore all indirect connections, treating the original binary choice data as direct measures of relations. To treat the data in this way, put a 1 in column 28 of the NETWORK command, forcing the program to limit relations to direct choices (one-step path distances).

This treatment of binary choices as direct measures of network relations is common in analyses using CONCOR to construct blockmodels, though operationalizing structural equivalence with CONCOR is in no way limited to binary network data (e.g., White et al., 1976:750, 759; Schwartz, 1977;

Arabie and Boorman, 1982). This idea is sometimes taken to the extreme of forcing quantitative measures of relationship into necessarily arbitrary categories of present and absent ties (e.g., Snyder and Kick, 1979). However where categorical choice data have been obtained, using raw choice data as direct measures of network relations has the virtue of preserving the raw data obtained from respondents. The respondent named three people and those three people will be presumed to be the only individuals to whom the respondent sends relations. Even granting the merit of such an argument, it does not extend to cover binary choice data created in ways never asserted by the respondents providing the original sociometric data. For example, White et al. (1976:759) recode the Newcomb–Nordlie sociometric rank order data into two networks of binary relations; a network of liking where each person's top two rankings are assumed to be sociometric citations of attraction, and a network of antagonism relations where each respondent's bottom two ranks are assumed to be antagonism citations. In the same analytical style, Breiger (1976) recodes the data obtained with a seven point scale of professional contact into multiple networks of binary contact such as mutual contact, asymmetric awareness, and symmetric unawareness. These artful recodings are advanced on the argument that structural equivalence is better revealed with networks of contrasting relations (attraction versus antagonism, etc.). This argument, nicely summarized by Arabie and Boorman (1982), is a significant insight and compels emulation. Note however that the argument is in no way compromised by including indirect connections in the search for structural equivalence across networks.

Moreover, there are advantages to including indirect links. First, it seems more realistic. Choice data are often arbitrarily limited to the first three to five people cited. Indirect connections such as friends of friends indicate less close contacts likely to have been cited if the study field work had recorded more sociometric choices. The decision to treat choice data as direct measures of relations is less a decision to treat cited relations as strong than it is a decision to treated uncited relations as trivial. But the lack of a citation from person i to person j needn't indicate the lack of a relationship from i to j. It merely indicates that i doesn't see j as one of the people with whom he has the strongest of the requested kind of relationship. For example, j might be a "good friend" rather than "best friend" so respondent i doesn't cite j in response to a sociometric question asking for the names of i's best friends.

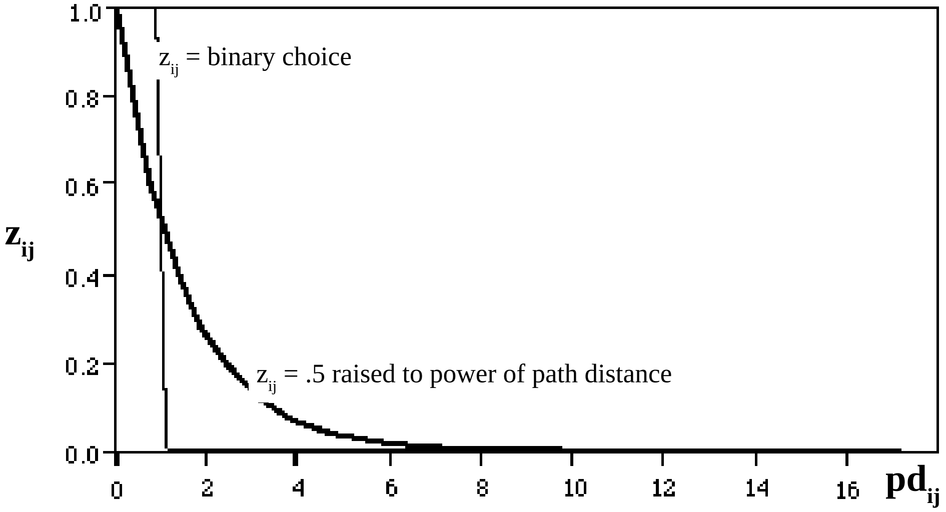
Second, including indirect connections can improve equivalence measures. Networks of direct choices tend to be sparse, creating computational difficulties for correlation models of structural equivalence, and lessening the reliability of structural equivalence measures more generally. Role equivalence measures tend to be dominated by the null triad. Including indirect connections in the network data used to define structural equivalence fills empty space in the initial choice matrix. Further, indirect connections can improve the precision of structural equivalence measures. Individuals structurally equivalent in a matrix of path distances are equivalent with respect to their direct as well as indirect relationships.

Fixed Decay Function

To include indirect connections, it is simple to measure relation strength as a constant function of path distance (1 in column 30 of the NETWORK command). Katz (1953) was first to propose the transformation, suggesting that the strength of a network relation be measured as a faction raised to the power of its corresponding path distance:

$$z_{ij} = \left\{ \begin{array}{c} 1,\ \text{if i = j} \\ a^{(pd_{ij})},\ \text{if i can reach j};\ \ 1 \leq pd_{ij} \leq \text{N-1} \\ 0,\ \text{if i cannot reach j} \end{array} \right\},$$

where the fraction *a* in Katz’s numerical illustration was .5 and it continues to be set at .5, although Katz used a more sophisticated definition of the



constant in which 1/*a* is to be larger than the maximum eigenvalue of the choice matrix. The fraction *a* is set to .5 in STRUCTURE. The key point is that relations corresponding to long path distances are much weaker than relations corresponding to short ones, z_{ij} decreasing from .5 for direct choices, to .25 for two–step path distances, to .125 for three–step path distances, and so on.

This transformation of path distances into network relations is described by the solid line in the above graph — in contrast to the dotted line for choices treated as direct measures of relations. Path distance is given on the horizontal axis and corresponding z_{ij} are given on the vertical axis. Note the sharp decay in relationship strength. Relations between individuals connected by path distances of five or more choices ($z_{ij} = .5^5 = .016$) are nearly the same strength as relations between individuals completely unconnected with one another. The rate of decay with increasing path distance can be slowed by increasing the fraction *a*, but there is evidence to support the idea that relation strength should decrease quickly beyond indirect connections through more than one intermediary (Friedkin, 1984).

There are two virtues to such a function. The most important is that it includes information on indirect connections in the measure of relation strength. A lesser virtue is that it preserves the path distance information obtained from the original choice data. You can see immediately from the strength of a relation how many intermediaries established the relation (.5 indicates a direct path, .25 indicates a two–step path, and so on).

At the same time, path distances of the same length can mean different things in different circumstances. One drawback to a fixed decay function is the presumption that the number of sociometric choices elicited from each respondent elicits the same strength relations. This is a problem in its own right, and a problem compounded by the practice of eliciting a fixed number of choices from each respondent (see Holland and Leinhardt, 1973; Hallinan, 1974). Consider two respondents, one in a group of four close friends and the other in a group of six close friends. If each is asked to name their three best friends, the first respondent has no problem because the number of choices equals the boundary of his social circle of close friends. Two–step path distances will represent relations beyond his social circle. The second respondent can only name three of her five close friends and some two–step path distances from her will refer to her relations with uncited close friends through cited friends to friends she was unable to cite.

A second drawback to a fixed decay function is the presumption that relation strength decays with increasing path distance at the same rate for all people. For example, if a respondent names three best friends who in turn each name three and none of the choices go to the same people, then the respondent has nine relations corresponding to two–step path distances. If another respondent names only one best friend who in turn names only one, then the respondent has one two–step path distance relation within a small social circle of three people including himself. On the presumption that relations require energy to sustain, the second respondent's two–step path distance, reaching fewer people within a smaller social circle, indicates a stronger relationship than the many relations corresponding to two–step path distances from the first respondent.

Frequency Decay Function

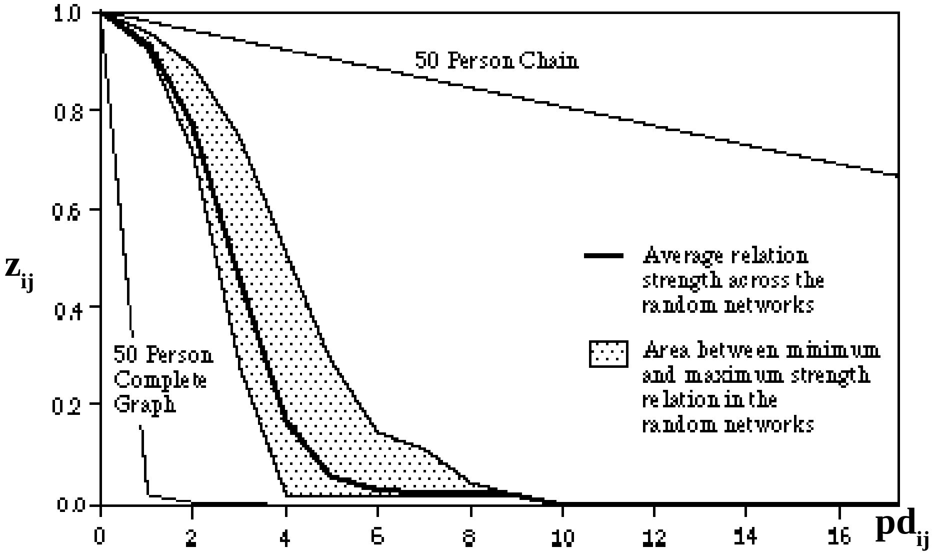
An alternative is to measure relation strength with a variable rate decay function of path distance where the rate of decay depends on the structural circumstances in which a path distance occurs. The following decay function was originally proposed to prepare choice data for a structural equivalence analysis (Burt, 1976:118–119; 1982:28–29):

$$z_{ij} = \left(\begin{array}{c} 1, \text{ if i = j} \\ 1 - \frac{f_{ij}}{N_i}, \text{ if i can reach j; } 1 \leq pd_{ij} \leq N\text{-}1 \\ 0, \text{ if i cannot reach j} \end{array} \right),$$

where N_i is the number of individuals that i can reach including herself in any number of choices and f_{ij} is the number of individuals that i can reach in the minimum number of choices needed to connect her with j. This decay function is the default in STRUCTURE (blank column 30 in the NETWORK command), so it is probably more widely used than it is known. The argument for this transformation (or any other variable rate decay function of path distances) is that the rate at which the strength of a relation decreases with the increasing length of its corresponding path distance should vary with the social structure in which it occurs. Here, decay is a function of the number of people reached at each path distance compared with the total number reached at the boundary of an individual's social circle. The larger the group over which you have to distribute network time and energy, the weaker the relationship you can sustain with any one member of the group, and the stronger the relations with the people connected most directly to you.

The figure at the top of the next page illustrates variation in relation strength around a given path distance in a network of 50 individuals. The extremes possible are indicated by dashed lines. If the network were a chain with the first person citing the second who cites the third and so on to the forty–ninth person citing person fifty, then the strength of relations for the first person would look like the dashed line at the top of the figure. Very few people are reached at each path distance, so it would be easy to sustain strong relations with individuals of path distance two, three, or four steps away and these people are dramatically closer than the 49–step outer boundary of the social circle around the first person. In contrast, if the network were a complete graph with every person making the 49 citations required to connect him or her directly to every other person, then the strength of direct ties would be very weak given the difficulty of sustaining so many relationships at the same strength and the fact that direct ties define the outer boundary of each person's social circle. This is illustrated by the lowest dashed line in the graph.

Between these extremes are results more likely in empirical research. The results describe 100 random networks of 50 people in which each person made 3 sociometric choices equally likely to go to each other person. The random choice data were generated with the MONTE command and a 639 seed number to begin each series of replicate networks (see pages 36-37). No path distance was longer than nine choices and the strength of relations decays quickly with path distance increasing beyond two–steps. The bold solid line describes the mean values of z_{ij} for each path distance. The shaded area indicates the variation in z_{ij} around each path distance length. For example, a four–step path distance represents a relation of .510 strength for people



reaching a small proportion of their contacts in four choices (top of the shaded area) but it represents a much weaker .020 strength relation for people who reach most of their contacts in fewer than four choices (bottom of the shaded area).

Here again, there are virtues and drawbacks. Like the fixed decay function proposed by Katz, this measure has the virtues of incorporating indirect connections in the network relations to be analyzed. In addition, this measure adjusts the strength of relation implied by a path distance for the structural context in which the path distance occurs. However, the measure does this in an arbitrary way. There is no systematic evidence to support the function relating relation decay to the number of persons reached at each path distance; it is simply less obviously wrong than a fixed decay function such as the one proposed by Katz and can draw some legitimacy from evidence on limits to the information that people can process simultaneously (e.g., Miller, 1956).

Output

The default printout is a tabulation of path distances with the mean strength of relation derived from each path distance (like the graph on page 71). The table at the top of the next page is generated when the command file and data on page 62 are submitted to STRUCTURE. Of the relations in the network (90 = N(N-1)); 20 are direct citations ($pd_{ij} = 1$), 16 are two-step indirect connections ($pd_{ij} = 2$), and the remaining 54 are instances of no direct or indirect connection from one another to another. The average z_{ij} for direct citations is .357 and the average z_{ij} corresponding to two-step path distances is .167. Where there is no direct or indirect connection, z_{ij} equals zero. No three-step path distances exist, so the search for indirect connections stopped at that point.

More detailed printout is elicited by a 1 in column 36 of the NETWORK command. The input choices are printed, then the path distances, and the final network of z_{ij} derived from the path distances. A zero is printed in the path distance output where no indirect connection could be found (within the limit of requested path distances). Also, the path distances and relations can be written as matrices or dyads to the output data file for analysis with other programs (1, 2, or 3 in column 38 of the NETWORK command).

Rank Order Sociometric Data

In small systems, it is feasible to ask respondents to rank order their relations with everyone else in the system. Such data are rare and typically limited to small systems, but they are more informative than binary choices. Rank-order choice data are handled in STRUCTURE using the procedure proposed by Killworth and Bernard (1974).

Like binary choice data, rank order sociometric data are read into STRUCTURE as a square matrix of integer variables for each network. Each row of each

```
                           frequency and
      path                 percentage of
    distance    mean Zij   all relations

       1          .357       20.  22.22%
       2          .167       16.  17.78%
       3          .000        0.    .00%
      NO
  CONNECTION      .000       54.  60.00%
```

network begins on a new line. In the command file, put a R in column 16 of the NETWORK command to indicate rank order choice data, put the number of actors in columns 12-14, and the number of networks in column 18. Here are the first few lines of the example data file NET-R on the program disk, taken from the second group in the fifteenth week of Newcomb and Nordlie's study of socializing and opinion in a college dormitory (Nordlie, 1958):

```
(17I3)
    12 15 05 10 11 06 04 07 16 08 09 02 03 13 14 01
 08    13 02 03 06 09 10 05 15 07 04 11 12 14 16 01
 08 11    10 12 03 05 13 04 14 06 02 09 15 07 16 01
 06 04 15    03 02 10 11 05 16 09 08 07 14 12 13 01
 05 04 13 02    08 10 06 01 14 12 11 03 09 15 16 07
 06 09 14 03 08    07 01 02 15 13 11 04 10 12 16 05
 12 04 08 06 14 10    05 09 16 02 01 07 11 13 15 03
```

and the following command file reads the data from the NET-R input data file:

```
DATA       NET-R
NETWORK    017 R
ANALYZE
```

The restrictions imposed by Killworth and Bernard are maintained. Ranks range from 1 to one less than the number of individuals in the system, where a respondent's strongest relation is rank 1 and his weakest is rank N-1. Each respondent is expected to rank his relations to every other person in each network. In the above example, the first respondent's strongest relation is with person 17 and his weakest relation is with person 10. Every relationship has to be given a unique rank (no ties), and diagonal elements are ignored.

A 1 in column 36 of the NETWORK command elicits detailed printout. First, the observed rank order data are printed. Second, the minimum rank order relations are printed indicating the minimum sum of observe rank order relations directly or indirectly connecting each row to each column (termed the minij relations by Killworth and Bernard). For example, the observed rank order relation of 12 from the first to the second person in the above data corresponds to a minimum rank order relation of 4. The sum of the observed rank order relations from person 1 to person 2 through person 17 as an intermediary is 4 (1's rank 1 relation to person 17 plus 17's rank 3 relation to person 2). This sum is smaller than any other indirect route from person 1 to person 2, so 4 is the minimum rank order relation from person 1 to person 2.

Third, the network of choice links needed to define each minimum rank order relation is printed (termed the catij relations by Killworth and Bernard). These linkage relations are analogous to path distances derived from binary choice data. A catij relation of 1 means that the observed rank order relation from i to j is smaller than any sum of rank order relations through intermediaries. A catij relation of 2 means that the sum of rank orders from i to j through one intermediary is smaller than the observed rank order relation from i to j. For example, there is a catij relation of 2 from person 1 to person 2 in the Newcomb-Nordlie data, indicating that a two-link rank order relation (through person 17 as just described) is the strongest connection from person 1 to person 2.

Network relations are derived from the catij relations just as network relations are derived from path distances. The frequency decay function is the default.

The fixed decay function is used if a 1 is put in column 30 of the NETWORK command.

If column 36 on the NETWORK command is left blank, the default printout is a tabulation of the observed rank order relations with the mean strength of relation derived at each rank and the distribution of catij relations. For example, the table below is generated when the command file and data on page 73 are submitted to STRUCTURE. The relations (272 = N*(N-1)) are equally divided between each of 16 observed ranks because respondents are forced to rank their relation with each other person without tied ranks. The table shows that all rank one and rank two relations are the strongest ties possible in the network (maximum z_{ij} and all catij relations equal to 1). Of the

```
initial              frequency and
rank                 percentage of        CATij linkages
order    mean Zij    all relations       1  2  3  4  5  6

    1       .602       17.   6.25%      17
    2       .602       17.   6.25%      17
    3       .578       17.   6.25%      16  1
    4       .453       17.   6.25%      10  7
    5       .318       17.   6.25%       4 11  1  1
    6       .325       17.   6.25%       6  5  5  1
    7       .311       17.   6.25%       4  9  4
    8       .253       17.   6.25%       3  9  3  2
    9       .180       17.   6.25%       1  9  6  1
   10       .149       17.   6.25%       0  6  9  2
   11       .201       17.   6.25%       1  9  6  1
   12       .325       17.   6.25%       6  7  3  1
   13       .384       17.   6.25%       9  4  1  3
   14       .398       17.   6.25%       9  4  3  1
   15       .270       17.   6.25%       5  6  5  1
   16       .388       17.   6.25%       7  9  1
```

17 observed rank twelve relations, in contrast, only 6 couldn't be improved by going through intermediaries (6 one-step catij relations). Seven of the relations were replaced by two-step indirect connections (7 two-step catij relations). One of these is the two-step catij relation from person 1 to person 2 through person 17 described above. Three of the observed rank twelve relations were replaced by three step indirect connections, and one was replaced by a four-step indirect connection.

Again, the catij path distances and final z_{ij} relations can be written to the output data file for analysis with other programs (column 38 of the NETWORK command).

JOINT INVOLVEMENT DATA

Network analysis of joint involvement data infers relations from involvement in the same events, or affiliations with the same groups. Breiger's (1974) discussion remains one of the best references on this topic. Two actors i and j are tied together to the extent that they are involved or affiliated with the same events or groups m. This is a very flexible form of network data, useful for studying a wide range of substantive questions. For example, where persons on boards of directors are events and corporations are actors, z_{ij} measures interlocking between the directorates of firms i and j (in the metric of number of directors they have in common). Where newspaper articles (or other archival accounts) are events and classes of people are actors, z_{ij} measures the relation between persons of class i with persons of class j (in the metric of number of news accounts in which they appear together). Where scientific articles are events and people cited in the article are actors, z_{ij} measures the connection between the kinds of knowledge for which persons i and j are known (in the metric of number of articles, or co-citations). Where an alter cited by a survey respondent is an event and the kinds of relationships cited in the respondent’s relation to the alter are “actors,” z_{ij} measures the tendency for kinds of relations i and j to occur in the same interpersonal relationships.

Networks are created by reading events in each network and aggregating the weight of events in which each pair of actors is involved. The general format is illustrated by the following example (see example file SOUTH.DAT):

```
(I2,1X,7A8/3X,7A8)
 3 Evelyn  Laura   Theresa
 3 Evelyn  Laura   Brenda
 4 Evelyn  Theresa Brenda  Charlott
 6 Evelyn  Laura   Theresa Brenda  CharlottFrances
 8 Evelyn  Laura   Theresa Brenda  CharlottFrances Eleanor
   Ruth
14 Evelyn  Laura   Theresa Brenda  Frances Eleanor Pearl
   Ruth    Verne   Myra    KatherinSylvia  Helen   Dorothy
 8 Evelyn  Laura   Theresa Brenda  CharlottEleanor Pearl
   Nora
10 Laura   Theresa Brenda  CharlottEleanor Ruth    Verne
   Sylvia  Nora    Helen
12 Evelyn  Theresa Pearl   Ruth    Verne   Myra
Katherin
   Sylvia  Nora    Dorothy Olivia  Flora
 5 Myra    KatherinSylvia  Nora    Helen
 4 Nora    Helen   Olivia  Flora
 6 Verne   Myra    KatherinSylvia  Nora    Helen
 3 KatherinSylvia  Nora
 3 KatherinSylvia  Nora
END NETWORK
```

The data are taken from Davis and Gardner’s (1941) description of cliques based on the frequency with which 18 women were mentioned together in 14 newspaper articles describing local social events. The first line is a format statement. If such statements are not familiar, see the Appendix to this

Manual, or Appendix D in the ASSISTANT command booklet. Each event starts on a new line. The first variable (in integer, or I, format) defines the number of actors in the event. For example, 3 women are mentioned in the article on the first event and 14 women are mentioned in the article describing the sixth event. The program reads the identification codes of participants in the event (in alphanumeric, or A, format). A maximum of 8 characters are retained to identify actors. Any ID CODES command in the command file is ignored. Actors are assigned identification codes as they are read in the event data. The END NETWORK line signifies the end of events defining the first network. If the file looked like this:

```
(I2,1X,7A8/3X,7A8)
 3 Evelyn  Laura   Brenda
 4 Evelyn  Theresa Brenda  Charlott
 6 Evelyn  Laura   Theresa Brenda  CharlottFrances
 8 Evelyn  Laura   Theresa Brenda  CharlottFrances Eleanor
   Ruth
END NETWORK
14 Evelyn  Laura   Theresa Brenda  Frances Eleanor Pearl
   Ruth    Verne   Myra    KatherinSylvia  Helen   Dorothy
 8 Evelyn  Laura   Theresa Brenda  CharlottEleanor Pearl
   Nora
END NETWORK
```

then two networks would be created; the first network defined by joint involvements in the first four events, and the second defined by involvements in the next two events.

If you select the option of reading event weights from the input data file (a 3 in column 20 of the NETWORK command), then each event is preceded by a real number (in F format or an equivalent) giving its weight. For example, the above articles could be weighted by the proportion of the society section allocated to the article (the weights are hypothetical):

```
(F2.2,I2,1X,7A8/5X,7A8)
20 3 Evelyn  Laura   Brenda
25 4 Evelyn  Theresa Brenda  Charlott
33 6 Evelyn  Laura   Theresa Brenda  CharlottFrances
40 8 Evelyn  Laura   Theresa Brenda  CharlottFrances Eleanor
     Ruth
END NETWORK
8014 Evelyn  Laura   Theresa Brenda  Frances Eleanor Pearl
     Ruth    Verne   Myra    KatherinSylvia  Helen   Dorothy
10 8 Evelyn  Laura   Theresa Brenda  CharlottEleanor Pearl
     Nora
END NETWORK
```

which would indicate that much more importance was given to the eighth event (80% of the society section) than the seventh (20%) — or the eighth event took place in a period with little society news to fill the society section of the newspaper.

Measuring Relations

The default is to sum across events in which i and j are both involved, where f_{im} equals the frequency of actor i's affiliation with event m, and the smaller of f_{im} versus f_{jm} is retained as the frequency of joint affiliation:

$$z_{ij} = z_{ji} = \sum_m \min(f_{im}, f_{jm}),$$

Joint involvements are typically binary; f_{im} either equals 0 indicating that actor i is not connected with event m, or it equals 1 to indicate that he is. In a network based on such data, element z_{ii} is the number of events in which actor i is involved and element z_{ij} is the number of events in which actors i and j are both involved. STRUCTURE also allows nonbinary event data. This is not usual, but it is possible. For example, f_{im} could be the number of times that actor i is cited in article m. If a single actor appears multiple times in any one event, a message to that effect is printed with the raw network data describing the interpretation of the network relations. Diagonal element z_{ii} is the total number of times that actor j is observed in events and element z_{ij} is the frequency with which actors i and j are observed in the same events. To generate the simpler event count measures of network relations, you can use the JEDIT utility program described below to eliminate redundant affiliations within events (see pages 82ff).

Events can be weighted to represent their relative importance. Joint involvement relations are derived from weighted events as follows:

$$z_{ij} = z_{ji} = \sum_m w_m[\min(f_{im}, f_{jm})],$$

where w_m increases with the importance of event m. Weights are defined in column 20 of the NETWORK command. The default is to set all weights to 1, giving equal weight to events and making z_{ij} equal the number of events in which i and j are jointly involved.

There are three alternative weight options. The first is to weight events by their coverage (1 in column 20). Inclusive events are emphasized by weighting in proportion to the portion of the entire system involved in an event: $w_m = N_m/N$, where N_m is the number of actors involved in event m and N is the total number of actors in the system. In data on participants in social events, this emphasizes ties through large events. In co-citation data, it emphasizes ties in long review articles. In corporate director data, it emphasizes ties through directors on the boards of many firms.

The next option has the opposite effect (2 in column 20). Exclusive events are emphasized by weighting in inverse proportion to the number of actors involved in an event: $w_m = 1/N_m$. In data on participants in social events, this emphasizes exclusive social gatherings of few individuals. In co-citations, it emphasizes short notes.

The last option lets you define your own weights and read them in with the data as on page 79 (a 3 in column 20). Where events are newspaper articles, for example, column inches are often used as weight under the presumption that the social importance of an event is correlated with the amount of space a newspaper editor allocates to a story about the event.

Additional measurement alternatives are available with column 32 of the NETWORK command. Diagonal element z_{ii} is always the largest network element in row i. Other elements can be as large, but none can be larger (z_{ii} equals z_{ij} if j and i are in identical events). A 1 in column 32, the option to measure z_{ij} relative to the maximum element in row i, will transform the z_{ij} to measure the proportion of events involving actor i that also involve actor j. Alternatively, a 2 in column 32 will transform the z_{ij} to measure the relative extent to which i is involved in the events in which j is involved.

Output

The default printout is a tabulation of events in descending order of the number of actors in them. For example, the first rows of the tabulation for the Davis and Gardner data in SOUTH.DAT is given at the top of the next page. It shows that the largest event involved 14 women, and there was one such event (constituting 7.14% of all events) with a total weight of 1 (given each event weighted equally). The next largest event involved 12 women, and so on, down to the events involving only one individual.

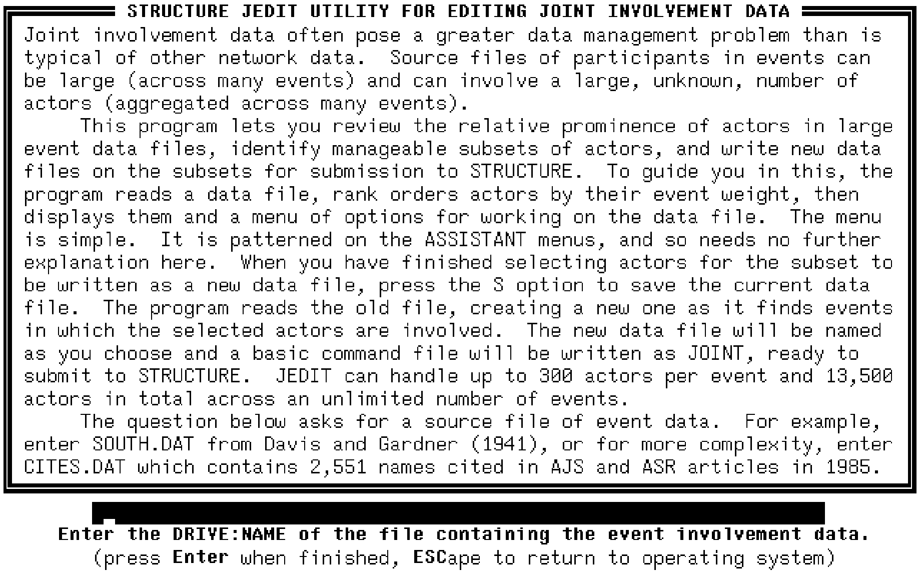
Detailed printout is requested with 1 in column 36 of the NETWORK command. Actors are listed in the program output in order of their detection. Actor 1 will be the first actor detected in the first event, actor 2 will be the second actor detected, and so on. The first matrix contains the raw z_{ij} defined above (e.g., joint involvement frequencies if each event is weighted equally), followed by the final matrix of relations. The joint involvement frequencies

```
                    frequency and
    number of        percentage
   participants     of all events      total weight

       14              1    7.14%             1.0
       13              0     .00%              .0
       12              1    7.14%             1.0
       11              0     .00%              .0
       10              1    7.14%             1.0
```

and final z_{ij} relations can be written to the output data file for analysis with other programs (column 38 of the NETWORK command).

JEDIT Data Editing Program

Joint involvement data often pose a special data management problem. Files of participants in events can be large (across many events) and can involve a large, unknown, number of actors (aggregated across many events). I have written a utility program, JEDIT, to help manage large event data files in preparation for network analyses. (Look for JEDIT.EXE on the program disk). The program will accept STRUCTURE data files or data in the format of a listing from a data base program. With JEDIT, you can review the relative prominence of actors in large event data files, identify manageable subsets of actors for analysis, and write new data files on the subsets for submission to STRUCTURE. To guide you in this, the program reads a data file, rank orders actors by their event weight, then displays them and a menu of options for working on the data file. When you've selected a subset of actors, press the S option to save them as a new data file. The program reads the old file, creating a new one as it finds events in which the selected actors are involved. The new data file will be named as you choose and a basic command file will be written as JOINT, ready to submit to STRUCTURE. JEDIT can handle up



```
═════ STRUCTURE JEDIT UTILITY FOR EDITING JOINT INVOLVEMENT DATA ═════
Joint involvement data often pose a greater data management problem than is
typical of other network data.  Source files of participants in events can
be large (across many events) and can involve a large, unknown, number of
actors (aggregated across many events).
     This program lets you review the relative prominence of actors in large
event data files, identify manageable subsets of actors, and write new data
files on the subsets for submission to STRUCTURE.  To guide you in this, the
program reads a data file, rank orders actors by their event weight, then
displays them and a menu of options for working on the data file.  The menu
is simple.  It is patterned on the ASSISTANT menus, and so needs no further
explanation here.  When you have finished selecting actors for the subset to
be written as a new data file, press the S option to save the current data
file.  The program reads the old file, creating a new one as it finds events
in which the selected actors are involved.  The new data file will be named
as you choose and a basic command file will be written as JOINT, ready to
submit to STRUCTURE.  JEDIT can handle up to 300 actors per event and 13,500
actors in total across an unlimited number of events.
     The question below asks for a source file of event data.  For example,
enter SOUTH.DAT from Davis and Gardner (1941), or for more complexity, enter
CITES.DAT which contains 2,551 names cited in AJS and ASR articles in 1985.

   Enter the DRIVE:NAME of the file containing the event involvement data.
      (press Enter when finished, ESCape to return to operating system)
```

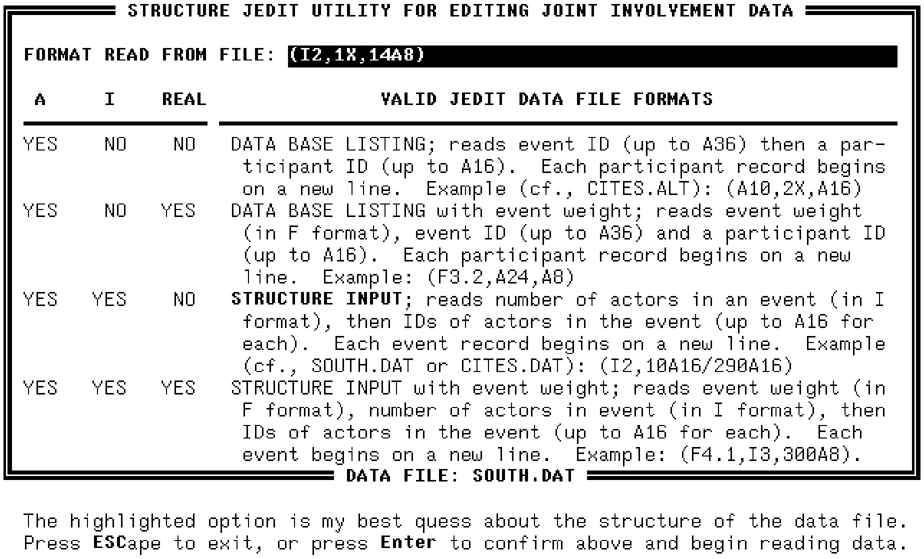
to 300 actors per event and 13,500 actors in total across an unlimited number of events.

Some example data base files are included on the program disk. For the moment, I'll use the simple data in SOUTH.DAT on 18 women and 14 events. Presuming you have JEDIT.EXE on your hard disk or the current default drive, enter JEDIT at the DOS prompt. You should see the screen displayed above.

The question at the bottom of the screen asks for the name of a data file to read as a source for creating new data files. Type SOUTH.DAT, then press Enter to enter the name of the file. If the data file isn't on the current default drive, include whatever drive and path names are needed to get to the file (e.g., C:\STRUC\DATA\SOUTH.DAT).   Press Enter without any file name to convert to a monochrome display.

As soon as you enter the name of the file and the program locates the source file, you see the file format screen displayed below. You could have come to this screen directly by entering the program and file name together at the DOS prompt (JEDIT SOUTH.DAT). The file format screen lists the format statement read from the first line of the source data file. It then lists the four file formats accepted by JEDIT. Of the four possible file formats, one will be highlighted to indicate the program's best guess about the structure of the source file based on the alphanumeric, integer, and real number format elements in the format statement. The displayed screen indicates that SOUTH.DAT is in STRUCTURE input format with no weight data.

This screen is your guide to generating source data files. STRUCTURE event data format has been described above. A quick note is in order on the data base listing format. Each record in the data base format begins with an event identification code (of up to 32 characters) followed by an actor identification



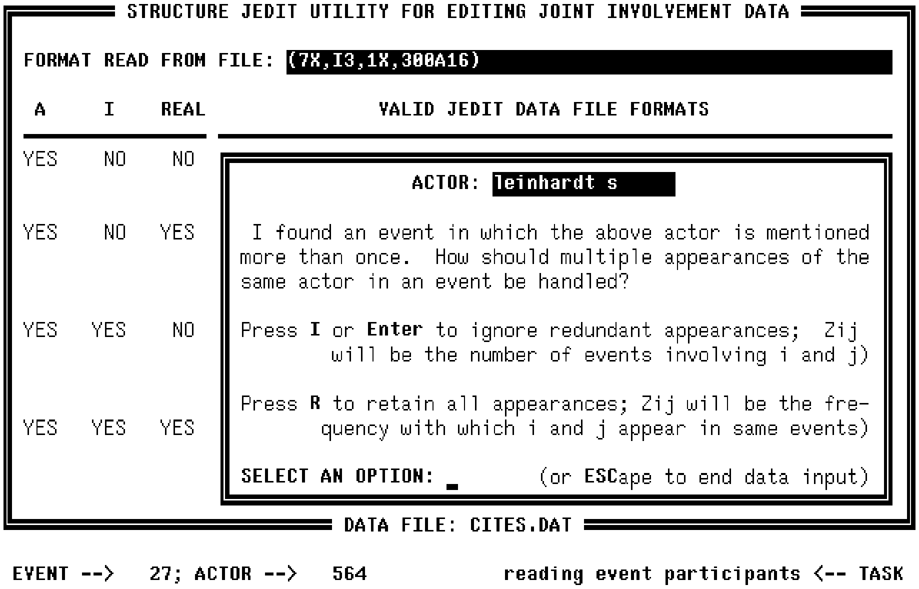
```
═══════ STRUCTURE JEDIT UTILITY FOR EDITING JOINT INVOLVEMENT DATA ═══════

FORMAT READ FROM FILE: (I2,1X,14A8)

 A     I    REAL                VALID JEDIT DATA FILE FORMATS

YES    NO    NO   DATA BASE LISTING; reads event ID (up to A36) then a par-
                   ticipant ID (up to A16).  Each participant record begins
                   on a new line.  Example (cf., CITES.ALT): (A10,2X,A16)
YES    NO   YES   DATA BASE LISTING with event weight; reads event weight
                   (in F format), event ID (up to A36) and a participant ID
                   (up to A16).  Each participant record begins on a new
                   line.  Example: (F3.2,A24,A8)
YES   YES    NO   STRUCTURE INPUT; reads number of actors in an event (in I
                   format), then IDs of actors in the event (up to A16 for
                   each).  Each event record begins on a new line.  Example
                   (cf., SOUTH.DAT or CITES.DAT): (I2,10A16/290A16)
YES   YES   YES   STRUCTURE INPUT with event weight; reads event weight (in
                   F format), number of actors in event (in I format), then
                   IDs of actors in the event (up to A16 for each).  Each
                   event begins on a new line.  Example: (F4.1,I3,300A8).
════════════════════════════ DATA FILE: SOUTH.DAT ═══════════════════════

The highlighted option is my best quess about the structure of the data file.
Press ESCape to exit, or press Enter to confirm above and begin reading data.
```

code (of up to 16 characters). It is important that all data on the same event are together. When reading data base list format, JEDIT creates a new event when the event identification code on one line is different from the identification code on the preceding line. Before creating a source file from your data base program, be sure to sort the file by event identification codes. Note also that the event identification code is long enough to let you use descriptive names of people or firms. This can be helpful in interpreting the output data file because the event identification code is included at the front of each event (cf., the interlocking directorate information in the example file INTERLOK.DAT).

When you press Enter at the file format screen, the program begins reading the source data file. As it reads, it displays the events and actors detected.

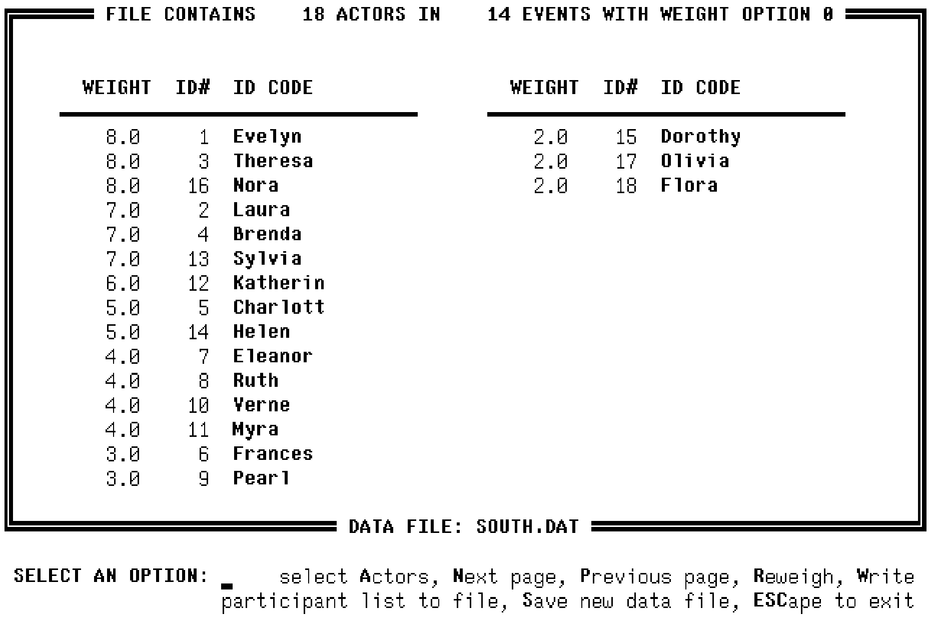
If an actor is detected more than once in a single event, then a message window is displayed. You will not see this window with SOUTH.DAT because each woman is mentioned only once when she participated in an event. However, you will see the window if you have JEDIT read the citation data in the example data file CITES.DAT. To illustrate, the screen you will get with CITES.DAT is displayed at the top of the next page. In article 27, the program detected more than one citation to Sam Leinhardt. You have three options. One is to exit. The second is to ignore redundant affiliations, whereupon the output data file will generate z_{ij} in the metric of number of events in which i and j are mentioned. The third option is to retain all mentions of any actor, whereupon the output data file will generate z_{ij} in the metric of frequency with which actors i and j appear in the same events. The **I** option generates more easily interpreted network relations and is the default here if you just press Enter.

Back to reading SOUTH.DAT, the next screen you see is the main work screen and menu. The screen for SOUTH.DAT is displayed at the top of page 87. The window contains a rank order list of all actors detected, beginning at



```
═══════ STRUCTURE JEDIT UTILITY FOR EDITING JOINT INVOLVEMENT DATA ═══════

 FORMAT READ FROM FILE: (7X,I3,1X,300A16)

  A     I    REAL              VALID JEDIT DATA FILE FORMATS

 YES    NO    NO
                               ACTOR: leinhardt s

 YES    NO   YES      I found an event in which the above actor is mentioned
                     more than once.  How should multiple appearances of the
                     same actor in an event be handled?

 YES   YES    NO     Press I or Enter to ignore redundant appearances;  Zij
                            will be the number of events involving i and j)

                     Press R to retain all appearances; Zij will be the fre-
 YES   YES   YES           quency with which i and j appear in same events)

                     SELECT AN OPTION: _        (or ESCape to end data input)

═══════════════════════════ DATA FILE: CITES.DAT ═══════════════════

 EVENT -->   27; ACTOR -->   564          reading event participants <-- TASK
```

the top of the list with the most frequent actor (or more generally, the actor appearing in the largest aggregate weight of events). The weight printed next to each name is the sum of weights across the actor's affiliations ($\sum_m w_m f_{jm}$). The ID# is the sequential order in which the actor was found (e.g., 16 is the sixteenth actor detected). The name is listed in two parts. JEDIT reads up to 16 characters of actor identification. The first 8 of the characters are highlighted in the work screen because these are the only characters that STRUCTURE will recognize. Any other characters past the first 8 are displayed here only to help you identify actors, especially actors who might look the same if they were only identified by the first 8 characters (e.g., Harrison White and Harrison Ford are identical in the first 8 characters). Where separate actors are identical in the first 8 characters of their name, you can go back to your data base program to edit the names to make them distinguishable. The women in SOUTH.DAT are only identified by 8

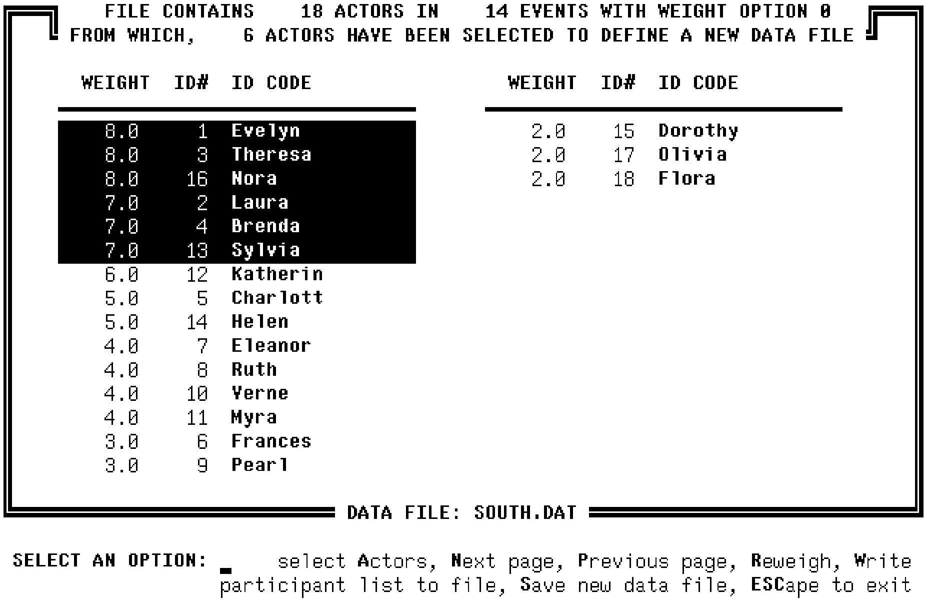


```
 FILE CONTAINS    18 ACTORS IN    14 EVENTS WITH WEIGHT OPTION 0

   WEIGHT  ID#  ID CODE                     WEIGHT  ID#  ID CODE
  ______________________________          ______________________________
     8.0     1  Evelyn                         2.0    15  Dorothy
     8.0     3  Theresa                        2.0    17  Olivia
     8.0    16  Nora                           2.0    18  Flora
     7.0     2  Laura
     7.0     4  Brenda
     7.0    13  Sylvia
     6.0    12  Katherin
     5.0     5  Charlott
     5.0    14  Helen
     4.0     7  Eleanor
     4.0     8  Ruth
     4.0    10  Verne
     4.0    11  Myra
     3.0     6  Frances
     3.0     9  Pearl

                       DATA FILE: SOUTH.DAT

SELECT AN OPTION: _    select Actors, Next page, Previous page, Reweigh, Write
                   participant list to file, Save new data file, ESCape to exit
```

characters, so every character in each woman's name is highlighted. When you load CITES.DAT you will see the importance of being able to see 16 characters of identification data.

Press the **A** option to select actors. You see a message asking you to enter the ID# of an actor to select/unselect or press **ESC**ape to return to the main menu to get a new window of actors. If you enter one of the numbers listed in the ID# column of the window, the actor is highlighted indicating that he or she has been selected to include in a new data file. If you enter the same number again, the highlighting is removed indicating that he or she will not be included in the new data file. More simply, if you just press Enter, the program will begin highlighting actors at the top of the list (or just after the last ID# you entered). Press Enter to select the six most often cited women in SOUTH.DAT. Evelyn is highlighted after your first press. Theresa is highlighted next, and so on until you reach Sylvia. Now press **ESC**ape. You are returned to the main menu and you screen should look like the one displayed below.

To save the selected actors as a new data file, press the **S** option. You are asked to name a file to contain the data (or to press Enter to accept the default of writing the data to a file named JOINT.DAT). Press Enter. The data are written with a rudimentary command file named JOINT defining the number of actors and networks in JOINT.DAT and a DATA command with the input data file name (in this case JOINT.DAT). You can edit the command file (e.g., with ASSISTANT) to define analyses of the data, or you can submit the file JOINT as is to STRUCTURE for a cluster and spatial analysis of structural equivalence among the actors.



```
      FILE CONTAINS    18 ACTORS IN    14 EVENTS WITH WEIGHT OPTION 0
   FROM WHICH,    6 ACTORS HAVE BEEN SELECTED TO DEFINE A NEW DATA FILE

    WEIGHT  ID#  ID CODE                 WEIGHT  ID#  ID CODE
   ______________________________       ______________________________
       8.0    1  Evelyn                     2.0   15  Dorothy
       8.0    3  Theresa                    2.0   17  Olivia
       8.0   16  Nora                       2.0   18  Flora
       7.0    2  Laura
       7.0    4  Brenda
       7.0   13  Sylvia
       6.0   12  Katherin
       5.0    5  Charlott
       5.0   14  Helen
       4.0    7  Eleanor
       4.0    8  Ruth
       4.0   10  Verne
       4.0   11  Myra
       3.0    6  Frances
       3.0    9  Pearl

 ========================== DATA FILE: SOUTH.DAT ==========================

  SELECT AN OPTION: _    select Actors, Next page, Previous page, Reweigh, Write
                     participant list to file, Save new data file, ESCape to exit
```

Viewing and writing new data files is the basic task of JEDIT. The other options in the main menu are related to that task. Press **N** to see the next window of actors in the source data file. Press **P** to see the previous window. All of the women in SOUTH.DAT fit into one window so the **N** and **P** options have no effect. Press **R** to reweigh events and re-rank actors. This will give you a sense of how the alternative weight options in STRUCTURE will affect the relative prominence of actors in the source file. The reweigh option will not be displayed if event weights are defined in the source data file. Press **W** to write a list of all actors to an output file. This option gives you a list of every actor in the source data file (up to 13,500 actors) in descending order of event weight. After the list is written, you can send it to your printer for a hard copy.

The SOUTH.DAT data set has heuristic value, but it is hardly a challenge or a realistic example of the data sets likely in serious contemporary research.

More realistic examples are included on the program disk. CITES.DAT is a data file on the 2,551 people cited in the 119 articles published in 1985 in the *American Journal of Sociology* and the *American Sociological Review*. The file defines two networks; a network of co-citations based on the AJS articles and a network based on the ASR articles. It would be inefficient to study the two networks among all 2,551 cited people. Most of the people are rarely cited. With the JEDIT utility, you can select subsets of people from CITES.DAT that define substantive areas of interest to you, then study the network structure of each area as represented by co-citations in the two leading sociology journals. The example command file CO-CITE describes the co-citation structure among 85 people prominent in these data. Another data file on the program disk, CITES.ALT, contains the same information as CITES.DAT but in the more familiar data base listing format rather than the STRUCTURE event data format. This is to show you how the same data would appear in the two file formats. Still another example file, GSS.DAT, contains data on 16 kinds of relations in the relationships cited by middle-age

(37-46 years old), married respondents in the 1985 General Social Survey. These data are described in the GSS example command file. From this file, you can create a network of connections between kinds of relations in order to describe kinds of network content in the elicited relations (e.g., Burt, 1990b). Finally, the example file INTERLOK.ALT is a data base listing of the 166 firms and 1917 directors used by Mizruchi (1982) to describe the structure of interlocking directorates among large American firms at the turn of the century. From this file you can select a subset of firms among which you wish to describe the network of interlocking directorates (compare files INTERLOK.DAT and INTERLOK).

MONTE CARLO DATA

Statistical reasoning rarely informs prevailing knowledge about the network models most often used in empirical research. A model's properties are typically "established" by an illustrative application, rarely with any discussion of how the system selected for illustration compares with other kinds of social structures to which the network model might be applied. And yet, as occasional simulation studies show (e.g., Holland and Leinhardt, 1970; Doreian, 1980; Dow et al., 1982; Yamagishi et al., 1988; Xia, 1991), the most efficient and effective way to discover a model's properties across kinds of social structures is to design diverse social structures and run Monte Carlo trials of the model across the structures. Taking their name from the city famous for its casinos, Monte Carlo studies are analyses of large numbers of replicate samples of random data. With its explicit aggregation of relations into causal variables and its sensitivity to the specific structure of relations in a system, network analysis is an especially fruitful place for such studies. Relative to obtaining data on observed social structures, simulated network data are less expensive (no field work), more quickly obtained (thereby shortening the time between theoretical question and empirical answer), more powerful (given unlimited replicate samples), and more easily tailored to the research question under study (since you define the population structure in which network data are to be drawn at random).

The MONTE command in the STRUCTURE program makes it possible to carry out Monte Carlo studies of any of the network models in the program. You can simulate binary sociometric choice data, rank order sociometric choice data, or joint involvement data, but the most sophisticated options for Monte Carlo analysis in STRUCTURE are limited to binary sociometric choice data. A large portion of the ASSISTANT program is devoted to making it easy to access those options. I strongly recommend using the design procedure in ASSISTANT to set up Monte Carlo analyses; both for ease of use, and for immediately showing you the population structure implicit in the constraints you request for an analysis. The design procedure in ASSISTANT helps you construct social structures with contrasting qualities that best reveal variation in the formal properties of a network model selected for study. Comparisons across the kinds of social structures of simulated applications within each kind of structure then describe the formal properties of the model as a tool for empirical research. The design procedure lets you explore alternative social structures to determine which to include in a Monte Carlo study. The questions to be answered range from theoretical questions about alternative formal definitions of abstract models to very specific methodological decisions.

Monte Carlo network analysis involves five steps: First, define constraints under which network data will be drawn at random. The constraints define the population social structure in which samples of data will be drawn. Second, draw a sample of random data and from them derive the network relations (the z_{ij}) to be analyzed. Third, run whatever analytical models have been selected for study (contagion, power, autonomy, etc.). Each pass through steps two and three constitutes a Monte Carlo trial. Successive trials create replicate samples where two samples are replicates in the sense that they are samples of network data drawn from the same population social structure. The fourth step is to run many, many trials. The fifth step is to describe the distribution of results obtained across the many, many replicate samples to define a sampling distribution of results for the population structure.

Let me illustrate with a simple methodological decision. In the course of designing a limited budget study of contagion, I wanted to know how sensitive contagion effects in the study population would be to alternative measures of network relations. Specifically, I wanted to know how few choices I could record without obscuring contagion effects. I knew that the study population was organized primarily in terms of separate cliques. To answer the question, I defined a hypothetical study population in which contagion was perfect within cliques. Everyone in the same clique gave the same response on a criterion variable. The task for Monte Carlo analysis was to determine how reliably I could recover the known strong contagion effect under different measures of network relations. I constructed the following target image matrix that defines a three position population social structure where each position is a clique in the sense that clique members have strong relations with one another and weak relations with people in other cliques:

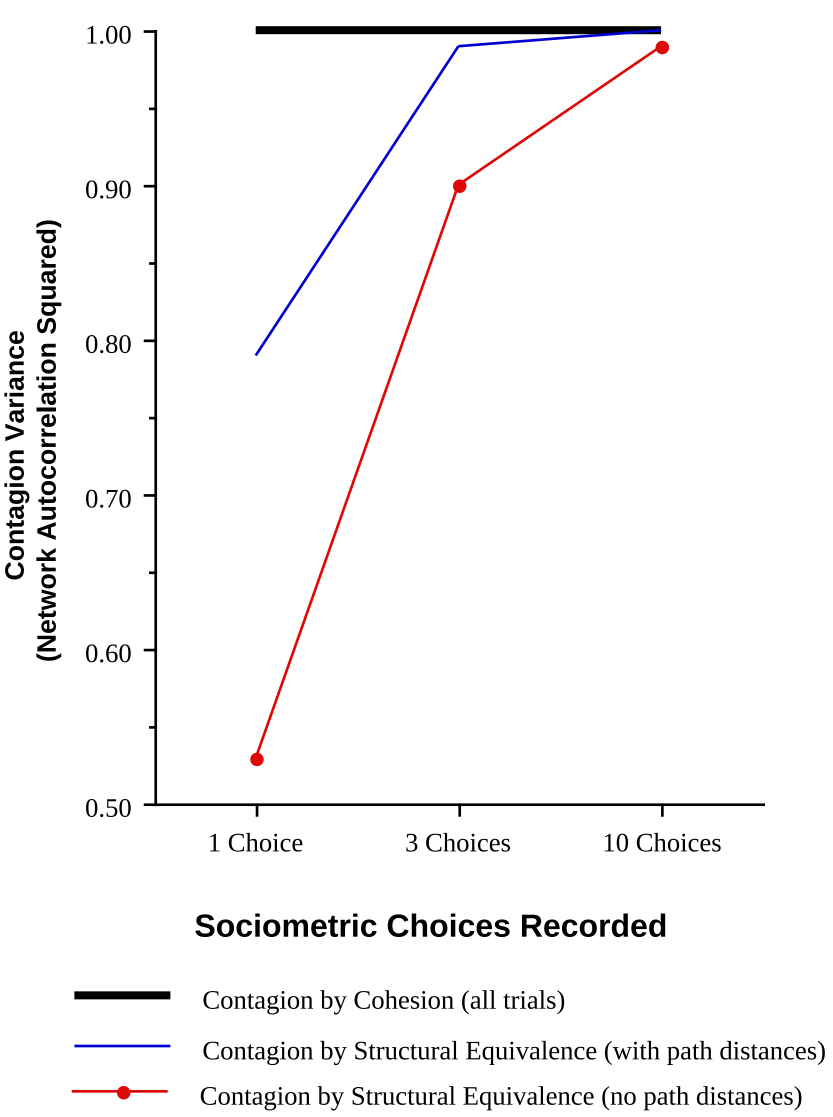
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and I created a response variable in which members of each clique gave identical responses and members of different cliques gave different responses. Members of the first clique all gave a response of “1”, members of the second clique all gave a response of “2”, and a response of “3” was attributed to everyone in the third clique.

I then ran contagion analyses in Monte Carlo trials varying network size and measures of network relations. I used the uniform probability distribution to draw sociometric choices.

The key results for networks of size 60 are graphed on the next page (as averages across 10 trials per test condition). Network size turned out to be a relatively minor issue beyond size 40, well under the size I would be studying, so I didn't pursue detailed results on size effects. The graph shows the effect of network measure on the expected strength of contagion effects. The graph plots the strength of an estimated contagion effect on the vertical axis as the variance in the response variable explained by contagion (network autocorrelation squared). The horizontal axis indicates the maximum number of sociometric choices recorded from each respondent.

In this example, contagion by cohesion or structural equivalence should describe 100% of the variance in the response variable because people in the cliques are both cohesive and structurally equivalent. The central point in the graph is that measures of contagion by cohesion are not at all affected by changing the relation measure, but measures of contagion by structural equivalence are strongly affected. At the top of the graph, the solid line shows that contagion by cohesion explains 100% of the variance in the response variable whether I record one, three, or ten sociometric choices per respondent. Not surprisingly, there is no effect of including indirect contacts by measuring relations as normalized path distances. In contrast, contagion by equivalence at times describes little more than 50% of the response variance — despite the fact that the effect is known to be 100%! The ability of structural equivalence to recover the known contagion effect improves quickly with the number of sociometric choices recorded from each respondent. It is also improved by measuring relations as a function of path distances rather than as directly measured by binary sociometric choices. The efficiency of estimates follows these results on the expected strength of effects. For example, plots of the


1.00
0.90
0.80
0.70
0.60
0.50
Contagion Variance
(Network Autocorrelation Squared)
1 Choice
3 Choices
10 Choices
Sociometric Choices Recorded
Contagion by Cohesion (all trials)
Contagion by Structural Equivalence (with path distances)
Contagion by Structural Equivalence (no path distances)

jackknife subsample estimates of contagion have the greatest variance where the expected contagion effect is lowest — for contagion by structural equivalence where only one sociometric choice is recorded per respondent and path distances are not used to measure relations.

My conclusion from this exercise was that it would be a mistake to record fewer than three sociometric choices per respondent — despite obvious cost savings in interview and analysis time — and that information on indirect connections must be included in the measures of network relations. Otherwise, the research would have been very nearly worthless as a comparative study of alternative contagion models, since it could have been strongly biased against detecting evidence of contagion by structural equivalence.

This result is easily understood once it is known. The Monte Carlo results simply made it easier for me to be aware of the problem. Under contagion by cohesion, each respondent is expected to give the response of the person to whom he is tied by his sociometric citation. These citations were forced to occur within cliques and all members of the same clique were forced to give the same response. Therefore, no matter how relations are measured, choosers will give the same response as the people they cite, so the strong contagion effect will be recovered.

In contrast, under contagion by structural equivalence, each respondent is expected to give the response of persons who are his structural peers in the network, people who have similar relations with the same other people. With so little information on the social structure recorded when one choice is recorded per respondent, the structural equivalence of clique members is hard to see. Structural equivalence within cliques is as strong as equivalence between cliques. This is why path distances are so important. By including indirect connections in relations — where all occur within cliques — the equivalence of people within cliques is more apparent because they are more obviously connected (directly or indirectly) to the same people and disconnected from the same people. When ten choices are recorded from each respondent, so positions in this network are clearly identified by choice data, the known contagion effect is almost equally well recovered with or without path distance data.

This very simple example illustrates the practical utility of Monte Carlo network analysis with the target image matrices. More, to the experienced network analyst thinking of the many parameters that could be varied, the example illustrates the potential power such analysis could bring to methodological and theoretical studies of popular network models so often used blindly in empirical research (see Xia, 1991, for elaboration).

Defining the Network Data for a Monte Carlo Study

The simplest command file for a Monte Carlo analysis contains three commands. A NETWORK command is needed to define the number of actors in the system, the kind of network data to define relations, and the number of networks (see pages 38-39). A MONTE command is needed to invoke the random number generator and disable the usual search for an input data file (see page 36). An ANALYZE command signals the program to begin the analysis. For example, here is a minimal command file requesting random binary sociometric choice data:

```
NETWORK    085 C
MONTE
ANALYZE
```

The NETWORK command defines 1 network (by default) of binary sociometric choice data (C data) among 85 actors. The path distance parameter is blank, so all indirect connections will be sought. Relations will be derived from path distances by the frequency decay function and basic descriptive indices will be reported. The results will be printed to a file named MONTE.PRT (again, by default). Note the absence of a DATA command. The network data for this analysis will be generated inside the program. The MONTE command is blank so 1 trial will be generated in which each of the 85 actors can make one sociometric choice and each other actor is equally likely to be cited. You will be prompted on the screen for a number to seed the random number generator (to be explained in a moment). The following command file is a more useful version of the above:

```
PRINT       Test
WRITE       Test.dat
NETWORK     085 C 0 0 0 0   0 0 0 2 0
MONTE       0  50 639 3
POWER       0 000 0 1
ANALYZE
```

A PRINT command is included so that the results of different analyses are not printed on top of one another in the default printout file. The NETWORK command is changed to suppress all relation printout (2 in column 34) and power scores are written to an output data file named TEST.DAT for statistical analysis. The MONTE command is changed to request 50 trials in which each actor makes 3 citations where each other actor is equally likely to be cited. The 639 on the MONTE command seeds the random number generator.

The seed number is an important control for comparing alternative analyses. It guarantees that analyses are based on the same random data. The random number generator produces a stream of random fractions initialized by the seed number. Two analyses requiring the same number of calls to the random number generator will be based on identical data if they begin with the same seed number. If you submit the above 50 trial analysis on Wednesday and again on Saturday, each relation in each of the 50 networks will be identical in the two analyses. More important, if you submit the following analyses of contagion by cohesion versus equivalence (command parameters are given on pages 24-25):

```
PRINT      Coh.prt
WRITE      Coh.dat
NETWORK    085 C 0 0 0 0   0 0 0 2 0
MONTE      0  50 987 3
CONTAGION  2     4
ANALYZE
PRINT      Equiv.prt
WRITE      Equiv.dat
NETWORK    085 C 0 0 0 0   0 0 0 2 0
MONTE      0  50 987 3
CONTAGION  0     4
ANALYZE
```

you know that the two analyses are based on identical network and response data because of their identical 987 seed number, so differences between the analyses are created by the alternative contagion models.*

Binary Sociometric Choice Data

It is time to be more specific about the kind of network data requested. If the simulated network data are binary sociometric choices, as in the above examples, the choice generator is analogous to a fixed choice sociometric question. Each person is given a fixed number of choices, n, to make in each network where the number of choices is defined by the parameter in columns

*The random numbers are pseudo-random numbers generated by the linear congruential method (e.g., Pidd, 1984:Chp. 7). Beginning with an integer seed number not divisible by 2 (even seeds are increased by 1 in STRUCTURE), successive random numbers are obtained as follows; $X_{i+1} = \mathrm{Mod}_m(aX_i + c)$, where the new random number equals X_{i+1}/m, X_0 is the initial seed number, a, c and m are constants, and Mod_m means that X_{i+1} is the remainder of dividing (aX_{i+c}) by m. The default in STRUCTURE is to run two iterations, or cycles, for each random number drawn from the uniform probability distribution. For more densely distributed random numbers, you can increase the cycles with the parameter in column 12 of the DEFAULTS command. However, graphic and chi-square tests show acceptable results from the generator with only two cycles when the source code is compiled for DOS with either Microsoft's FORTRAN or MicroWay's NDP-FORTRAN-386.

21-22 of the MONTE command. Choices are generated for each row actor as a respondent; all choices for row actor 1, then all choices for row actor 2, and so on. For each row actor, a pool of potential choices is identified. In the above examples, each row actor can cite any other actor (but that need not be true when a target image matrix is used to constrain choices to specific blocks in a population social structure). The row actor draws n names at random from the pool of everyone he can cite. A six-digit random number between 0 and 1 is drawn for each choice. The person receiving the choice is then determined by multiplying the random fraction by the number of people in the pool of potential choices and rounding to the nearest integer. For example, a random number of .512 drawn for a pool of ten possible choices generates a choice to the fifth person of the ten possible. Choices occur with replacement, so the number of people cited by the row actor can be less than the number of choices made. For example, drawing random numbers .512 and .523 for a row actor with ten choice targets would generate two choices to the same person; the fifth person of the ten possible.

Rank Order Sociometric Choice Data

If the simulated network data are rank order sociometric choices, requested with an R in column 16 of the NETWORK command, then each row actor's choices within each network are a random rearrangement of the other actors in the network. This is carried out for a row actor by drawing a random number between 0 and 1 for each of the N-1 other actors in a network. The N-1 random numbers are then sorted from highest to lowest. The actor with the highest number is the row actor's first choice, the next highest his second choice, and so on. This is repeated for each row actor in each network. Note that this procedure is unaffected by the probability distribution from which random numbers are drawn or the number of choices drawn. This means that rank order sociometric data does not offer the same controls (described below) for varying network density and prominence that you have with binary choice data or event data.

Joint Involvement Data

If the simulated network data are joint involvements, requested with a J in column 16 of the NETWORK command, then actors are randomly involved in events and a network of joint involvements is built in the usual way. The number of events defining a network is set equal to the number of actors in columns 12-14 of the NETWORK command. A fixed number of actors, n, is chosen in each network to participate in each event where the number of choices per event is defined by columns 21-22 of the MONTE command. A binary matrix of random connections between events (rows) and actors (columns) is created by the procedure used to draw binary sociometric choices from row actors to column actors. The joint involvement network is then built from actors participating in the same events.

Varying Network Density

Density is the average strength of relations in a group or network (mean z_{ij}). Within cell blocks of a target image matrix or within a whole network of unconstrained choice, density increases with the number of choices drawn for each person and decreases with network size. Varying these two parameters to change network density across trials is useful for assessing the sensitivity of models to research methods limiting respondents to a fixed number of choices, or assessing the power of resolution between alternative network models purporting to describe the same phenomenon.

To minimize density in sociometric choice data, let each person make only one sociometric choice in a very large network and measure relations as binary choices. In event data, let each person in a large network be involved in only one event.

To maximize density, keep network size small and the number of choices high. Also, ceteris paribus, the density of relations derived from binary sociometric choice data will increase if you measure relations as a function of path distances (rather than binary choices) because relations will be created between people whose only connection is through intermediaries.

Varying Prominence

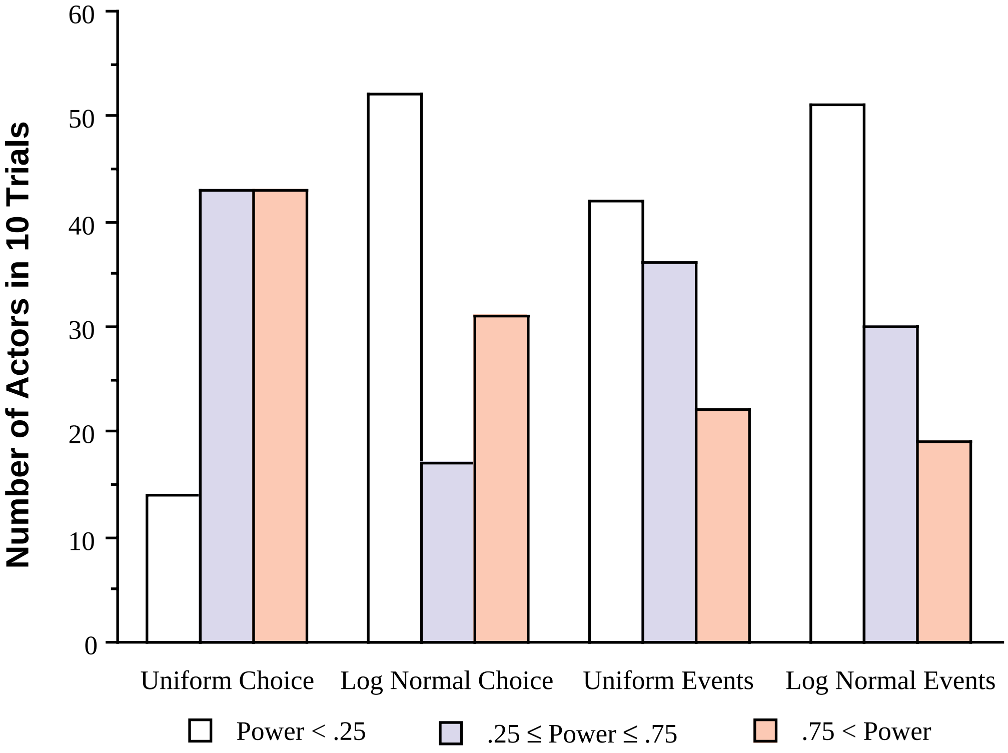
Prominence inequalities are created much as they are measured in empirical research. Prominence here refers to a variety of concepts used in network analysis to capture the vertical axis of organization; prestige, centrality, power, etc. Given a fixed number of possible choices from each person in a network, prominent leaders are identified by their tendency to receive a disproportionate share of all choices made or to appear in a disproportionate number of events. Varying prominence is useful in Monte Carlo network analysis for assessing the sensitivity of models to unequal levels of social participation within a network. Prominence is varied in STRUCTURE by using different probability distributions to generate the data. Column 12 of the MONTE command selects a probability distribution.

If column 12 is left blank, as it has in all the above examples, a uniform probability distribution is invoked. Uniform distribution random numbers are evenly distributed between 0 and 1, making all choices equally likely. If binary sociometric choice data are requested, each of a row actor's possible choices is equally likely to be chosen. If event data are requested, each event is equally likely to be chosen by each actor.

A 1 in column 12 invokes a nearly normal probability distribution. Four numbers from the uniform probability distribution are averaged to define a choice. The resulting aggregate random number, $(Z_1+Z_2+Z_3+Z_4)/4$, has a nearly normal distribution between 0 and 1 where Zi is the ith draw of a random number from a uniform distribution between 0 and 1. When this average random number is used to define binary sociometric choices, it gives people in the middle of the pool of potential choices high probability of being chosen. They then appear more prominent in the final network than the people at the two tails of the distribution. There is no meaning to the order of people in the network, so the structure can be characterized simply as one in which some people are more prominent in the sense of receiving more citations. I use a mean of only four uniform random numbers — rather than the 8 to 25 typically used to define a random variate with a normal distribution — to thicken the tails of the distribution and so decrease the number of people who receive no choices (C data) or end up involved in no events (J data).

A 2 in column 12 invokes a log normal probability distribution. Choices are based on a natural log transformation of the nearly normal variate, where Z is the above described nearly normal variate defined by averaging four random numbers from the uniform distribution, ln(Z(*e*-1)+1). The log transformation concentrates choices more than the nearly normal distribution. This in turn generates greater distance between leaders and followers.

The effect is illustrated in the results displayed on the next page for 10 trials of 10 person networks with two choices. Trials were run in each of four conditions; binary sociometric choice data drawn from a uniform probability distribution (Uniform Choice), binary choices drawn from a log normal distribution (Log Normal Choice), joint involvements in events drawn from a uniform distri-bution (Uniform Events), and event data drawn from a log normal distribution (Log Normal Events). The trials in all four conditions began with the same seed number (639). The bars show the relative prominence of actors in each condition where prominence is the eigenvector measure of power. Of the 100 actors in the first condition, 14 are on the periphery of the network (power scores below .25). This increases to 52 when choices are drawn from a log normal probability distribution. At the same time, leadership becomes more concentrated. There are 43 people with high power scores when choices are drawn from the uniform distribution. This drops to 31 people when choices are drawn from the log normal distribution. The differences are less striking in the joint involvement data, but still



noticeable. The 42 people with low power when connections between people and events are drawn from a uniform probability distribution increase to 51 when the data are drawn from a log normal distribution.

Varying Social Structure

Social structural variations are created with target image matrices. This option is only available for relations derived from binary sociometric choice data. The image matrix for a network is a binary matrix indicating where relations are strong between positions in the network. A target image matrix constrains the sociometric choice generator by limiting choices to ties in the image matrix. In the example on page 92, each person's choices are limited to other members of the chooser's clique. In the following example;

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

persons in the first position can only cite one another. Persons in the third position are constrained to avoid one another, directing all of their choices to people in the first and second positions. These target image matrices can be a powerful tool for studying the properties of network models under conditions of variably dense, variably prominent, choices within and between structurally equivalent actors.

Target image matrices are defined with the MONTE command. The number of positions in the target are defined by columns 27-28, and the subsequent columns give the number of people assigned to each position (in 15I3 format). The binary data constituting the target image matrix follow on the next line with each row of the matrix beginning on a new line. For example, here is a command file that requests 99 trials of 100 person networks where each person makes two choices from the uniform probability distribution:

```
NETWORK    100 C
MONTE         99 639 2 0 0 6 10 35 10 05 10 30
100100
010000
101000
000100
000011
101001
ANALYZE
```

The population social structure contains six positions (6 in column 28 of the MONTE command). Of 100 people to be drawn from the population, 10 are from the first position, 35 from the second, 10 from the third, 5 from the fourth, 10 from the fifth, and 30 from the sixth. The target image matrix defining connections between positions begins on the next line. Persons in the first position are allowed to cite one another and the occupants of the fourth position. Persons in the sixth position are allowed to cite one another and people in the first and third positions.

If multiple networks are generated, a target image matrix is expected for each network. For example, here is a command file requesting 99 trials of 60 persons drawn from a population defined by two networks where the first network is a clique structure and the second is a hierarchical structure:

```
NETWORK    060 C 2
MONTE         99 639 2 0 0 3 10 40 10
100
010
001
100
110
100
ANALYZE
```

The limits on density tables in STRUCTURE apply. Target image matrices are limited to 15 positions with no more than 99 occupants per position. To see the density table realized in each trial without generating reams of network data printout, put a 1 in column 26 of the MONTE command. This is useful if you want to see a summary of the relations responsible for an unusual result in a specific Monte Carlo trial. Further details on target image matrices can be found in Chapter 5 of the ASSISTANT *Command Booklet* with the general discussion of designing population structures for Monte Carlo network analysis.

NETWORK MODELS

by Thomas Schøtt, University of Pittsburgh

Analysis starts with one or more square tables of variables where each table represents a network and each variable represents a relation within a network. I presume that you have a data input file named on a DATA command and appropriate NETWORK command parameters, as described in the previous section, sufficient to define:

- a system of N actors,
- K networks of relations among the N actors,
- where the variables measuring relations within each network are comparable in metric and content, and
- each relation variable z_{ij} measures the form (the strength or intensity) of a relation from actor i to actor j.

The K square matrices of z_{ij} data defining the K networks among the N actors in a system are the beginning point for the models discussed in this section. For each STRUCTURE procedure command, I discuss the substantive purpose of the procedure, define relevant concepts and measures, describe how the default analysis can be customized with parameters on the procedure command line, describe input where special input is needed or optional, and describe printed and written output.

The wide range of analytical possibilities with STRUCTURE is much wider than the possibilities to be described in this section. I focus on alternative specifications of network models, but the results of applying these models vary enormously across alternative strategies for deriving network relations

AUTHOR's NOTE: Work on this chapter was supported by an NIMH FIRST award (1R29-MN45050). The discussion builds from the Version 3.2 *Manual*, and was improved by comments from Ron Burt, Pat Doreian and students in my seminar on Structural Sociology.

from raw network data. This is especially true of relations derived from binary sociometric choices or joint involvements. These kinds of data offer the most varied alternatives for deriving relations from raw network data. For example, substantive research and Monte Carlo analyses suggest that the decision about how to measure relations is more important for equivalence analyses than the decision between alternative structural equivalence models (Burt, 1988a; Burt and Carlton, 1989). This paragraph is merely a reminder of the potentially significant interaction between how you define relations and conclusions from a network analysis of the relations. The message: Check the robustness of analytical conclusions across alternative transformations of your raw network data.

In the following pages, I discuss network models within the five principles of network analysis outlined in the first chapter of this *Manual*. This is also the order in which analytical results are reported by the program: analyses of cohesion and clique detection (CLIQUES command), analyses of equivalence, position detection and density table images of social structure (POSITIONS command), analyses of social contagion processes (CONTAGION command), analyses of range and brokerage (AUTONOMY command), and analyses of power and equilibrium structure (POWER command).

Throughout, I refer to a focal actor as ego, the ith actor under study, and anyone connected to ego as alter, the jth actor under study. I refer to ego as she and alter as he. In the matrix of relations within a network, I follow the convention of having the row actor send relations to column actors, so the element in cell (i,j), z_{ij}, measures the strength of ego i's relation to alter j and the element in cell (j,i), z_{ji}, measures the strength of the relation ego i attracts from alter j.

COHESION

Grounded in the image of socializing relations within a primary group, the cohesion between two actors increases with the extent to which they have strong, intense relations with one another. Cohesion is a symmetric condition aggregated across two actors' relations with one another within one or more input networks. Corresponding to the concept of a primary group, a clique is then a set of actors within which cohesion is high. These ideas are modeled with the CLIQUES command.

MEASURING COHESION

The CLIQUES command gives you four options for measuring cohesion in preparation for detecting cliques.

The default is the most stringent and is the basis for defining strong component cliques (to be described in a moment). The default is to assume that cohesion can be no stronger than the weakest relation between two actors. Specifically, cohesion between actors i and j is the weakest relation between them in any network (i.e., min(z_{ij1}, z_{ji1}, z_{ij2}, z_{ji2}, ...), noting that 0 is considered stronger than a negative relation). This default measure is invoked if you leave the CLIQUES command line blank or put a 0 in column 12. Note that relations within a single network can be transformed to this cohesion measure by forcing relations to be symmetric (with a 2 in column 24 of the NETWORK command).

The second option (requested by putting a 1 in column 12) is the other extreme, and the basis for defining weak component cliques. Cohesion equals the strongest relation between two actors. Specifically, the cohesion between actors i and j is the strongest relation between them in any network (i.e., max(z_{ij1}, z_{ji1}, z_{ij2}, z_{ji2}, ...), where again it is worth noting that 0 is considered stronger than a negative relation). Again, relations within a single network can be transformed to this cohesion measure by forcing relations to be symmetric (with a 1 in column 24 of the NETWORK command).

The third option (requested by putting a 2 in column 12) is to measure cohesion as the overlap between the social circles of two actors (Alba and Kadushin, 1976). Like the preceding option, this measure generates clique memberships within weak component cliques. If i is connected with Ni alters (including herself) and j is connected with Nj alters (including himself), of whom Nij are connected to both i and j, then overlapping social circle cohesion is the ratio of mutual contacts to combined contacts, Nij/(Ni+Nj-Nij). This measure varies from 0, when i and j have no contacts in common, toward 1 to the extent their circles overlap (this ratio is also known as the Jaccard coefficient). Cohesion by overlap is set to 0 if i and j are not connected in any network.

The fourth option (requested by putting a 3 in column 12) is to read cohesion data from the input data file. This option lets you use the analytical procedures in STRUCTURE with cohesion data of any kind. The cohesion data appear in the input data file immediately following the input network data. The cohesion data are read as a symmetric matrix of real numbers preceded by an appropriate format statement. If you are not clear about format statements, see the Format Appendix in this *Manual*.

The cohesion data prepared for an analysis are not displayed. However, you can request a display of the data. The printout will include the symmetric matrix of cohesion data if you put a 1 in column 18 of the CLIQUES command. Also, you can write the cohesion data to an output data file for further analysis, either as a full matrix (1 in column 20 of the CLIQUES command), as the part of the matrix that is below the diagonal (2 in column 20), or as a dyadic file (3 in column 20) which will contain a line for each pair of actors listing title identification, identification of one actor, identification of the other actor, and the cohesion between them.

MAPPING COHESION

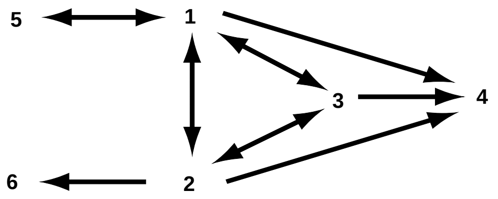
You have two ways of summarizing the cohesion data. One option is to use the cohesion data to define a roster of subgroups within which cohesion is high. I'll cover this in a moment. The other option is to build a spatial map of the data where actors are close together in the map to the extent that they are connected by cohesive ties. Edward Laumann has been especially productive with this kind of map (e.g., see Laumann, 1966; Laumann and Pappi, 1976; Heinz and Laumann, 1982; Laumann and Knoke, 1987). STRUCTURE doesn't produce maps of cohesion data (an equivalence map is provided). However, you can have the cohesion data written to an output data file for analysis with other programs. A 1 or 2 in column 20 of the CLIQUES command will generate a full matrix or lower diagonal respectively. Generate whichever is required by the program with which you wish to generate a map of the data. If you put a 4 in column 20 of the CLIQUES command, an output data file will be written named DIST.DAT. This file contains the ID CODES for each actor in the system under study and a matrix of the cohesion data. You can analyze the DIST.DAT file directly with the DIM routine in the UCINET network analysis package available from Lin Freeman at the University of California, Irvine. The DIST.DAT file is in the format of an input file for UCINET. To generate a multidimensional scaling of the cohesion data, type DIM to load the dimensional analysis routines in UCINET, press M for the multidimensional scaling option, then enter the disk drive and DIST file name (e.g., F:DIST) to read the DIST.DAT file, then take all the default options for scaling. The DIM program will produce a multidimensional scaling of the cohesion data and give you the option of saving the map coordinates to disk so that you can create a high quality picture of the map with a graphics program.

DETECTING CLIQUES

Several ideal type kinds of cliques are offered by the mathematical theory of graphs (Alba, 1973; Harary et al., 1965). Commonly the criteria focus on intragroup relations and tend to disregard extragroup ties. The cliques detected by STRUCTURE differ in two respects, by directness of the linkage between clique members and by mutuality of the linkage between clique members. In the first respect, some clique concepts require direct ties in all pairs of members of a clique while other clique concepts consider indirect linkage between the actors sufficient for co-membership of a clique. In the second respect, some clique concepts require mutual reachability in the link between clique members thus emphasizing direction of relations while other clique concepts disregard direction and consider contact between the actors sufficient for co-membership of a clique. These two conceptual distinctions yield the fourfold typology of cliques shown below.

	Direct relation required	Indirect tie sufficient
Mutual reachability Required	Reciprocating completely connected clique	Strongly connected clique
Contact sufficient	Completely connected clique	Weakly connected clique

By this typology each kind of cliques is defined as follows, using graph theoretical terms. A ‘reciprocating completely connected clique’ is a set of actors where each member is directly and reciprocally tied to every other member (graph theoretically described as a completely connected subgraph with symmetric lines; Harary et al., 1965:7). This definition is restrictive regarding both mutuality and directness; relaxing one or both criteria yields the three other kinds of cliques. Maintaining the requirement of direct ties but



relaxing the requirement of mutuality, a 'completely connected clique' is a set of actors where each member has a direct relation to or from every other member (a completely connected subgraph, Harary et al., 1965:7). Maintaining some mutuality but instead relaxing the requirement of a direct relation to include actors mutually reachable through intermediaries, a 'strongly connected clique' is a set of actors where each member can reach each other member directly or via other members through a directed path of relations (strongly connected subgraph, Harary et al., 1965:51). Fourth, relaxing both the requirements of directness and mutual reachability to allow for any joining, a 'weakly connected clique' is a set of actors where any two members can be joined by a direct tie between them or via other members through a sequence of ties disregarding direction of every relation (a weak subgraph, Harary et al., 1965:51). Furthermore, each kind of clique is called maximal if no other actors can be added to the clique without losing its defining property. Notably, a weakly connected clique is maximal if no member has any relation to or from any outsider.

I'll use the above hypothetical network to illustrate clique detection. The network illustrates how the kinds of cliques I have distinguished are partially ordered by the partly successive relaxation of their defining criteria. Actors 1, 2 and 3 are a reciprocating completely connected clique, actors 1, 2, 3 and 4 are a completely connected clique, actors 1, 2, 3 and 5 are a strongly connected clique, and the whole system is a weakly connected clique. In general, any reciprocating completely connected clique is a completely connected clique and is also a strongly connected clique; furthermore, any completely connected clique and any strongly connected clique is a weakly connected clique. Typically, a sociometric choice network has mostly very small reciprocating completely connected cliques; conversely, it is likely to have very large but very few weakly connected cliques. In particular, if the system is obtained by snowballing, it will just be one weakly connected clique. For these reasons, one ideal type grouping may have a too restrictive definition of a clique while another graph theoretic principle may embody too lax a criterion. Whichever definition of a clique is chosen, there is no statistical principle for ascertaining adequacy of the detected cliques (this is unlike detecting positions on the criterion of equivalence where adequacy of a grouping can be tested as discussed in the next section).

The kind of clique detected by STRUCTURE depends on how relations are measured from raw data and how cohesion is defined from relations. If cohesion is defined by the minimum relation across networks, then a clustering procedure is invoked which requires cohesion in each pair of members of a cluster and therefore will detect reciprocating completely connected cliques (default blank or 0 in column 12 of the CLIQUES command). If, furthermore, relations between two actors have been symmetrized in each network as their more intense raw relation (by putting 1 in column 24 in the NETWORK command) then direction is disregarded; unidirectional contact in each network is sufficient to yield a relationship measure that is positive and symmetric in each network and therefore yield positive cohesion such that the same clustering requiring cohesion in each pair will detect groups that are completely connected cliques.

Alternatively, if cohesion between actors is defined as the maximum relation between them across all networks (thus considering a unidirectional uniplex relation sufficient for cohesion), then a clustering procedure is invoked which considers indirect linkage between actors sufficient for co-membership in a clique and therefore will detect weakly connected cliques (1 in column 12 of the CLIQUES command). If, furthermore, relations between two actors have been symmetrized within each network as their less intense raw relation (by putting 2 in column 24 in the NETWORK command) then reciprocated raw relations in at least one network are required to yield a relationship measure that is symmetric and positive in at least one network and therefore a positive cohesion tie such that the clustering allowing for indirect links will detect groups that actually are strongly connected cliques.

Before illustrating detection of cliques, the automechanics of the clusterings shall be described more precisely. A clique is operationally defined as an aggregate of actors clustered on a criterion of cohesion and cliques are detected by an algorithm applied to the matrix of cohesion among actors. Actors are hierarchically agglomerated (Everitt, 1974). Initially each actor is a cluster by itself. Two most cohesively related actors are merged to form a new cluster replacing the two initial ones. Merging of two most proximate clusters continues sequentially. The history of successive mergers is a hierarchy of clusters, each stage is a set of clusters where a cluster is a fusion of earlier clusters and is nested within clusters at subsequent agglomeration. Note that at later stages of agglomeration, cohesion within clusters tends to be lower.

The clustering procedure invoked by defining cohesion between actors by their minimum strength relation across all networks (or by symmetric cohesion data read from an input file), detects reciprocating completely connected cliques as follows. The first step clusters together two actors that are most proximate, i.e. most cohesive in the system. Proximity between an actor and this cluster is then defined as the minimum cohesion between the actor and the actors in the cluster, thus defining a measure of proximity of any unit to any other unit (cluster or actor). Again, as the second step, two most proximate units are merged. At this new level, proximity of one unit to another is again defined as the minimum cohesion obtaining between one unit's actors and the other unit's actors. Again, as the third agglomeration, two most proximate units are clustered, and so forth. In each step the criterion for merging is the greatest proximity between clusters, a criterion level that is less than or equal to the minimum cohesion within the preexisting clusters and in that step becomes the new minimum cohesion within the new clusters. Each level in the hierarchy thus corresponds to a minimum cohesion obtaining within clusters, and as the agglomeration proceeds, the minimum cohesion within clusters decreases. As long as this minimum cohesion remains positive, every cluster is a reciprocating completely connected clique. One level of agglomeration is special, namely that at the smallest positive minimum cohesion where each cluster is a still a reciprocating completely connected clique, a property that would be lost by any further merger of clusters. At this level the clustering ends. At each level, a clique is reciprocally and completely connected within each network but it is possibly not maximal.

The same clustering procedure will detect completely connected cliques (possibly without reciprocity) if relations are first symmetrized. More precisely, if relations between two actors are symmetrized within each network as their more intensive raw relation and cohesion is defined as above, i.e. the minimum strength of the transformed relations across all networks, then the same clustering procedure is invoked and it will detect groups that are completely connected cliques but relations within a clique are not necessarily reciprocal. At any level of the clustering, such a clique is completely connected within each network but is possibly not reciprocating and possibly not maximal.

The clustering procedure invoked by defining cohesion between actors by their maximum strength relation across all networks or by their overlapping social circles (both allowing unreciprocated and uniplex relations as basis for cohesion) detects weakly connected cliques as follows. The first step merges two most proximate or cohesive actors. Proximity of an actor to this cluster is now defined as the maximum cohesion between the actor and the actors in the cluster. The second step again merges two most proximate units (actors or clusters). Proximity between two units is again defined as the maximum cohesion obtaining between one unit's actors and the other unit's actors. The two most proximate units are again merged, and so on. In each step the criterion for merging is the greatest proximity between clusters. This criterion level of the strongest cohesion tie between the two old clusters merged into a new cluster actually is the weakest link in a chain that an actor in one of the old clusters has to go through to join an actor in the other old cluster. It ignores weak and absent ties and emphasizes cohesive ties linking actors in a cluster. More precisely, the criterion level is the strength of the weakest chain between actors within a cluster, where a chain between actors is a sequence of ties connecting them which is of greatest strength as measured by the weakest cohesion tie in the sequence (intuitively, a chain is a path that best avoids any weak cohesion ties). As the agglomeration proceeds the criterion level decreases, but at any positive criterion level every cluster is a weakly connected clique. More precisely, if several networks are input, a cluster may possibly not be a weakly connected clique within each network, but when contents of relations are disregarded and the combined network is considered then the cluster is a weakly connected clique. Also at the smallest positive criterion level in the hierarchy, every cluster is a weakly connected clique, but any further merger would jeopardize that property. At this level the clustering ends. Actually, each of these last ones are maximal weakly connected cliques.

This clustering procedure will detect strongly connected cliques if relations between two actors are symmetrized as the least intensive within each network and if cohesion is defined by the greatest of these transformed relations across all networks (cohesion thus requires a reciprocated relation in at least one network). The invoked clustering allows for indirect cohesion links and the detected groups will be strongly connected cliques. More precisely, if several networks are input, a group may possibly not be strongly connected within each network (i.e., a member may not be able to reach any other co-member by a path of only one kind of relation), but when contents of relations are disregarded and the networks are combined then each cluster is a strongly connected clique. At any level such a strongly connected clique is possibly not maximal.

CLUSTER ANALYSIS PRINTOUT

Clustering is printed out in two ways, as a tree diagram and as a landscape diagram. The tree may offer the best first overview but may be somewhat compressed and the landscape diagram adds detail. Note that the 132 column tree diagram is twice as precise as the 80 column diagram and the graphics diagram is much easier to read than the typewriter diagram (column 16 of the CLIQUES command).* The tree and landscape diagrams display the history of the agglomeration process from the extreme of each actor being a cluster by itself to the other extreme of maximum aggregation. Between these extremes, a cut across the diagram shows the clusters at a level of agglomeration determined by the criterion level of cohesion, the cut shows the number of clusters and the actors in each cluster. For your convenience, the printout also lists the actors in each clique at the lowest positive criterion level. These cliques may prove suitable for reporting. Often, though, these final cliques are extremely large, so that this level is not informative. Among the levels of agglomeration, it is sometimes recommended to focus on the level just before the biggest drop occurs in criterion level; these sequential criteria are printed alongside the landscape diagram.

*The graphics diagram is in the printout file and also in a separate file named _TREE.PRT that contains only the graphics diagram. The separate file is provided so that you can more easily print only the tree diagram.

For example, consider again the system of six actors used earlier to illustrate the four clique concepts. Raw relations among the actors are given in the input file listed to the right.

```
(6F2.0)
- 6 5 9 4 0
7 - 5 2 0 5
4 6 - 3 0 0
0 0 0 - 0 0
8 0 0 0 - 0
0 0 0 0 0 -
```

First, letting cohesion be the default without symmetrizing raw relations, defines cohesion between actors by their minimum strength relation and generates the following cohesion matrix and clustering printed out as a landscape,

```
                             MINIMUM
- 6 4 0 4 0                 COHESION
6 - 5 0 0 0               IN CLUSTER    1 2 3 4 5 6
4 5 - 0 0 0              ---------------------------
0 0 0 - 0 0                  6.000      XXX . . . .
4 0 0 0 - 0                  4.000      XXXXX . . .
0 0 0 0 0 -                   .000      -----------
                                         END OF CLUSTERS
```

In this clustering, actors 1 and 2 are merged first since they are most cohesively linked and the level of cohesion within this cluster is 6. As the second step, actor 3 is added, and the level of cohesion within this cluster drops to 4, the minimum cohesion between a pair of members of the cluster. No other actor is cohesively tied to every members of this cluster and not cohesively tied to any other actor so clustering ends at this level. The clusters detected at the two levels of aggregation are reciprocating completely connected cliques.

Second, letting cohesion again be the default (the minimum strength relation) but measuring relations between two actors by their most intensive raw relation, generates the following cohesion matrix and landscape output,

```
                              MINIMUM
 - 7 5 9 8 0                 COHESION
 7 - 6 2 0 5               IN CLUSTER    2 3 1 4 5 6
 5 6 - 3 0 0              __________________________
 9 2 3 - 0 0                 9.000       . . XXX . .
 8 0 0 0 - 0                 6.000       XXX XXX . .
 0 5 0 0 0 -                 2.000       XXXXXXX . .
                              .000       ___________
                                          END OF CLUSTERS
```

This clustering differs from the first clustering because relations have been transformed. The clustering principle remains the same. The symmetrization made actors 1 and 4 most cohesive, so they are clustered first, with a cohesion of 9 between them. No other actor is as cohesively tied with both 1 and 4 as actors 2 and 3 are tied to one another so these get merged next. Neither actor 5 nor actor 6 is cohesively tied to both members of the first cluster or to both members of the second cluster or to one another, but actors 1, 2, 3 and 4 are all cohesively tied to one another so they all cluster as the third and last step. At each level of agglomeration, every detected cluster is a completely connected clique.

Third, let relations between two actors be measured by their least intensive raw relation and now define cohesion between them as their maximum strength transformed relation; this generates the following cohesion matrix and clustering output,

```
                              MINIMUM
 - 6 4 0 4 0                 COHESION
 6 - 5 0 0 0               IN CLUSTER    4 1 2 3 5 6
 4 5 - 0 0 0              __________________________
 0 0 0 - 0 0                 6.000       . XXX . . .
 4 0 0 0 - 0                 5.000       . XXXXX . .
 0 0 0 0 0 -                 4.000       . XXXXXXX .
                              .000       ___________
                                          END OF CLUSTERS
```

Cohesion is here the same as in the first cohesion matrix (they would be likely to differ if several networks were input). However, the clustering principle differs and leads to a clustering output different from the first landscape. Actors 1 and 2 are most cohesive and so are clustered first (just like in the first landscape). The next greatest cohesion tie in the matrix is here seen to be 5 and obtains between actors 3 and 2 so this tie adds actor 3 to the cluster containing actor 2 (like in the first landscape). The next highest cohesion tie is 4 which obtains between actors 1 and 3 who are already merged and also between actors 1 and 5 so actor 5 is added to the cluster containing 1 (unlike the first landscape where the restrictive clustering principle prohibited any further merger). Actors 4 and 6 have no cohesion with one another or with anyone in the cluster so the clustering ends here. At each level, every detected group is a strongly connected clique.

Fourth, define cohesion between actors by their maximum strength relation, without first symmetrizing relations. This generates the following cohesion matrix and the landscape printout,

```
                                MINIMUM
 - 7 5 9 8 0                   COHESION
 7 - 6 2 0 5                  IN CLUSTER    3 2 1 4 5 6
 5 6 - 3 0 0                 ___________________________
 9 2 3 - 0 0                    9.000       . . XXX . .
 8 0 0 0 - 0                    8.000       . . XXXXX .
 0 5 0 0 0 -                    7.000       . XXXXXXX .
                                6.000       XXXXXXXXX .
                                5.000       XXXXXXXXXXX
                                            ___________
```

This cohesion matrix differs from the third cohesion matrix because of the transformation of relations in the third case. The clustering principle remains the same. Actors 1 and 4 are merged first by their cohesion tie of 9. The next greatest cohesion tie in the matrix is seen to be the tie of 8 between actors 1 and 5 so actor 5 is added to the cluster. The next greatest cohesion tie is 7 obtaining between actors 1 and 2 so actor 2 is added to the cluster. Etc. At each level of agglomeration every cluster is a weakly connected clique.

These four clusterings, based on the same matrix of raw data, emphasize that the detected cliques depend greatly on how relations are measured and how cohesion and cliques are operationalized from relations.

Clique as a Special Type of Position

Finally, I wish to compare cliques to the concept of a position jointly occupied by structurally equivalent actors (cf. Burt 1983a). Two members of an ideal-typical clique both have cohesive ties with other members of the clique and both have weak ties with outsiders. Thus they are structurally equivalent. Further, an actor who is structurally equivalent with members of a clique has strong ties with the clique members and weak ties with others in the system, and therefore qualifies as a member of the clique. These properties imply that a clique is actually a position jointly occupied by structurally equivalent actors. The opposite is not true, two equivalent actors are not necessarily cohesively tied, so a position is not necessarily a clique. Combining these two considerations of the two concepts as ideal types leads to the conclusion that a clique is a special kind of position. Aggregation based on structural equivalence will therefore tend also to detect cliques. In a particular system, there may be little, moderate, considerable or even perfect overlap between positions and cliques. This is exemplified at the end of the next section on equivalence.

This is consequential for assessing substantive results based on analyses of cliques. If homogeneity is found within cliques it may be due to the cohesion or due to the structural equivalence among clique members, two empirically confounded but analytically distinct conditions expectedly generating quite different influence processes. A substantive interpretation based solely on clique results may therefore quite possibly be erroneous. A test of the two alternative explanations can be based on data on positions, if homogeneity obtains also within non-cohesive positions that can be attributed to equivalence and not to cohesion, but if heterogeneity obtains within non-cohesive positions, then only cohesion is effective. If not only group-level data but actor-level data are available, cohesion and equivalence effects can be estimated using the CONTAGION command.

EQUIVALENCE

Grounded in the image of status-role/sets within a social structure, equivalence between two actors increases with the extent to which they have identical patterns of relations within the social structure. Actors with identical profiles of relations are equivalent and make up a social category, a group termed a position jointly occupied by equivalent actors. Their pattern of relations, abstracted from the actors in the position, defines a status with a role-set, a set of relations with other statuses. The system can be described as interrelated positions or, at a higher level of abstraction, as role-sets interlocking statuses, as is the classical conception of social structure (Linton, 1936; Merton, 1957; Nadel, 1957). These ideas are modeled with the POSITIONS command.

MEASURING EQUIVALENCE

Lorrain and White's (1971) structural equivalence concept came to operational fruition in the mid 1970s with a flurry of articles proposing categorical models of structural equivalence, termed blockmodels (e.g., Breiger et al., 1975; White et al., 1976; Arabie and Boorman, 1982), and continuous distance models (e.g., Burt, 1976, 1977, 1982). Two actors are structurally equivalent to the extent that they have identical relations with every person in every network within a social structure. Structural equivalence remains the workhorse concept guiding substantive network studies of the status/role-sets defined by social structure. A role, however, is not defined by relations with specific individuals; rather, abstracting from actors, it is defined by relations with other roles. Structural equivalence models therefore drew criticism almost as soon that they began to appear in substantive analyses. More abstract equivalence models — models of role equivalence — were proposed as better substantive representations of roles (Sailer, 1978; White and Reitz, 1983; Winship, 1988; Winship and Mandel, 1983). Distinctions between alternative role equivalence models are still being worked out. Where cohesion was the methodological frontier of network analysis through the 1950s and 1960s, and structural equivalence the frontier during the 1970s, role equivalence was the frontier through the 1980s. While the algebra of equivalence models continues to pepper *Social Networks*, Hummell and Sodeur developed an intuitively and computationally simple measure in the mid 1980s that is the basis for role equivalence in STRUCTURE (Hummell and Sodeur, 1987; Burt, 1990a). Instead of profiling ego's relations with specific others, role equivalence focuses on the pattern of triads in which she is involved. Characterizing a triad of ego and two alters by ego's relations to and from the alters and relations between the alters yields a typology of triads, and ego's role refers to her involvement in each type of triad. Two actors are role equivalent to the extent they are identically involved in each triad type. Role equivalence broadens the concept of structural equivalence in that structurally equivalent actors are role equivalent but role equivalent actors are not necessarily structurally equivalent. By either equivalence concept, relation patterns are cast as data profiles and equivalence is measured by comparing two actors's profiles. A Euclidean distance is computed where d_{ij} equals zero if actors i and j have identical relation patterns, and so are equivalent. This is a standard data analysis problem — with Cronbach and Gleser's (1953) review article still providing the classic statement of the profile level, scatter, and shape components in measures of profile similarity. Distance is defined below for one network. The extension to multiple network structures is routine. With K input networks, the program squares the distance d_{ij} between i and j within each network, sums the K squared distances, and reports the square root of the sum as the aggregate measure of equivalence between i and j across all K networks. The POSITIONS command gives you seven equivalence options, options ranging from the strictest measures of structural equivalence to highly flexible measures of abstract role equivalence.

The first two options — a blank, 0, or 1 in column 12 of the POSITIONS command — are strict measures of structural equivalence. Two individuals are structurally equivalent in a network to the extent that they have identical relations with every individual in the network. The pattern, or profile, of relations defining an individual j's network position can be arranged in a vector Z_j of 2N relation variables; N variables measuring j's relations to others (z_{jl}, z_{j2}, ... z_{jN}) and N variables measuring relations received from others (z_{1j}, z_{2j}, ... z_{Nj}). The extent to which two individuals i and j are involved in identical relations so as to be structurally equivalent can be expressed as the Euclidean distance, d_{ij}, between their relation patterns:

$$d_{ij} = [\ (z_{ij}-z_{ji})^2 + \Sigma_q(z_{iq}-z_{jq})^2 + \Sigma_q(z_{qi}-z_{qj})^2\ ]^{1/2}, \quad q \neq i,j$$

where a distance of zero indicates completely equivalent patterns and increasing values of d_{ij} indicate increasingly equivalent patterns. If self-relations are included in the calculation, the Euclidean distance is defined without a special term for relations between actors i and j:

$$d_{ij} = [\ \Sigma_q(z_{iq}-z_{jq})^2 + \Sigma_q(z_{qi}-z_{qj})^2\ ]^{1/2},$$

where the choice between including or excluding self-relations is under your control with column 22 of the NETWORK command. In matrix notation, the Euclidean distance can be written as:

$$d_{ij} = (\ [(\mathbf{Z}_j-\mathbf{Z}_i)'\ (\mathbf{Z}_j-\mathbf{Z}_i)]\ )^{1/2},$$

where ′ indicates transpose (cf. Cronbach and Gleser, 1953:459). The above distance between raw relation patterns is the default measure of structural equivalence in STRUCTURE (blank or 0 in column 12 of the POSITIONS command). Dividing by the number of relations compared yields the root mean squared difference, the average difference in someone's relation with i versus his corresponding relation with j (these are the Euclidean distances generated by SYSTAT):

$$(\ [(\mathbf{Z}_j-\mathbf{Z}_i)'\ (\mathbf{Z}_j-\mathbf{Z}_i)]\ /\ [2N]\ )^{1/2},$$

which is merely the raw Euclidean distance divided by a constant, the square root of 2N. This mean distance is requested by putting a 1 in column 12 of the POSITIONS command. The decision to use one or the other of these distance measures depends on the weight to be given to absent relations. The raw Euclidean distance measure ignores all relations beyond those involving i or j. The mean Euclidean distance averages differences between existing relations across all possible relations. Adding isolates to a network, for example, will decrease mean Euclidean distances but have no effect on raw Euclidean distances. This is a minor issue, of interest perhaps where metric distances will be compared across networks of different sizes.

The mean difference between patterns — i.e., the difference in interaction "level" between two patterns — is a more serious concern. The next equivalence option (requested by putting a 2 in column 12 of the POSITIONS command) removes mean differences. For example, input–output tables define network relations between economic sectors. Relations are measured as the dollars of commodity sold by establishments in the row sector to establishments in the column sector. Economic and sociological theory about the structure of production relations, however, is not concerned with dollars of sales so much as they are concerned with the relative strength of relations. The raw input–output table data are usually divided by the column marginals of the table to produce network relations measuring the proportion of input to each column sector that is purchased from each row sector. In sociometric data, individuals can differ in their average tendencies to be involved in relations as a function of response bias or inaccurately measured relations, some seeing themselves as very active socially and being often cited while others report few contacts and themselves escape notice in other's citations. To control the effect of such differences on structural equivalence measures, the "level" component in relation patterns can be removed by subtracting out the mean strength of an individual's relations. With a 2 in column 12 of the POSITIONS command, distance will be the root mean squared difference between relations that have been adjusted for means (cf. Cronbach and Gleser, 1953:460):

$$([(\mathrm{Z}_j-\mathrm{Z}_i)' \ (\mathrm{Z}_j-\mathrm{Z}_i)] \, / \, [2N] \,)^{1/2},$$

where,

$$Z_j = (\mathbf{Z}_j - \mathbb{Z}_j) = \{\mathbf{z}_{jk} - \mathbb{z}_j\},$$

and $\mathbb{Z}_j$ is a 2N vector of elements each equal to the mean, $\mathbb{z}_j$, of all relations involving person j (the mean for all elements in $\mathbf{Z}_j$). Equivalence is sometimes measured with correlations, so I want to clarify the connection with Euclidean distance. The above distance can be restated as follows in terms of the variances of i's and j's relations and the covariance between their relations:

$$(s_i^2 + s_j^2 - 2s_{ij})^{1/2},$$

where s_j is the standard deviation of j's relations (the standard deviation of all elements in $\mathbf{Z}_j$, i.e., $s_j^2 = (\mathbf{Z}_j - \mathbb{Z}_j)' \ (\mathbf{Z}_j - \mathbb{Z}_j)/2N$) and s_{ij} is the covariance between j's and i's relations.

Scatter is another consideration. The next equivalence option (requested by putting a 3 in column 12 of the POSITIONS command) removes mean and variance differences between relation profiles. Again, as a function of response bias or inaccurately measured relations, individuals can differ in the amplitude of their relationships, some individuals reporting relations that range from intensely close to intensely hostile while other individuals report little variation between their relationships. To control the effect of such differences on structural equivalence measures, the "scatter" component in relation patterns can be removed by dividing each relation by the standard deviation of an individual's relations. With a 3 in column 12 of the POSITIONS command, distance will be the root mean squared difference between relations adjusted for means and standard deviations:

$$([(\mathrm{Z}_j-\mathrm{Z}_i)' \ (\mathrm{Z}_j-\mathrm{Z}_i)] \, / \, [2N] \,)^{1/2},$$

where,

$$\mathrm{Z}_j = (\mathbf{Z}_j - \mathbb{Z}_j)/s_j = \{(\mathbf{z}_{jk} - \mathbb{z}_j)/s_j\}.$$

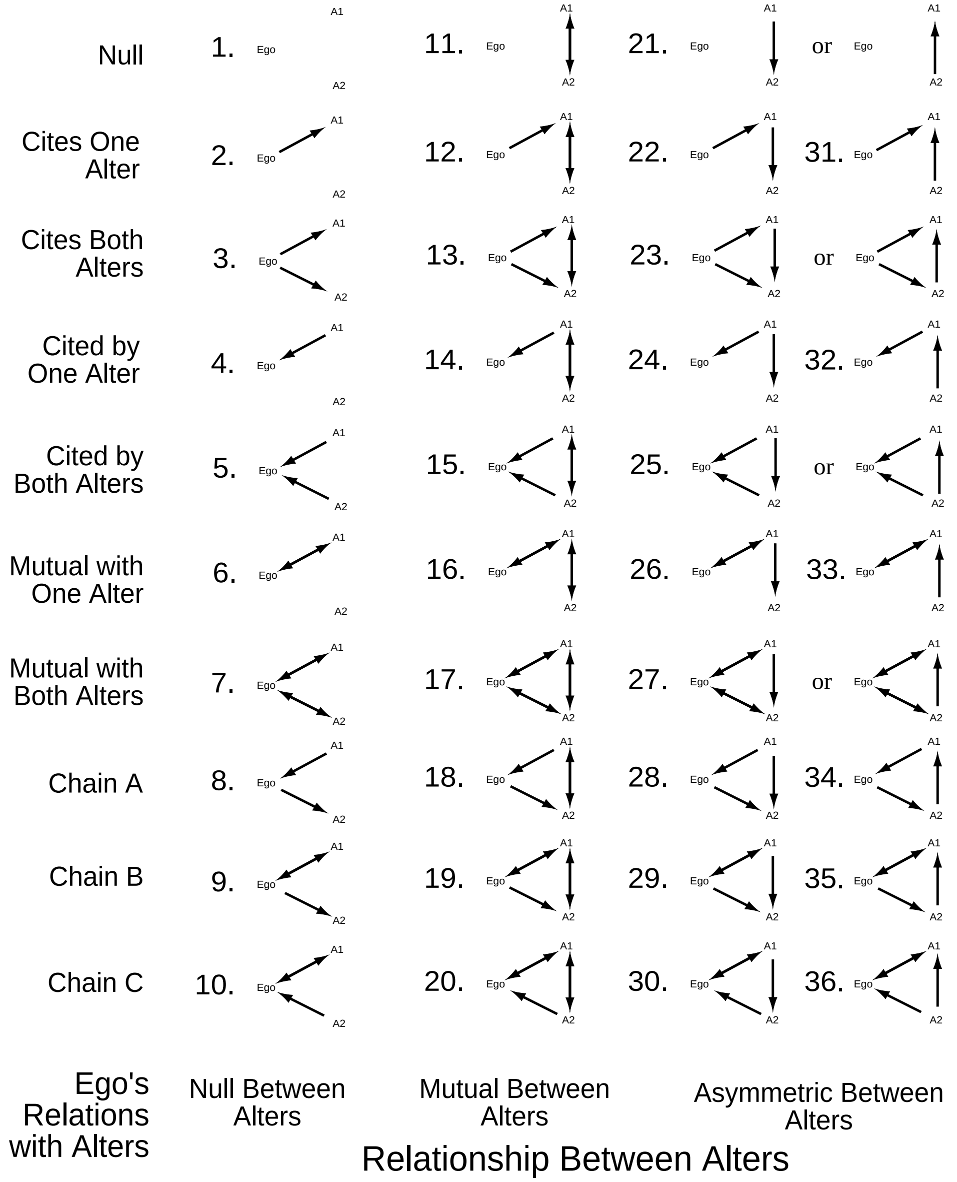
These distances could be computed as follows from the correlation (r_{ij}) between j’s and i’s relation patterns (cf. Cronbach and Gleser, 1953:461):

$$[\ 2(1 - r_{ij})\ ]^{1/2},$$

which makes it clear that the correlation between two relation patterns (the measure of structural equivalence in CONCOR) is inversely proportional to the Euclidean distance between the two patterns stripped of their means and standard deviations. This is an important link between two ostensibly different pattern similarity measures often contrasted in debate over structural equivalence measures. The same structural equivalence results would be obtained from the clustering or multidimensional scaling algorithms applied to the z-score distance or the correlation measures of equivalence.

Now to role equivalence. The next equivalence option (requested by putting a 4 in column 12 of the POSITIONS command) invokes Hummel and Sodeur's (1987) role equivalence model. This model is very similar, but not identical, to Winship and Mandel's (1983) model of role equivalence defined by direct and two-step ties. The analogy is easily made exact, and is provided as the next equivalence option (requested by putting a 5 in column 12). By casting an individual's role as a pattern of triads in which the individual is involved, Hummel and Sodeur offer a Euclidean distance measure of equivalence that generalizes equivalence beyond having identical relations with specific individuals.

In a network of N actors each ego is involved in (N-1)(N-2)/2 triads. These triad patterns only consider present versus absent relations, so nonbinary input networks are dichotomized by setting any nonzero relation to 1. With 36 triad types listed on the next page and included in STRUCTURE printout (by putting a 1 in column 20 of the POSITIONS command), actor i’s role is given by frequencies (t_{i1}, t_{i2}, ... , $t_{i,36}$) where t_{iq} is the number of triads of type q in which actor i plays the role of ego. The important point in this table is that


Null
1.
11.
21.
or
Cites One Alter
2.
12.
22.
31.
Cites Both Alters
3.
13.
23.
or
Cited by One Alter
4.
14.
24.
32.
Cited by Both Alters
5.
15.
25.
or
Mutual with One Alter
6.
16.
26.
33.
Mutual with Both Alters
7.
17.
27.
or
Chain A
8.
18.
28.
34.
Chain B
9.
19.
29.
35.
Chain C
10.
20.
30.
36.
Ego
A1
A2
Ego's Relations with Alters
Null Between Alters
Mutual Between Alters
Asymmetric Between Alters
Relationship Between Alters

the identities of the alters are defined by their relations with each other and ego. They are no longer individuals who receive and send relations. Each is now a kind of intersection between two relationships, a relationship with ego and a relationship with the other alter. The triads of these relationship combinations are role components. The network structure of a role has an identifiable signature as a pattern of these component microstructural orientations toward others. The table is an inventory of role components, and the relative frequency with a person as ego plays each of the triad types within a network defines the person's role in the network. Two individuals i and j play the same role to the extent that they are identically involved in the role components tabulated in their respective triad patterns (i.e., t_{iq} equals t_{jq} for all triad types q). This is measured by the Euclidean distance between i's and j's respective triad patterns:

$$d_{ij} = [\ \Sigma_q\ (t_{iq}-t_{jq})^2\ ]^{1/2},$$

where summation is across the 36 triad types q on page 130 and equivalence declines with increasing distance between i and j. This measure of role equivalence uses the raw triad frequencies and so is affected by the volume, or scale, of a role.* It is a scale-sensitive measure of role equivalence. A role played out with respect to many people will not be equivalent to the same role played out with respect to few people. The leader of a small group, for example, will not be equivalent to the leader of a large group. In this, the measure deviates from Winship and Mandel's concept of role equivalence,

*This is not exactly correct. Raw triad frequencies are converted to percentages before distances are computed. This has no effect on the relative magnitude of distances between actors because each triad profile is divided by the same constant (i.e., the total number of triads possible for any actor in the network). The conversion to percentages is made because triad counts quickly become enormous numbers, which create enormous distances, both of which are cumbersome to display in the printout. However, when triad counts are written to an output data file for analysis outside STRUCTURE, the percentages are converted back to raw frequencies.

which is invariant to the number of relations defining a role. The analogy to Winship and Mandel one and two-step equivalence is exact if the triad pattern frequencies are converted to binary data before distances are computed (i.e., t_{iq} in the above equation is set to 1.0 for all frequencies greater than zero, see Burt, 1990a:13-14). This scale-invariant measure of role equivalence is requested by putting a 5 in column 12 of the POSITIONS command.

The next equivalence option gives you more control over the kinds of roles in terms of which equivalence is measured. When a 6 is put in column 12 of the POSITIONS command, the program reads a vector of 36 triad weights from the next line in the command file. The weights are read in 36I2 format (see page 42 for examples). Where w_q is the weight for triad type q, the role equivalence between actors i and j is measured as follows:

$$d_{ij} = ([\Sigma_q w_q(t_{iq}-t_{jq})^2] / [\Sigma_q w_q])^{1/2},$$

where dividing by the sum of the input weights converts them to proportions to measure the relative weighting of triads. At minimum, this is useful for eliminating the null triad (type 1 on page 130) from equivalence definitions in sparse networks. More generally, this weighting option is a powerful tool for exploring the role components responsible for equivalence (e.g. brokerage as on page 42 and described ahead in the Autonomy section).

If the above six equivalence options are not enough, the seventh option is to read in equivalence distances defined elsewhere. When a 7 is put in column 12 of the POSITIONS command, the program reads a format statement in the line immediately following the network data in the input data file. The next lines of the input data file should contain a square distance matrix in the specified format with each row of the matrix beginning on a new line.

Very different equivalence results can be obtained with the six equivalence measures. The verb should be emphasized; different results can be obtained, but different results will not always be obtained. The stronger the similarity between two relation patterns, the less it matters which of the alternatives is used to measure structural equivalence. Arguments for measuring structural equivalence as similarity between raw relation patterns do not rule out the value of holding level and scatter constant in some data sets. Rather, they inveigh against the indiscriminate use of such controls because relation pattern level and scatter can be significant features of social structure in a study population (e.g., Burt, 1982:47–48; 1986; Burt and Minor, 1983:274–277). In addition, Burt (1988a) and Burt and Carlton (1989) discuss conditions in which alternative structural equivalence measures differ. Burt (1976, 1987) discusses conditions in which cohesion and structural equivalence are similar and conditions in which they are different. Burt (1990a) provides a review with illustrations of conditions in which role equivalence differs from structural equivalence.

The alternative pattern similarity measures are all readily available distance options in STRUCTURE. However, in large part because of default options in readily available computer programs, structural equivalence is most often measured as similarity in all aspects of relation pattern (e.g. Euclidean distance between raw relation patterns in STRUCTURE) or with level and scatter held constant with relation patterns (e.g. correlation between relation patterns in CONCOR corresponding to Euclidean distance between standardized relation patterns in STRUCTURE). Comparisons across studies have been further made difficult by specific pattern similarity measures being used typically with a single kind of relation measure, CONCOR being used typically to measure structural equivalence from z-score patterns of binary network relations and STRUCTURE being used typically to measure structural equivalence from raw patterns of network relations based on path distances normalized by the frequency decay function described in the earlier chapter on Network Data. In the absence of substantive reasons for preferring one measure over others it seems advisable to check and report robustness of results over the alternative equivalence measures. And even if one measure is a priori preferred for substantive reasons, it seems worthwhile knowing and reporting whether other measures yield similar results.

The distance data prepared for an equivalence analysis are not displayed. However, you can request a display of the data. The printout will include the symmetric matrix of equivalence data (and triad patterns for role equivalence data) if you put a 1 in column 20 of the POSITIONS command. Also, you can write the data to an output data file for further analysis, either as a full matrix (1 in column 22 of the POSITIONS command), as the part of the matrix that is below the diagonal (2 in column 22), as a dyadic file (3 in column 22) which will contain a line for each pair of actors, or, by putting a 4 in column 22, you can get the triad patterns used to define role equivalence distances (with each actor's triad pattern on a new line).

MAPPING EQUIVALENCE

As with cohesion data, you have two ways of summarizing the equivalence data. One option is to use the data to define a roster of subgroups within which equivalence is high. I'll cover this in a moment. The other option is to build a topological map in which equivalent actors are close together (e.g., Burt, 1976, 1988b; Burt and Carlton, 1989).

You have two options for building such maps. The simplest is to use the map automatically printed by STRUCTURE for equivalence analyses. The map is a eigenvector representation of the equivalence data. The eigenvector corresponding to the largest eigenvalue of the matrix defines the horizontal axis of the STRUCTURE map and the eigenvector for the second largest eigenvalue defines the vertical axis. The orientation of a map is immaterial, only relative location matters. The map corresponds to an unrotated principal component factor analysis of the distance data. The parameter in column 18 of the POSITIONS command gives you control over the format of the map to get the most out of your printer. The STRUCTURE spatial display is a very close fit to the observed distances. Outliers — actors especially nonequivalent to any other actor in the system — can severely distort the map. You can delete the outlier actors and re-run the analysis.

Alternatively, you can write the equivalence distance data to an output file for analysis with other programs. A 1 or 2 in column 22 of the POSITIONS command will generate a full matrix or lower diagonal matrix respectively. Generate which ever is required by the program with which you wish to analyze the distance data. Alternatively, a 4 in column 22 will generate an output file named DIST.DAT. This file contains the ID CODES for each actor in the system under study and a matrix of the distance data. You can analyze the DIST.DAT file directly with the DIM routine in the UCINET network analysis package available from Lin Freeman at the University of California, Irvine. The DIST.DAT file is in the format of an input file for UCINET. To generate a multidimensional scaling of the distance data, type DIM to load the dimensional analysis routines in UCINET, press M for the multidimensional scaling option, then enter the disk drive and DIST file name (e.g., F:DIST) to read the DIST.DAT file, then take all the default options for scaling. The DIM program will produce a multidimensional scaling of equivalences and give you the option of saving the map coordinates to disk so that you can create a high quality picture of the map with a graphics program.

DETECTING EQUIVALENT POSITIONS AND ROLES

Hierarchical cluster analyses are provided to identify subsets of equivalent actors within a system. A strong criterion of equivalence is zero distance, but usually only few actors are equivalent under this restrictive criterion. Less restrictively, actors are considered equivalent under a weak criterion of equivalence if they are separated by only a small distance. Typically, in practice, a position is operationally defined as an aggregate of actors equivalent under a weak criterion of equivalence, a set clustered on the criterion of distance.

The cluster analyses correspond to those already described for detecting cliques, however, there is no cut-off for equivalence boundaries as there is for cliques (where the cut-off is zero cohesion). The cluster analyses are generated automatically unless you suppress them (by putting a 1 in column 14 of the POSITIONS command). Initially, each actor is a cluster by itself, in the first step two most equivalent actors are joined, in the next step two highly equivalent units are again merged, etc., and in step N-1 all are collapsed. This procedure yields not a definitive set of clusters but a hierarchy of clusters, each level is a set of clusters where a cluster is a fusion of earlier clusters and is nested in larger clusters at subsequent stages of agglomeration; but at later levels of agglomeration equivalence within clusters tends to be lower. Two algorithms are used, the single-linkage method and Ward's method. Equivalent actors have little distance between them and are merged early in the clustering. The algorithms differ in the rule for merging and may therefore yield somewhat different clusters. Clusters of widely different size and elongated clusters with hardly equivalent actors at different ends of a cluster are produced more often with the single-linkage method than with the Ward method.

The single-linkage method (also called the minimum method) in its first step clusters together two actors that are most equivalent, i.e. separated by the smallest distance obtaining among all actors in the system. Then distance from an actor to this cluster is defined as the minimum distance between the actor and the actors in the cluster, thus defining a measure of distance from any cluster to any other cluster. Then the second step merges two clusters separated by the smallest distance. At this new level, distance from one cluster to another is again defined as the minimum distance between actors in the first cluster and actors in the other cluster (hence the label minimum method). Again, as the third agglomeration, two clusters separated by the smallest distance are merged. This process of sequential agglomeration continues until all actors are merged together in a single cluster. At each step the criterion for merging is the smallest distance between clusters which in that step becomes a new lower bound for the maximum distance between any two actors within the new clusters. More precisely, the criterion level is the maximum distance separating a member of a cluster from the closest other member of the cluster. This criterion level of maximum distance between nearest neighbors in a cluster increases as the agglomeration proceeds.

The Ward method (also called the minimum variance, or error sum of squares method) provides another clustering. Ward's method differs from the single-linkage method in its rule for merging clusters. The idea is that equivalent actors have identical distance to each actor in the system, so occupants of a position have distances to each actor that hardly deviate from their mean position distance to each actor. More precisely, a cluster with members m with distances d_{mq} to each actor q and mean distance $d_{.q}$ has error sum of squares defined as the grand sum of their squared deviations;

$$\text{ESS} = \sum_q \sum_m (d_{mq} - d_{.q})^2,$$

where summation is across all members m in the cluster and all actors q in the system (this may be recognized as an ANOVA within-group sum of squares). Initially, every actor is a cluster by itself which has an ESS of 0. As the first step, ESS is computed for every pair of actors and a pair with the lowest ESS is merged. In step one, in other words, the cluster is formed which produced the smallest increase in ESS in the system. This defines the general rule for merging. As next step, given a set of clusters each with its ESS, ESS is computed for each pair of possible new clusters, and the two are merged which produce the smallest increase in ESS (where the increase is the ESS of the merged pair minus the sum of the two individual cluster ESS). As the agglomeration proceeds, the criterion level ESS goes up, reflecting increasing differences among the actors clustered together as equivalent. Eventually, all actors are merged in a single cluster.

Cluster Analysis Printout

The cluster analyses are printed in two formats, a landscape diagram and a tree diagram. The tree may offer the best first overview but may be somewhat compressed and the landscape diagram adds detail. I'll illustrate both. The tree and landscape diagrams display the history of the agglomeration process from the extreme of each actor being a cluster by itself to the other extreme of the whole system in one cluster. A cut across the diagram shows the number of clusters and the actors in each cluster. Among the levels of agglomeration, it may be best initially to focus on the level just before the largest change occurs in the criterion for merging; these sequential criteria are printed alongside the landscape diagram. Additional output will be printed if you specify positions and occupants in the POSITIONS command as described below.

For example, the system of six actors with raw relations shown below at the left have the default Euclidean distances shown at the right,

```
 - 9 6 0 5 0              -    3.7  13.8  18.4  14.2  20.0
 9 - 3 0 5 0             3.7    -   12.3  19.2  14.3  18.7
 6 4 - 8 4 0            13.8  12.3    -   16.6   9.2  10.3
 0 0 8 - 7 9            18.4  19.2  16.6    -   12.7  11.4
 5 6 4 7 - 6            14.2  14.3   9.2  12.7    -   12.3
 0 0 0 9 6 -            20.0  18.7  10.3  11.4  12.3    -
```

The single-linkage landscape diagram is displayed at the left of the next page. The lowest distance in the distance matrix is the 3.7 between actors 1 and 2. They are clustered together in the first row (indicated by the XXX connecting their columns in the diagram). The second lowest distance in the matrix is the

```
   MAXIMUM                                MAXIMUM
  DISTANCE                               DISTANCE
 IN CLUSTER    1 2 4 3 5 6               VARIANCE     1 2 3 5 4 6
__________________________         __________________________

    3.742      XXX . . . .                3.742       XXX . . . .
    9.165      XXX . XXX .                9.165       XXX XXX . .
   10.344      XXX . XXXXX               11.402       XXX XXX XXX
   11.402      XXX XXXXXXX               15.699       XXX XXXXXXX
   12.329      XXXXXXXXXXX               29.088       XXXXXXXXXXX
```

9.2 distance between actors 3 and 5 so they are clustered second. The third lowest distance is the 10.3 distance between actors 5 and 6 so actor 6 is merged into the cluster with actor 5 as the third step (the larger distance between 6 and 3 is irrelevant). The fourth lowest distance is the 11.4 distance between actors 4 and 6 so in the fourth step actor 4 is joined into the cluster with actor 6. The fifth lowest distance is the 12.3 between actors 2 and 3 so their two clusters are then joined. As this illustrates, by the single-linkage method the criterion for merging is the closest linkage, all larger distances are disregarded; hence the name of the clustering algorithm.

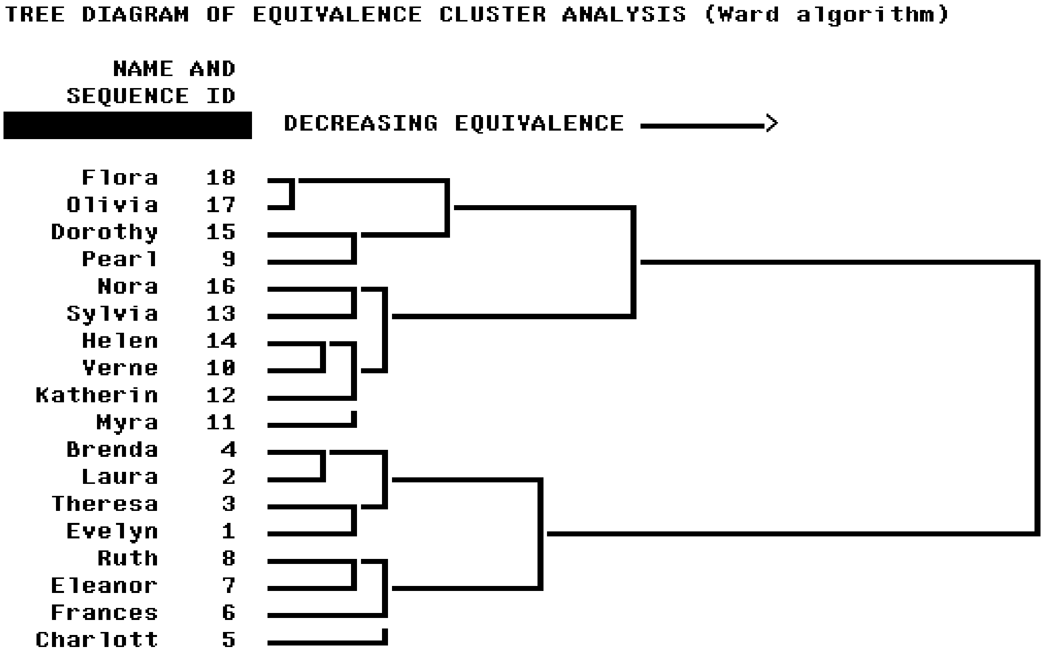
The Ward's cluster analysis is displayed above at the right. Although the two displayed analyses are based on the same distance matrix, the clusters are obviously somewhat different. Actors 1 and 2 are still equivalent in the Ward's analysis relative to the rest of the network, but a cleavage is suggested between actors 3 and 5 versus actors 4 and 6.

Typically, the two clusterings will yield different results because of differences between the algorithms. This is different from the two algorithms for detecting cliques. They can give different results also; but because they are invoked with different operationalizations of cohesion. More generally, an equivalence clustering depends on how relations are measured, on how equivalence is measured, and on how the algorithm merges equivalent actors (Burt, 1988a; Doreian, 1988). The moral is that all cluster analyses are no

```
                            C         K
                      T     h F E     a           D
                    E h   B a r l     t     S     o O
                    v e L r r a e     h V H y   P r l F
                    e r a e l n a R M e e e l N e o i l
     MAXIMUM        l e u n o c n u y r r l v o a t v o
    DISTANCE        y s r d t e o t r i n e i r r h i r
    VARIANCE        n a a a t s r h a n e n a a l y a a
-------------------------------------------------------
      .000          . . . . . . . . . . . . . . . . XXX
     1.414          . . XXX . . . . . . . . . . . . XXX
     3.464          . . XXX . . . . . . XXX . . . . XXX
     3.742          . . XXX . . . . . . XXX . . XXX XXX
     3.742          XXX XXX . . . . . . XXX . . XXX XXX
     4.000          XXX XXX . . . . XXX XXX . . XXX XXX
     4.243          XXX XXX . . . . XXX XXX XXX XXX XXX
     4.472          XXX XXX . . XXX XXX XXX XXX XXX XXX
     5.548          XXX XXX . . XXX XXXXXXX XXX XXX XXX
     6.000          XXX XXX XXX XXX XXXXXXX XXX XXX XXX
     6.574          XXXXXXX XXX XXX XXXXXXX XXX XXX XXX
     6.836          XXXXXXX XXXXXXX XXXXXXX XXX XXX XXX
     8.040          XXXXXXX XXXXXXX XXXXXXXXXXX XXX XXX
    11.036          XXXXXXX XXXXXXX XXXXXXXXXXX XXXXXXX
    19.502          XXXXXXXXXXXXXXX XXXXXXXXXXX XXXXXXX
    25.898          XXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXX
    55.473          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-------------------------------------------------------
```

more than suggestive. They are clues for detecting potentially equivalent actors. The Ward algorithm is the one to give stronger weight in conflicts between the two equivalence clusterings because it is more likely to generate clusters of equivalent actors that pass the variance tests for equivalence explained in the next section.

Before moving to that topic, consider a slightly more realistic example of cluster analysis printout. The example is the network of 18 women described in *Deep South* (Davis and Gardner, 1941). This is also a ridiculously simple example, but illustrates some generic qualities you will see in your cluster analyses. Relations in the network are based on newspaper society stories about the women (see pages 77ff in this *Manual* for more details). The landscape diagram of the equivalence clustering by the Ward algorithm is presented above. As explained on pages 138-139, a cluster of equivalent women is a set of women clustered with one another at the top of the diagram (strong equivalence) and not clustered with anyone else until the bottom of the



```
TREE DIAGRAM OF EQUIVALENCE CLUSTER ANALYSIS (Ward algorithm)

      NAME AND
   SEQUENCE ID
                 DECREASING EQUIVALENCE ———————>

    Flora   18
   Olivia   17
  Dorothy   15
    Pearl    9
     Nora   16
   Sylvia   13
    Helen   14
    Verne   10
 Katherin   12
     Myra   11
   Brenda    4
    Laura    2
  Theresa    3
   Evelyn    1
     Ruth    8
  Eleanor    7
  Frances    6
 Charlott    5
```

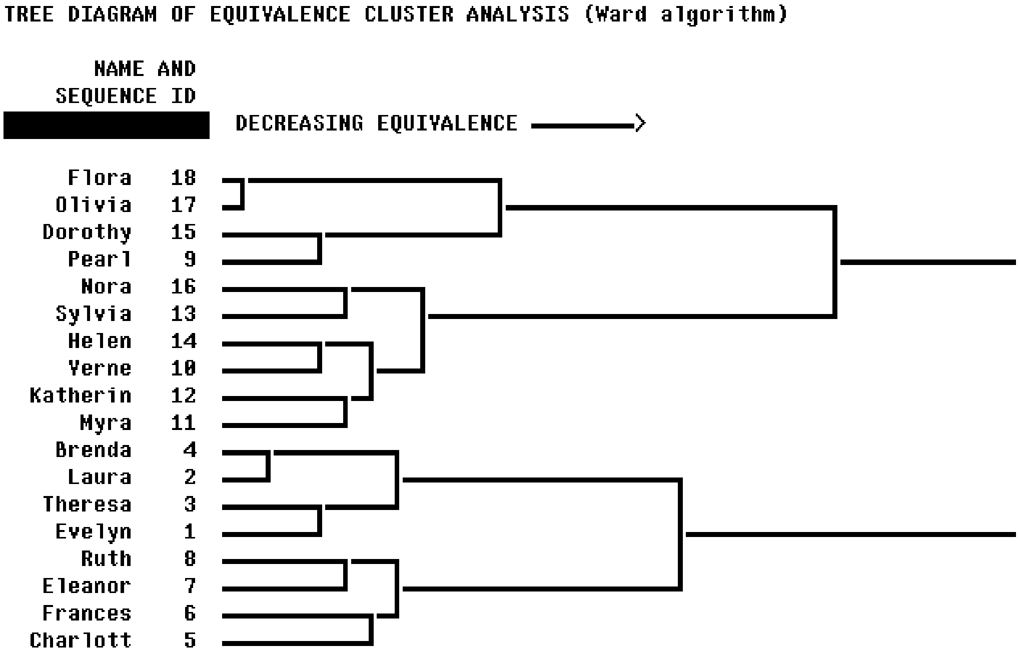
diagram (weak equivalence). Clusters are distinct hills of XXs. For example, the first eight women (Evelyn to Ruth) are a set of structurally equivalent women distinct from the other ten women, with an internal distinction between the first four women (Evelyn to Brenda) versus the other four.

The corresponding tree diagram is presented above. Both diagrams can be found in the printout of the first analysis in the example command file, EXAMPLES, on the program disk. The degree of equivalence between two women is indicated in the tree diagram by the length of the line that connects them. Flora and Olivia are the most equivalent. Flora and Brenda are among the least equivalent. A cluster of equivalent women is a set of women clustered with one another to the left of the diagram (strong equivalence) and not clustered with anyone else until the right of the diagram (weak equivalence). As I already mentioned with respect to the landscape diagram on page 140, the set of eight women, Brenda through Charlotte, are sharply distinct from the remaining ten women. The most obvious distinction within the set of eight women is between the four women Brenda through Evelyn versus the other four women.

Notice how the two kinds of diagrams complement one another. The tree diagram obscures fine-grain distinctions in equivalence to highlight the deepest divisions between sets of structurally equivalent women. The landscape diagram highlights fine-grain distinctions. The three distinctions between clusters to the right of the tree diagram account for about two thirds of the diagram. All other distinctions are condensed in the barely differentiated branches at the left of the diagram. That balance is reversed in the landscape diagram. The three deepest equivalence distinctions are given by the bottom three (of N-1) rows in the diagram. The remaining rows describe equivalence distinctions condensed at the left of the tree diagram.

The tree diagram can be displayed with more precision. Instead of the 80 column display on page 141, ask for the 132 column display (3 in column 16 of the POSITIONS command, see page 40). The 80 column display on page 141 distinguishes 25 levels of equivalence. Twice that many are in the 132 column display. If you re-run the EXAMPLES command file with a 3 in column 16 of the POSITIONS command in the first analysis, and browse to the bottom of the printout file SOUTH.PRT, you will see the screen at the top of the next page. This is the first 80 columns of the 132 column display. The line that connects the two clusters lies beyond the right margin of the screen, but the branches to the left of the diagram are a much better display of relative equivalence levels between the women.

The first 80 columns of the 132 column display contain most of the information in the tree diagram, so you might prefer to get the broader display even if you later print only the first 80 columns. A printer utility is available to give you control over printing the final copy (see Chapter 6 of the ASSISTANT *Command Booklet*). Note that if you request the graphics tree diagram



```
TREE DIAGRAM OF EQUIVALENCE CLUSTER ANALYSIS (Ward algorithm)

     NAME AND
  SEQUENCE ID
                DECREASING EQUIVALENCE ———>

    Flora   18
   Olivia   17
  Dorothy   15
    Pearl    9
     Nora   16
   Sylvia   13
    Helen   14
    Verne   10
 Katherin   12
     Myra   11
   Brenda    4
    Laura    2
  Theresa    3
   Evelyn    1
     Ruth    8
  Eleanor    7
  Frances    6
 Charlott    5
```

(displayed above and on page 141), the tree diagram is in the printout file and also in a separate file named _TREE.PRT (so that you can quickly print only the tree diagram).

ADEQUACY OF AN EQUIVALENCE HYPOTHESIS

STRUCTURE provides methods with which you can assess the adequacy of a hypothesis that a set of actors are equivalent. The measures of adequacy are routinely reported by STRUCTURE with density table results (as described below) and can be obtained interactively with the ASSISTANT program (see Chapter 4 of the ASSISTANT Command Booklet). Equivalent actors have identical distances to every other actor, so distances to equivalent actors should be perfectly correlated — allowing for measurement error. In other words, the matrix of covariances among distances to equivalent actors should have a rank of one (e.g., Burt, 1982:73-89; Ziegler, 1987). Given a set of M actors hypothesized to be equivalent, two kinds of measures are provided for the adequacy of the hypothesis.

The first is an aggregate measure. The actors are equivalent with one another to the extent that the covariance matrix of distances to them has a rank of one. This is reported as the ratio of the largest eigenvalue extracted from the matrix, divided by its maximum value (given by the sum of variances in distances to each occupant). The ratio is multiplied by 100 and printed as a percentage. It equals 100% for a set of perfectly equivalent actors.

The second kind of measure indicates the adequacy of each individual included in the set of equivalent actors. Distances to an individual are correlated with the mean distances to the other M-1 actors in the set. To the extent that the individual lies at the center of the set, this reliability correlation will be close to one. The actor does not belong to the position if the reliability is low. There is no fixed cut-off for acceptable levels of reliability. As in any scale construction, you are looking for actors for whom reliability is much lower than the reliabilities obtained for the others in the set of equivalent actors. Guided by your clustering output, you can use the ASSISTANT program to interactively detect adequate positions, as described in Chapter 4 of the ASSISTANT *Command Booklet.*

DENSITY TABLES & BLOCKMODELS

Once you have a roster of equivalent actors, then you can summarize the observed network(s) of relations in terms of characteristic relations between positions or roles. A density table is constructed for each input network where the cell z_{AB} is the average relation from actors in equivalence set A to actors in equivalence set B. A density table can be turned into an image matrix, an adjacency matrix of dichotomized densities. If derived from densities among structurally equivalent actors, the image matrices are called a blockmodel of the system. Alternative blockmodels are created by STRUCTURE when you request a density table. For further details, see Burt's review (1982:62-69) or the original discussion by White et al. (1976). However, you should be able to interpret the density table printout from the explanation included in the printout.

Structurally equivalent actors have similar relations to and from every actor, so all relations from each position to every position are similar, and therefore also similar to their average, their density. The network of relations among actors within a network is therefore concisely described by the densities among positions. STRUCTURE prints a density table for each network if the positions with their occupants are specified with the POSITIONS command (number of positions in columns 27-28, number of actors in first position in columns 29-31, number of actors in columns 32-34, etc.; then on a new line, the occupants of the first position listed by identification number, one occupant listed in column 1-4, another in columns 5-8, etc.; then on a new line the occupants of the second position, etc; see page 43). When you use the ASSISTANT program to define sets of equivalent actors, the command file needed to create a density table is automatically written to disk, ready to submit to STRUCTURE (see page 50 of the ASSISTANT Command Booklet). Individuals can be structurally unique in the sense of not being equivalent to anyone else in the system and therefore not grouped with others into a position. Actors unassigned to any position make up a residual group. Occasionally, it can be informative to specify a unique actor as a position by herself. Notably, if the most prestigious or central actor in the system is unique, her important location in the system can be captured by specifying her as a position; conversely, leaving her in the residual group would entail largely ignoring her in the analysis.

The density table printout provides equivalence statistics on each position, namely the percentage measure of overall equivalence within the position,

```
DATA ON ACTORS OCCUPYING POSITION  1:
 Occupant  Reliability  Correlations among Distances
 ________  ___________  ______________________________>
        1      .991        1.000
        2      .982         .950 1.000
Percentage of variance in distances to the occupants
accounted for by a single principal component: 97.52%
(149.146 eigenvalue in 3 iterations)


DATA ON ACTORS OCCUPYING POSITION  2:
 Occupant  Reliability  Correlations among Distances
 ________  ___________  ______________________________>
        4      .852        1.000
        6      .580         .416 1.000
Percentage of variance in distances to the occupants
accounted for by a single principal component: 70.82%
(86.463 eigenvalue in 4 iterations)


The following actors are NOT assigned to any position.
They define a RESIDUAL group:
           3        5
```

and the reliability of each occupant as a representative of the position. A density table is printed for every network. For each density table two image matrices are derived by different cutoffs for dichotomized densities. Setting positive densities to 1 and nonpositive densities to 0 yields one image matrix. Another image matrix is based on setting a density to 1 if it is greater than the overall average relation in the network and otherwise setting it to 0.

Let me illustrate with the system of six actors on page 138. Actors 1 and 2 are structurally equivalent and actors 4 and 6 are somewhat equivalent. Specifying these two positions in the POSITIONS command generates the equivalence results displayed above. These printed statistics indicate that the first position is adequate in that the accounted for percentage is nearly 100% and reliabilities are all nearly 1.00. The second position seems adequate only for exploration since the accounted for variation is only about 70% and the lowest reliability is .580.

The printout also includes the page of density table output displayed on page 148. The density table shows that the average relation among occupants of position 1 is 9.0 which is also the density among the occupants of position 2. The density of relations from the occupant of position 1 to the occupants of position 2 is 0.0 and to the residual actors is 4.750, etc. Finally, the printout lists two image matrices obtained from the density table by selecting different cut-off points. The two image matrices are identical in this system.

THIS IS THE DENSITY TABLE FOR NETWORK 1, where cell i,j is the average relation from someone occupying position I to someone occupying position J. The average relation between any two people in the network is 4.2.

```
ROW        |
DENSITIES  |  3.700 3.900 5.000
           |
COLUMN     |
DENSITIES  |  3.900 3.900 4.800
           |
POSITION   |     1     2  RESIDUAL
-----------+---------------------
     1     |   9.000  .000 4.750
     2     |    .000 9.000 5.250
           |
RESIDUAL   |   5.250 5.250 4.000
```

THESE ARE TWO (of many alternative) IMAGE MATRICES FOR THE ABOVE DENSITY TABLE, where the first is based on a zero cutoff (cell i,j = 1 if cell i,j density is greater than 0) and the second is based on the overall density of the network (cell i,j = 1 if cell i,j density is greater than the average relation in the network between any two people)

```
POSITION |  1 2            POSITION |  1 2
---------+------           ---------+------
     1   |  1 0                 1   |  1 0
     2   |  0 1                 2   |  0 1

CUTOFF AT ZERO                  CUTOFF AT OVERALL DENSITY
```

CONTAGION

The image here is of social structure positioning two people such that what one person says or does is contagious for the other. The two people come to resemble one another in opinion and behavior. Where responses on some criterion variable are affected by contagion, a network correlation results. Ego's response is correlated with alter's response because they are proximate in social structure. This idea is modeled with the CONTAGION command in conjunction with a definition of what makes ego and alter proximate in social structure. The CLIQUES command gives you options for defining the probability that two people communicate — which results in their similar responses. This is the tradition of social psychology in the 1930s and 1940s, most often recognized in Festinger et al.'s (1950) study of student opinion, and opinion surveys in the 1940s and 1950s by the Bureau for Applied Survey Research (e.g., Lazarsfeld et al., 1944; Berelson et al., 1954; Katz and Lazarsfeld, 1955; Coleman et al., 1957). The POSITIONS command gives you options for defining the probability that two people are similarly positioned in the broader network and so use one another as a frame of reference — which results in their similar responses. Burt (1992b) discusses theoretical models for cohesion and equivalence contagion, and uses the models to describe response data on study populations of physicians, scientists, corporate officers, and political lobbyists. Illustrative substantive studies taking advantage of recent advances in network analysis include White et al. (1981), Friedkin (1984), Johnson (1986), Burt (1987), Schøtt and Morrissey (1989), Guilarte (1990), Friedkin and Cook (1990), and Galaskiewicz and Burt (1991).

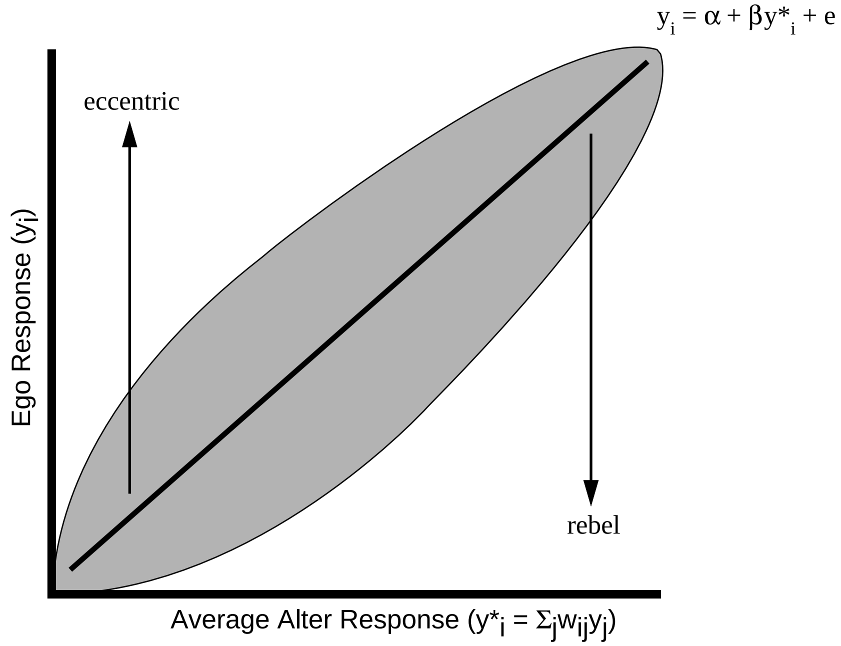
ANALYSIS

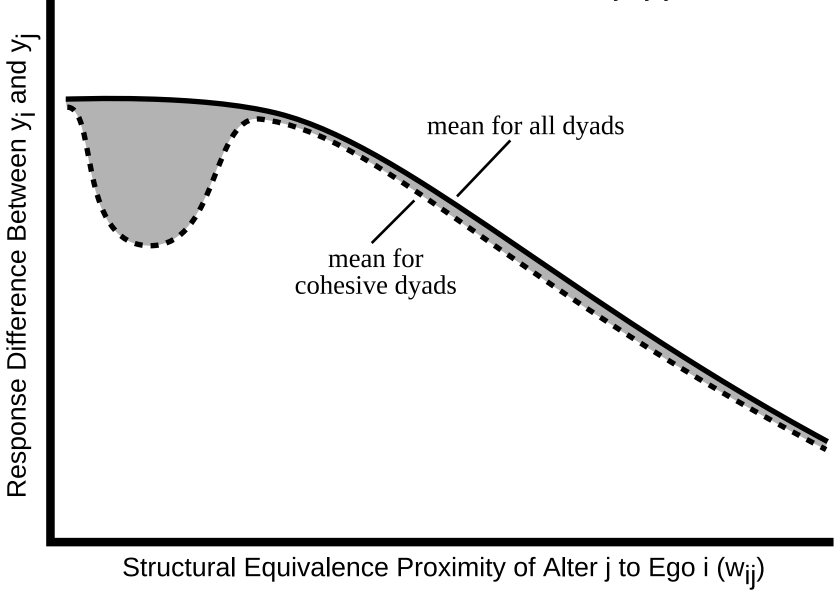
There are diverse analyses possible with the CONTAGION command (and the simulation options expand the possibilities to many more). Analysis reveals the extent to which contagion affects a criterion variable and the mechanism by which contagion operates in the study population. I've got ego's response y_i on some criterion variable, alter responses y_j, and a network weight w_{ij}. The weight is a proportion ($0 \leq w_{ij} \leq 1$, $\sum_j w_{ij} = 1$), that excludes ego as an interpersonal influence on herself ($w_{ii} = 0$), and increases with the extent to which ego i's response is expected to resemble alter j's response. The two generic lines of analysis with cross-sectional data are to predict ego's response from all alters (aggregate analysis) or individual alters (dyad analysis).*

The aggregate level analysis involves averaging alters, $y^*_i = \sum_j w_{ij}y_j$, and using the average to predict ego's response. If my network consists of two equally influential alters who responded with a 2 and a 3 on the criterion variable, my response is expected to be 2.5. The graph at the top of the next page illustrates the analysis. The summary association is described by the regression equation — a network correlation model — where α is the regression intercept (expected ego response when alters average a zero response), β is the regression slope coefficient (expected increase in ego's response from a unit increase in average alter response), and e is a residual. The bold line shows ego response on the vertical axis increasing with average alter response on the horizontal axis ($\beta > 0$). The program printout includes estimates of α and β, and a graph of y_i across y^*_i. Additionally, ego and average alter responses can be written to the output data file for more detailed analysis (1 in column 30 of the CONTAGION command), or the weight matrix of w_{ij} can be written to the output data file to estimate α and β with other programs (4 in column 30).

There are two reasons for more detailed analysis; studying the meaning of deviations from the regression line and identifying reasons for deviations.

*If the timing of responses is known, then population diffusion curves are a third generic line of analysis (e.g., see Burt, 1992b:Chap. 3, for models and references).


$y_i = \alpha + \beta y^*_i + e$
eccentric
rebel
Ego Response (y_i)
Average Alter Response ($y^*_i = \Sigma_j w_{ij} y_j$)


mean for all dyads
mean for cohesive dyads
Response Difference Between y_i and y_j
Structural Equivalence Proximity of Alter j to Ego i (w_{ij})

Deviations from the regression line can mean different things such that deviations above and below the line are different kinds of contagion evidence. Where rules of conformity are broadly defined in a study population, contagion is less well measured by conformity than by deviance. The two are not polar opposites; deviance has direction. This is an insight in the early Festinger et al. (1950) study. They report conformity in terms of a specific deviance. Studying conformity and deviance, Burt and Doreian (1982; Burt, 1982:Chap. 6) find stronger contagion evidence in deviations from alters. Consider the graph again at the top of page 151. Burt and Doreian describe scientist opinions about where to publish their work. Interest in a journal ignored by peers is not the same deviation as rejecting a journal prescribed by peers. Scientists who express interest in a journal ignored by their alters are eccentrics; they have interests that deviate from alter opinion, but without questioning the validity of alter opinion. In contrast, scientists who see little value in a journal prescribed by their alters are rebels; their deviance from alters questions the validity of alter opinion. Burt and Doreian find their strongest evidence of contagion in the frequency of rebel responses. The asymmetric meaning of deviance from alter opinion varies between study populations. The point here is merely that it is a good idea to write the response data to the output data file, separate deviations above alter response ($y_i > y^*_i$) from deviations below alter response ($y_i < y^*_i$), and check correlations with the two kinds of deviations. If deviations in the two directions have similar correlations with third variables under alternative contagion models, then contagion is adequately described by the summary measure of association between y_i and y^*_i. If deviations in the two directions have different correlations, then alternative contagion models might be better tested with a criterion variable measuring ego's rejection of alter opinion or behavior.

Another reason for more detailed analysis is to identify control variables that explain why certain deviations occur. The response data are written to the output data file and merged with other data on ego. The most obvious other data are personal preferences that might affect ego response. In Burt and Doreian's study, for example, scientists had personal preferences for certain journals because they had published there or they were active editors. In addition to personal preferences, social structural conditions can dissipate contagion's effect. Variation in alter response around the average y^*_i is not constant from one ego to another. Some people have heterogeneous alters; y_j vary widely around y^*_i. Others have homogeneous alters (low variance). Where alters disagree, ego might feel more free to express her personal preferences. When you put a 1 in column 30 of the CONTAGION command to write y_i and y^*_i to the output data file, data are also written on the extent to which ego has no alters, and the standard deviation of alter response.

A second line of analysis is to study the dyadic conditions for contagion. Instead of measuring ego's response similarity to the aggregate of alters, similarity is predicted between individual ego-alter pairs. Put a 2 in column 30 of the CONTAGION command to write ego-alter results to the output data file. This will include their responses y_i and y_j, relations between them z_{ij} and z_{ji}, and the network weight w_{ij}. When you read these data into a program for statistical analysis, compute the absolute difference between ego and alter responses, then plot the difference across levels of w_{ij} to get a graph like the one at the bottom of page 151.* The solid line shows a declining mean difference between ego and alter as they become increasingly structurally equivalent. A sharp drop in the line would indicate a threshold for contagion, but available evidence shows smooth decrease with increasing w_{ij} as presumed in the network correlation model (Burt, 1992b:Chap. 9). In addition to checking for nonlinearity in the contagion effect, dyad analysis is useful for

*If dyads will be compared across study populations, compute a z-score dyad difference, $|y_i-y_j|/s_i$, where the absolute response difference between i and j is divided by the standard deviation between responses (cf. Burt and Uchiyama, 1989; Burt, 1992b:Chap. 8).

studying the manner in which cohesion and equivalence interact to generate contagion. The dashed line in the graph at the bottom of page 151 shows mean differences between ego and alter connected to one another, versus the solid line of differences in all ego-alter pairs (for this analysis, the solid line should be based on disconnected pairs). Where the dashed line is lower than the solid line (the shaded area in the graph), cohesion brings ego and alter responses closer together. If cohesion generated contagion at all levels of structural equivalence, the dotted line would parallel the solid line at a lower level. Instead, the pattern displayed in the graph is observed. Cohesion generates no contagion between structurally nonequivalent people (far left of the graph) nor between structurally equivalent people (right half of the graph). It has its effect on people who are weakly equivalent; people who have some, but not many contacts in common. This effect is the shaded bulge between the solid and dashed lines to the left of the graph. Relations between nearly equivalent people seem to bring them into one another's frame of reference and so trigger contagion (Burt and Uchiyama, 1989; Burt, 1992b:Chap. 8).

MEASURING CONTAGION

There are two basic tasks in a contagion analysis: defining the network weights (the w_{ij}), and estimating effects (such as the β in the network correlation model, correlations with y_i deviations above and below $y*_i$, and rates of change in $|y_i-y_j|$ across levels of w_{ij}). Software solutions can handle one task, or both, or some part of each. The most common solution is to estimate α and β under rudimentary network weight options. The substance of contagion, however, lies in defining the network weights. STRUCTURE is written to solve that problem; helping you select the network weights appropriate for a specific criterion variable in a specific study population. Options for writing network weights and contagion variables to the output data file give you the data for analysis with statistical and data display software.

Measuring the Aggregate Contagion Effect

The network correlation is used to decide between alternative network weights. Given network weights (explained below), the network correlation model predicts ego's response:

$$y_i = \alpha + \beta(y^*_i) + e,$$

where y^*_i is the weighted average of alter responses ($\sum_j w_{ij} y_j$). The program computes ordinary least-squares estimates of the regression intercept, slope, and corresponding correlation coefficient. A graph of the y_i across the y^*_i is also presented to describe the shape of the association and let you spot outliers.

If the input data are a random sample of people in a much larger population, these least-squares estimates can be inconsistent and inefficient (Ord, 1975:121-122). To get maximum-likelihood estimates, write the network weights to the output data file (4 in column 30 of the CONTAGION command) for analysis with other software (e.g., Doreian, 1980; Dow et al., 1982).

However, if the input data are population data as is typically the case in a network analysis (relations among the largest producers, relations among the core players, relations among everyone in the organization unit), then the ordinary least-squares estimates accurately describe the association between ego and alter responses. The relevant statistical question is not one of making an inference about a population parameter, but is instead one of determining the robustness of the known population contagion effect. Is contagion a strong effect robust across individuals in the population, or is it a fragile effect that changes dramatically across individuals?

Jackknife results are computed to answer the question "How significant is the contagion effect?" The observed data are used to construct a sampling distribution specific to the study population. The aggregate contagion effect is then assessed as strong or weak as a function of how much it varies within the study population.*

Let r be the correlation corresponding to the regression slope β in the network correlation model estimated across all N egos. Let r_i be the same correlation, but computed with all data on actor i deleted — deleted as ego and deleted as alter to any other individual. The subsample correlation r_i is the network correlation that would have been observed if actor i's response were unknown. Repeating the computation for each of the N actors generates N subsample estimates of the network correlation.

The subsample correlations contain two kinds of information. First, they indicate actor i's contribution to the contagion effect. Subsample correlation r_i is higher than r to the extent that i shows little evidence of contagion (network correlation is higher with actor i deleted). It is lower than r to the extent that i contributes to the evidence of contagion (network correlation is lower with actor i deleted). Second, they indicate the stability of the network correlation. If the subsample correlations are distributed over a wide range of values that includes zero, then you can't have much confidence in the strength of the aggregate contagion effect. To the extent that they are distributed in a narrow range well above zero, the aggregate contagion effect is significantly greater than zero (rejecting the null hypothesis that β is 0).

Finifter (1972:159ff) provides an intuitive introduction to jackknife statistics. Mosteller and Tukey (1968:133ff) provide a widely cited introduction with several numerical illustrations. Efron (1982, especially Chap. 3) provides a more theoretical discussion, and Hinkley (1978) and Miller (1974) discuss jackknife statistics for correlation and regression models. All jackknife calculations in STRUCTURE are double precision. Results are not provided for individuals missing responses (y_i or y^*_i missing), nor for criterion variables constant across alter responses (y variance less than .001), nor for variables complete on only three or fewer individuals (trivial correlations).

The jackknife results provide a t-test to measure effect significance. The following, dubbed a "pseudovalue" by Tukey, is like an estimate of the network correlation in the deleted observation i:

$$\mathbb{r}_i = Nr - (N-1)r_i.$$

The mean of these pseudovalues is the jackknife estimate, $\mathbb{r}$, of the aggregate network correlation r,

$$\mathbb{r} = (\sum_i \mathbb{r}_i)/N,$$

and the estimated variance around the mean,

$$s^2 = [\sum_i(\mathbb{r}_i-\mathbb{r})^2]/[N-1],$$

defines the variance of the estimated mean;

$$s_{\mathbb{r}}^2 = s^2/N$$

so the following is a t-test of the extent to which the estimate is greater than zero;

$$t = \frac{\mathbb{r}}{s_{\mathbb{r}}}.$$

The t-test has N-1 degrees of freedom under the null hypothesis of no contagion effect. This calculation is carried out with Fisher's Z-transformation of the correlations (inverse hyperbolic tangent; $.5[\ln(1+r_i)-\ln(1-r_i)]$), to make the distribution of correlations slightly more normal and eliminate inadmissible pseudovalues (Miller, 1974:6-7, 12-13; Hinkley, 1978).

Network Weights

The social structural condition for contagion is defined by column 14 of the CONTAGION command. If the column is blank or 0, then contagion by equivalence is measured. The program looks for a POSITIONS command to see how you wish to define equivalence (the eight options are described on pages 40 and 124-134). The program defaults to structurally equivalent raw relation patterns if there is no POSITIONS command (the default POSITIONS command, see page 40). Equivalence is a linear transformation of distance;

$$\text{proximity i to j} = \text{dmax}_i - d_{ij},$$

where dmax_i is the largest distance from actor i. If you put a 1 or a 2 in column 14 of the CONTAGION command, then contagion by cohesion is to be measured. A 1 sends the program to a CLIQUES command to find out how you wish to define cohesion (the four options are described on pages 109-111). A 2 in column 14 aggregates the input relations across networks to define cohesion;

$$\text{proximity i to j} = \Sigma_k \, z_{ijk}.$$

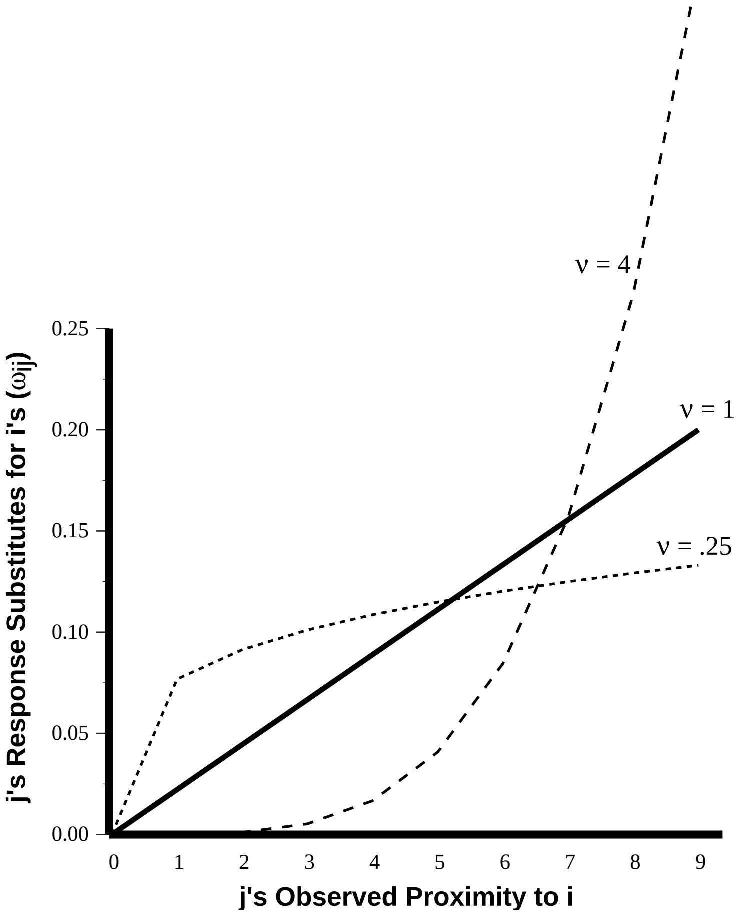
If i sends relations to no one, z_{ii} is set to 1. Substitution weights ω_{ij} are computed from the proximities. The weight ω_{ij} measures the proportional extent to which j's response could substitute for i's ($0 \leq \omega_{ij} \leq 1$, $\Sigma_j \, \omega_{ij} = 1$):

$$\omega_{ij} = \frac{(\text{proximity of i to j})^{\nu}}{\Sigma_q(\text{proximity of i to q})^{\nu}}.$$

Weights for the network correlation model delete the self-weights:

$$w_{ij} = \omega_{ij}/(1-\omega_{ii}),$$

so $w_{ii} = 0$ and $\Sigma_j \, w_{ij} = 1$. Network weights are undefined if there are no alters ($\omega_{ii} = 1$). Without alters, i is deleted as an ego, and y^*_i is set to the missing data code for y_i (or -999). More generally, the magnitude of the self-weight ω_{ii} indicates the extent to which i is free of alter influence. High ω_{ii} occur for isolates in contagion by cohesion and structurally unique actors in contagion by equivalence. The self-weight ω_{ii} is included in the written output as a control variable for analysis (personal preferences more likely determine responses by people with only minor alter influence, i.e., with high ω_{ii}).



The exponent ν varies ego's frame of reference for contagion and has proved to be essential to modeling equivalence contagion. It is a nonzero real number that redistributes proximity to define a domain of alternative weights — with interdependence at one extreme (ν <<1), independence at the other extreme (ν >> 0), and observed social structural conditions in the middle (ν = 1). Consider the above graph of observed proximity between i and j (horizontal axis) and the substitution weight measuring the extent to which j's response substitutes for i's (vertical axis).

If ν equals 1, it drops out of the weight definition; substitution weights, and so network weights, are proportional to observed social structural conditions. This is the straight line in the above graph. For example, if sociometric choices define cohesion proximity, then w_{ij} for influential alters equals one over the number of i's choices. Ego i citing two people has network weights of .5 from each cited person. Ego's predicted response is the average response of the two cited people ($y^*_i = .5y_1 + .5y_2$).

More continuous measures of i's proximity to j often require higher values of ν. Otherwise, everyone influences i's response to some extent, so predicted responses regress to the central limit of the mean. Values of ν greater than 1 take proximity from the most distant people and reallocate it to the most influential people. Increasing ν, in other words, narrows ego's frame of reference around alters relevant to a specific response. The higher ν is, the more ego is influenced only by her most proximate alters. In the graph on the preceding page, the ν of 4 decreases the substitution weight of people with observed proximity less than 7 (dashed line below solid line) and increases the weight of people with observed proximity of 8 or 9 (dashed line above solid line).

Alternatively, a fractional ν between 0 and 1 expands ego's frame of reference. The closer to zero ν is, the weaker the criterion for alters influencing ego's response. This moves everyone's predicted response toward the average for the whole system. In the graph on the preceding page, the ν equal to .25 gives nearly equal substitution weight to any alter with nonzero proximity to ego (dotted line).

The exponent ν is read from columns 23-27 of the CONTAGION command in F5.3 format. If the columns are blank or zero, then the program searches across alternative values of ν for one that yields the strongest contagion effect. The program begins with ν equal to 1 and increases by integers.* The average correlation at each value of ν is compared to the average correlation obtained

*Exponent values less than 1 have not been useful in substantive research, so they are not routinely included in the iterative search. You can force the program to consider them by entering positive fractions in columns 23-27.

with the preceding value. When the correlation increases by less than .05 from ν to ν+1, the iterations stop, and the contagion analysis is performed with the value ν. You can change the default by entering a negative fraction in columns 23-27. If you enter -.200, for example, iterations will stop when the network correlation increases by less than .2 from ν to ν+1. If you enter -.005, iterations will stop when the correlation increases by less than .005 from ν to ν+1.

Default Analysis

Suppose that you request a CONTAGION analysis and provide no other information, as in the following command file:

```
DATA        MYDATA
NETWORK     010 D
CONTAGION
ANALYZE
```

Default parameters are invoked (see pages 24-25); all 10 people in the network will be analyzed as ego (column 12), alters are ego's structurally equivalent peers (column 14), network and response data are read from the input file MYDATA (columns 16 and 18), there is one criterion response variable (column 20), no responses are missing (column 22), and the program is to search across integer values of ν to find the value at which the network correlation increases by less than .05 (columns 23-27).

This is the input data file MYDATA (cf. the input file on page 58 and the sociogram on page 57).

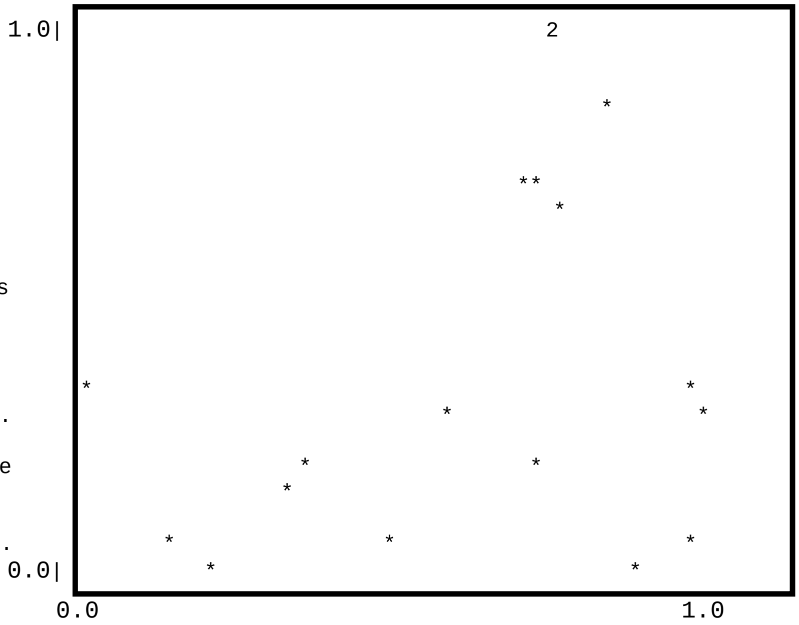
```
(10F1.0)
 000100000
0 00100000
00 0100000
000 100000
0000 01111
00000 0000
000000 111
0000001 11
00000011 1
000000111
(F2.1)
1
1
1
1
0
0
4
4
4
4
```

The response data are at the end of the input data file, with data on each respondent beginning on a new line and the response data preceded with a format statement (limited to 80 columns) describing how the response data are to be read.

The default printout is a summary description of contagion. The following is generated by the example command file MEDICAL. The network is advice relations among physicians in the fourth city of Coleman, Katz and Menzel's (1957, 1966) diffusion study of a new drug. The criterion variable is the month in which each physician began prescribing the new drug. Contagion is measured between structurally equivalent physicians.

```
OBSERVED RESPONSES                              mean:  8.778
                                  standard deviation:  5.094


RESPONSES EXPECTED FROM CONTAGION               mean:  7.586
                                  standard deviation:  1.158


CONTAGION EFFECT ( 18 observations)
                               regression intercept:  -2.041
                                   regression slope:   1.426
                                network correlation:    .324
                          jackknife t-test ( 17 df):   1.858
```

First, observed responses are described (the y_i); on average, physicians adopted the new drug 8.8 months after it was released, with a standard deviation of 5.1 months. Second, the expected responses are described (y^*_i). These are more tightly distributed around their mean because alter responses are averaged (1.2 standard deviation for y^*_i versus 5.1 for y_i). The summary results on contagion are then printed; intercept α and slope β in the network correlation model, followed by the network correlation (standardized β),



RESPONSE GRAPH: Horizontal axis plots responses expected from contagion (where 1.0 is the maximum expected response and 0.0 is the minimum) and the vertical axis plots the observed responses. Responses on the axes are correlated .324, as reported above. A * in the graph indicates a single observation, and a + indicates 9 or more observations.

followed by the jackknife t-test. The t-test has N-1 degrees of freedom. The 1.858 t-test with 17 degrees of freedom for the physicians gives the null hypothesis a .04 probability (one-tail test for the positive network correlation measuring contagion) — structurally equivalent physicians tended to begin prescribing the new drug at about the same time. A one month average delay by alters is associated with an average 1.4 month delay in ego's adoption (β is 1.426).

The other part of the default printout is a graph of ego response against aggregate alter response; displayed above for the physicians. Observed response varies on the vertical axis; the lowest value of y_i is set to 0 and the highest is set to 1 (i.e., $[y_i\text{-min}(y_i)]/[\max(y_i)\text{-min}(y_i)]$). Physicians adopting late are at the top of the graph. Alter response varies on the horizontal axis, also scaled to vary from 0 to 1. The graph shows a possible nonlinear association (physician delays increasing slowly with initial alter delays, then increasing more quickly with later alter delays), and four outliers in the lower-right of the graph. If the four outlier physicians (numbers 3, 8, 11 and 14) are excluded as egos, the network correlation increases to .78 and the t-test increases to 2.73 with 13 df, which gives the null hypothesis less than a .01 probability. The data analysis task is to find a variable that distinguishes the four physicians to explain their adoptions so far ahead of their alters.*

These results describe contagion with ν equal to 1. The automatic search for ν tried ν equal to 2, and obtained a slightly larger network correlation of .349. Since this is less than a .05 increase over the correlation for ν equal to 1, the exponent is left at 1. A -.005 in columns 23-27 of the CONTAGION command will increase the precision of the search across alternative values of ν.

Variations on the Default Analysis

There are many variations on the default analysis. I'll quickly sketch them to get you started, but there are too many to describe in the same detail as the default analysis.

Output Options

You can expand the default output in various ways. For example, the example command file MEDICAL has a 1 in column 32 to request details on the jackknife estimation of the contagion effect. This generates another page of printout behind the summary, default printout. The page of details for the MEDICAL analysis is displayed on the next page. The jackknife estimate and standard error are displayed (1.858 = .5038/.2711), followed by a histogram

*The four physicians were identified by writing ego results to the output data file (1 in column 30 of the CONTAGION command), and sorting on y^*_i to find the highest values. The outlier physicians were deleted as egos as described below on pages 170-171.

```
ESTIMATION DETAILS:

The jackknife t-test is the ratio of the jackknife estimate of the Fisher
Z-transformed network correlation:  .5038, divided by the  .2711 standard
error of the estimate.

Distribution of correlations within subsamples where each asterisk is one
subsample:
                                                *
                                              * *
                                            ******
                                           *********
         -----+------------------+------------------+-----
                0.5              0.0                0.5


Subsample results aggregated in the jackknife estimates:

      ID = RESPONSE VARIABLE, SEQUENTIAL ID NUMBER FOR EGO, EGO NAME
      X1 = SUBSAMPLE CORRELATION with row actor deleted (plotted above)
      X2 = CORRESPONDING PSEUDOVALUE from Fisher Z transformation of the
             subsample correlation (average is above jackknife estimate of
             network correlation)

IDENTIFICATION     X1       X2

1   1       1    .25513   1.61582
1   2       2    .25374   1.64117
1   3       3    .44731  -2.13166
1   4       4    .29347    .91096
1   5       5    .36247   -.40439
1   6       6    .25037   1.70229
1   7       7    .24133   1.86582
1   8       8    .34940   -.14992
1   9       9    .27950   1.16956
1  10      10    .30145    .76205
1  11      11    .41342  -1.42449
1  12      12    .33750    .07944
1  13      13    .29526    .87753
1  14      14    .37456   -.64220
1  15      15    .31645    .48016
1  16      16    .32943    .23383
1  18      18    .31862    .43905
1  30      30    .23143   2.04415
```

of the subsample correlations. The histogram is the sampling distribution for the contagion effect in the study population. If it covers a wide range, the effect is unstable. The MEDICAL subsample correlations are clustered around .3, from a minimum of .23 to a maximum of .45. The histogram is followed by a listing of the subsample correlations and pseudovalues (r_i and $𝕣_i$ on page 157). The mean of the pseudovalues is the jackknife estimate of the Z-transformed network correlation and variation around the mean defines the standard error of the estimate ($𝕣$ and $s_{𝕣}$ on page 157). A 3 in column 30 of the CONTAGION command writes this listing to the output data file. The subsample correlations and pseudovalues can be a useful diagnostic for understanding who is most susceptible to contagion and who is most resistant (e.g., see Burt, 1992b:Chap. 7). Negative pseudovalues mark resistant actors.

To see the responses for each ego, ask for ego results (1 in column 28 of the CONTAGION command). Here is the top of the ego printout requested in the MEDICAL command file (note that ν is spelled out as nu in the printout):

```
CONTAGION RESULTS with alters defined by equivalence, and the exponent
nu set to a value of 1.0:

   ID = RESPONSE VARIABLE, SEQUENTIAL ID NUMBER FOR EGO, EGO NAME
    X1 = SELF-WEIGHT
   X2 = NUMBER OF ALTERS (Wij > .001, excluding self)
   X3 = PERCENT ALTER RESPONSES MISSING
    X4 = EGO RESPONSE (Yi)
   X5 = AVERAGE ALTER RESPONSE (SUMj Wij*Yj)
   X6 = STANDARD DEVIATION OF ALTER RESPONSE

IDENTIFICATION   X1    X2    X3        X4         X5         X6

1   1       1  .089   29  34.9    18.000      8.117      4.861
1   2       2  .085   30  35.4    18.000      8.060      4.822
1   3       3  .097   30  35.5     4.000      8.981      5.369
1   4       4  .110   30  54.3     5.000      6.479      4.216
1   5       5  .088   30  59.0     8.000      5.123      3.043
1   6       6  .090   30  60.6     4.000      5.625      3.141
```

Following the identification data at the front of the row for each ego, six variables are printed: (1) the substitution weight measuring the extent to which ego has no proximate alters (ω_{ii} on page 158), (2) the number of alters (number of others for whom the network weight w_{ij} on page 158 is larger than .001), (3) percentage of alter response missing (explained below), (4) ego's observed response (y_i), (5) the average response by ego's alters (y^*_i), and (6) the standard deviation of their responses. As the aggregate alter response is a network-weighted average:

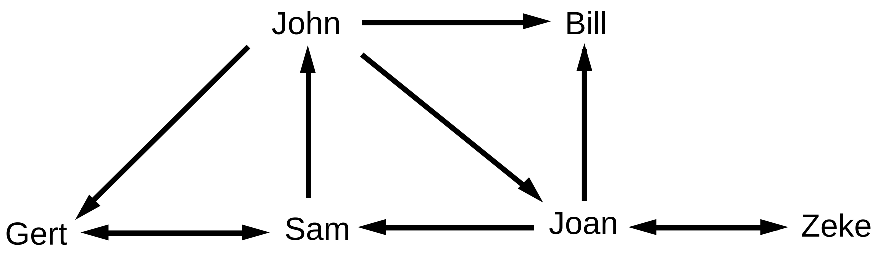
$$y^*_i = \Sigma_j w_{ij} y_j,$$

the variance of their responses is a network-weighted average:

$$s_i^2 = \Sigma_j \, w_{ij}(y_j - y^*_i)^2.$$

A new row is printed for each response variable analyzed (column 20 of the CONTAGION command), and a 1 in column 30 of the CONTAGION command writes the listing to the output data file as you see it in the printout.

To see the specific alter responses combined to predict ego's response, ask for dyad results (2 in column 28 of the CONTAGION command). Consider contagion in the network at the top of the next page. This is the fourth analysis in the example command file CONTAG (command file and data are listed on page 172). Sam cites two people, Gert and John. Under contagion by cohesion, Sam's opinion should resemble theirs. The computer output under the sociogram is the page of printout on contagion around Sam. His alters are listed by row, with their network weight (here both .5) and responses on the criterion variables (here two variables). The results in the ego-level printout appear at the end of the alter list. The first is the percentage of alter response missing (explained below). Next, ego's observed responses (y_i) are printed to compare to the average alter response (y^*_i) in the row below. Sam answered 4 on the first variable and his alters gave an average answer of 3 (the sum of



Sam's CONTAGION RESULTS with alters defined by cohesion, and the exponent nu set to a value of 1.0.

```
OBSERVED RESPONSES BY ALTERS (imputed responses are marked by an asterisk and pre-
ceded by the ID# in parentheses of the actor providing the imputed response)

--- ALTER ---  NETWORK         DATA ON EACH RESPONSE VARIABLE
  ID   NAME    WEIGHT      1        2         3         4         5
-------------------------------------------------------------------------


                        *( 2)
   3   John     .500     1.000     5.000


   5   Gert     .500     5.000     5.100


% Alter Response Missing ----------------------------------------------------
                          50.0%       .0%

Ego Observed Response ------------------------------------------------------
                         4.000     4.100

Ego Expected Response ------------------------------------------------------
                         3.000     5.050

SD of Alter Responses ------------------------------------------------------
                         2.000     0.050
```

.5 times John's 1 plus .5 times Gert's 5). He answered 4.1 on the second variable and his alters gave an average answer of 5.05. The line below that gives the standard deviation of alter responses. Sam's alters largely agree in their response on the second variable (standard deviation of .05 on a mean response of 5.05), but vary widely on the first variable (standard deviation of 2 on a mean response of 3, where the 2 is the square root of .5(1-3)^2 for John's deviation from the alter mean plus .5(5-3)^2 for Gert's deviation from the alter mean).

A 2 in column 30 of the CONTAGION command writes ego-alter dyad results to the output data file. For analysis, the written data differ from the ego-alter printout. Here are the written data on Sam for the first response variable (decimal zeroes deleted from the last two variables):

```
Exam  1   1   2    Sam    Joan 0 .0000    4.0000    1.0000    0    1
Exam  1   1   3    Sam    John 1 .5000    4.0000    1.0000    1    0
Exam  1   1   4    Sam    Bill 0 .0000    4.0000    9.0000    0    0
Exam  1   1   5    Sam    Gert 0 .5000    4.0000    5.0000    1    1
Exam  1   1   6    Sam    Zeke 0 .0000    4.0000     .0000    0    0
```

The first four columns are the first four letters from the title in the TITLE command (written for identification at the beginning of all indices written to the output data file), followed by the variable number (these are for criterion variable 1), followed by the sequential identification numbers and alphanumeric identification for ego and alter (Sam is the first person in the input data file). Although only John and Gert are alters for Sam in the printout, a line is written for every ego-alter pair; Sam with each of the five other people in the network. This way, the ego-alter dyad data can be merged easily with other dyad data files. Following alter name, a missing data dummy is written; 1 if the alter's response is missing and had to be imputed, 0 otherwise (John's response had to be imputed). If the alter's response couldn't be imputed, it is set to the missing data code on the criterion variable (as described below). The next column contains the network weights (w_{ij} is .5 for John and Gert, 0 for everyone else), followed by ego's response (y_i is 4 for Sam) and alter's response (y_j). The absolute value of the difference between y_i and y_j is plotted across values of w_{ij} for the dyad graph at the bottom of page 151. The last two columns are the relations from ego to alter, and back ($\sum_k z_{ijk}$ and $\sum_k z_{jik}$). As illustrated in the sociogram at the top of page 168, Sam cites John and Gert and is cited by Joan and Gert. The relation data are useful for studying mixtures of cohesion and equivalence contagion in the dyad (as illustrated with the dashed line in the graph at the bottom of page 151). For multiple criterion variables, a new row is written for each alter on each variable.

The final output option is to write the matrix of network weights to the output data file (4 in column 30 of the CONTAGION command). These are the weights used to compute the network correlation, not the raw substitution weights (they are the w_{ij} on page 158, not the ω_{ij}). They are provided in case you want to get maximum-likelihood estimates of α and β with other software. Weights are written as a square matrix with each row beginning on a new line. The default is to print the weights with three decimal places, but you can change that precision with an RFORMAT command (pages 48-49).

Selecting Egos

The contagion analysis involves building a network of alters around an individual ego and aggregating alter responses to predict ego's. The default in column 12 of the CONTAGION command is to analyze every actor in the input network as ego. A 1 or 2 in the column limits the computation to the first or last actor in the input network (for ego-network analyses). If there is a 3 in the column, the program reads the line in the command file following the CONTAGION command in 80I1 format to know which actors are to be analyzed as ego (for networks larger than 80, continue on a second line; or use the ASSISTANT program to select actors by menu). For example, contagion is computed for only some of the physicians in the MEDICAL example command file. Here is the CONTAGION command and following line from the file:

```
CONTAGION  3         1     1   1
11111111111111110100000000000100 IDENTIFY PRESCRIPTION SAMPLE
```

There are 18 physicians identified with a 1 in the second line as the people to be studied for contagion effects. These 18 were deemed most likely to find a use for the new drug studied by Coleman, Katz and Menzel (1957), so local pharmacy prescriptions were searched for records of them writing prescriptions for the new drug. The other 14 physicians were interviewed for sociometric and background data, but not included in the prescription search. They are in the study to flesh out the social context around the 18 study physicians. The summary results on pages 162-164 are an analysis of the selected 18 physicians in the broader context of all 32 physicians. To estimate contagion without the four outlier physicians (top of page 164), I used the ASSISTANT program to deselect physicians 3, 8, 11 and 14, and re-ran the analysis across the remaining 14 physicians.

Missing Response Data

The program won't accept missing network data, but it has limited capabilities of working around incomplete response data. Command lines and input data are listed on the next page for the six-person network at the top of page 168. This is the fourth analysis in the example command file CONTAG.

The 1 in column 22 of the CONTAGION command tells the program that some responses are missing. The program then expects to see a missing data record just after the response data format statement. This is the line labeled MISSING DATA on page 172 (cf. line 35 in the MEDICAL.DAT input data file). The label is only included to help you identify the missing data record. The program reads the missing data record like any other, but stores the scores

```
TITLE      Example System with Missing Data
DATA       CONTAG.DAT
NETWORK      6 D
ID CODES
     Sam    Joan    John    Bill    Gert    Zeke
CONTAGION    2     2 1 1.0 2
ANALYZE
```

This is the input data file CONTAG.DAT:

```
(6F1.0)
 01010
1 0101
01 110
000 00
1000 0
01000
(F1.0,2X,F2.1)
9  00      MISSING DATA
4  41      Sam’s responses
1  32      Joan’s responses
9  50      John’s responses
9  49      Bill’s responses
5  51      Gert’s responses
0  20      Zeke’s responses
```

and treats subsequent scores as missing when they match the missing data record. In the six-person network, 9 is a missing data code on the first variable. John and Bill have no response on the first variable. No one is missing on the second variable.

With missing responses defined, two things happen. First, deletion is listwise. Any ego missing a response on one criterion variable is deleted as an ego and retained only as a possible alter. Any alter for whom a missing response on one variable cannot be imputed is deleted from the analysis. Don't mix in the same analysis criterion variables on which all responses are known with variables missing responses. Run the variables separately.

Second, when an alter's response is missing, the program looks for someone to speak for the alter. Given y_j missing, the program finds the largest w_{jq}, and uses y_q to fill in the missing y_j. In other words, missing alter responses are imputed as hypothesized by the contagion model being testing.

The imputation is constrained in certain ways: (a) Ego is excluded from the people who can speak for the missing alter. Otherwise, y^*_i could be built in part from the y_i it predicts. This means that missing data have to be re-imputed for each ego, which slows the computations. (b) Only one person speaks for the missing alter. If there are two or more people q equally suitable to speak for missing alter j, one of them is picked at random. This seems preferable to averaging their responses, which would create a regression toward the mean in alter responses. (c) If no one can be found to speak for missing alter j, j is deleted as one of ego i's alters, and the w_{ij} are renormalized to sum to 1 excluding j. (d) If more than half of ego's alters are missing responses and cannot be imputed, ego is coded as missing, and y^*_i is replaced with the missing data code for y_i.

The percentage of alter response missing is output so that you can test for effects:

$$100(\Sigma_j w_{ij} \delta_j),$$

where δ_j is 1 if y_j is missing, 0 otherwise. This measure varies from 0 (if all i's alter responses are known) to 100 (if none of i's alter responses are known). Use the measure as a covariate to check for change in the contagion slope with change in missing alter response.

Consider the printout on Sam on page 168. John's response on the first variable is unknown. That makes 50% of Sam's alter response unknown on the first variable as reported in the printout. The program looks for the people to whom John sends relations; Joan, Bill and Gert. Joan is picked at random. Joan gave a response of 1 on the first variable. In the page 168 printout, a response of 1 is reported for alter John, but the response has an asterisk and a 2 in parentheses above it. John's response of 1 is imputed from the response of actor 2, Joan.

That's how it works without a problem. The dyadic printout on John shows what happens when there is a problem. Portions are displayed below. John cites three alters listed as equally important in the printout (Joan, Bill and Gert with network weights of 1/3 each; cf. the sociogram at the top of page 168). Joan and Gert provide responses on the first variable, but Bill doesn't. A third of John's alter response is missing (33.3% in printout). However, there are

```
  John's CONTAGION RESULTS

--- ALTER ---  NETWORK         DATA ON EACH RESPONSE VARIABLE
  ID   NAME    WEIGHT      1        2         3         4
------------------------------------------------------------------


   2   Joan     .333     1.000    3.200

   4   Bill    .333     THIS ALTER COULD NOT BE IMPUTED

   5   Gert     .333     5.000    5.100


% Alter Response Missing ----------------------------------------
                          33.3%    33.3%

Ego Expected Response -------------------------------------------
                         3.000    4.150
```

no alters to speak for Bill under the contagion model being tested. Bill cites no one. Therefore, a message is printed stating that missing data on Bill couldn't be imputed (note the listwise deletion), and the network weights are renormalized to sum to 1 across the remaining alters. The response expected of John on the first variable is 3; the average of Joan's response of 1 and Gert's response of 5.

The ego-level printout for the six-person network on the first criterion variable is given below (put a 1 in column 28 of the CONTAGION command). Half of Sam's alter response is missing. A third of Joan's and John's are missing. Bill has no alters (self-weight is 1 and number of alters is 0), so y^*_i is set equal to the missing data code, 9, on the criterion variable.

```
CONTAGION RESULTS with alters defined by cohesion, and the exponent
nu set to a value of 1.0:

    ID = RESPONSE VARIABLE, SEQUENTIAL ID NUMBER FOR EGO, EGO NAME
     X1 = SELF-WEIGHT
    X2 = NUMBER OF ALTERS (Wij > .001, excluding self)
    X3 = PERCENT ALTER RESPONSES MISSING
     X4 = EGO RESPONSE (Yi)
    X5 = AVERAGE ALTER RESPONSE (SUMj Wij*Yj)
    X6 = STANDARD DEVIATION OF ALTER RESPONSE

IDENTIFICATION   X1    X2    X3        X4         X5         X6

1   1     Sam  .000    2  50.0     4.000      3.000      2.000
1   2    Joan  .000    3  33.3     1.000      2.000      2.000
1   3    John  .000    3  33.3     9.000      3.000      2.000
1   4    Bill 1.000    0    .0     9.000      9.000      9.000
1   5    Gert  .000    1    .0     5.000      4.000       .000
1   6    Zeke  .000    1    .0      .000      1.000       .000
```

Monte Carlo Analyses

There are several options for studying contagion through simulations. The network can be simulated (column 16 of the CONTAGION command) and the response data can be simulated (column 18). The six classes of contagion analyses possible with STRUCTURE are described in a summary way on pages 26-27 with referrals to example command files on the program disk. The simulation options are provided to answer two kinds of questions; inference questions and design questions (see the ASSISTANT *Command Booklet* for an illustrative application, pages 67ff; or Xia, 1991, for a full study).

Option 1 in column 18 is unusual, and so warrants brief explanation. Network and response data are read as usual, then the response data are randomly redistributed before analysis begins. The redistribution is across individuals selected to be ego: (a) A random number is drawn from the uniform distribution for each ego. (b) The random numbers are sorted by magnitude. (c) The response of the ith person in the input file is assigned to the person in the ith rank of the random numbers. Everything is the same as in the usual contagion analysis; same network structure defining alters, same distribution of responses, but the response data predicted by contagion are randomly rearranged. This is the null condition of no network correlation in the exact network context and response distribution from which actual contagion is measured. Repeated trials will create a sampling distribution around the null hypothesis tailored to the study population network structure and response distribution. Armed with this sampling distribution, more accurate statements can be made about the strength of the observed contagion effect discussed in the preceding pages.

AUTONOMY

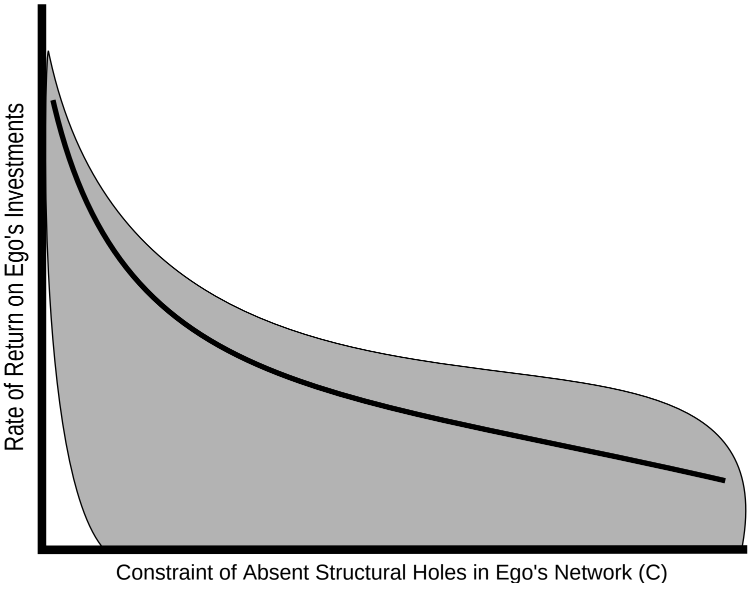
The image here is of an individual reaching diverse social worlds to broker contact between them. The image is familiar in Granovetter's (1973, 1983) weak tie metaphor, Freeman's (1977; Freeman et al., 1980) model of betweenness centrality, Cook and Emerson's (1978; Cook et al., 1983) model of exchange power (Marsden, 1984; Markovsky et al., 1988), Burt's (1980, 1982, 1983b, 1988b) model of structural autonomy, and more abstract network images of social capital (e.g., Berkman and Syme, 1979; Lin, 1982; Lin et al., 1981; De Graaf and Flap, 1988). The information and control benefits of reaching diverse social worlds to broker contact between them is the basis for Burt's (1992a) structural hole theory of competition, which provides the network variables available with the AUTONOMY command. Information and control benefits come to those who broker connections between contacts in separate social worlds — on opposite sides of a structural hole. These information and control benefits lead to higher levels of performance and achievement, by people as well as organizations. The summary conclusion is that individuals rich in structural holes know about, have a hand in, and exercise control over, more rewarding opportunities.

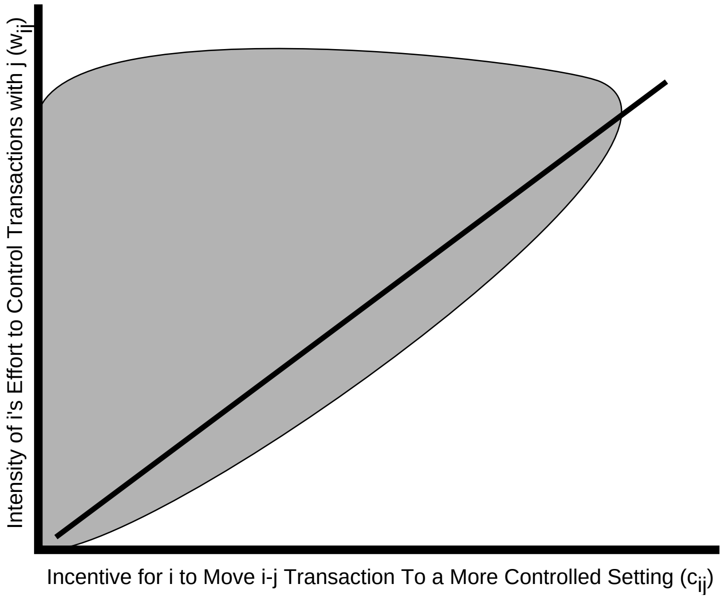
ANALYSIS

There are diverse analyses possible with the AUTONOMY command, but I'll discuss two generic ones for illustration. The nonzero elements in row and column i define a network of contacts, alters, around the individual, ego, distinguished by the row and column. The key variable here is constraint; interlocked relations that limit ego's freedom of movement in the aggregate and with respect to each ego-alter relationship.

The first line of analysis is to correlate the several measures of aggregate constraint generated by STRUCTURE with measures of ego qualities and outcomes. Range and brokerage measures are written to an output file (1 in column 20 of the AUTONOMY command) and merged with other ego variables. Individuals in low constraint networks are expected to show higher rates of return on their efforts (higher performance and achievement) because of knowing about, having a hand in, and exercising control over, more rewarding opportunities. The are also expected to have the qualities needed to exercise control in a socially diverse environment; more motivation to be entrepreneurial, higher intelligence, less dependence on stereotyping, more leadership qualities. Complexities in the association are illustrated in graph at the top of the next page (cf. Burt, 1992a:Fig. 1.7). The vertical axis shows rate of return (e.g., an organization's profit margin, or a person's income return to seniority) and the horizontal axis shows the constraint of decreasing structural holes around ego's alters. Three points are illustrated: (1) The solid line shows the negative association; performance decreases with increasing constraint. (2) The association is nonlinear. The strongest effects are at low levels of constraint. The association can be linear in a study population of high constraint networks, but is nonlinear across the full range of constraint (see Burt, 1992a, for examples of both). (3) Deviations from the association are expected more in one direction than the other. As indicated by the shaded area in the graph, observations are expected around the solid line, and below it, but rarely above it. Individuals unconstrained by their network have freedom of choice to develop the network benefits of information and control. Some can choose not to. They lie below the solid line in the figure — not realizing the full potential of their network. Individuals without opportunities for information and control benefits do not have the choice — the area above the line is almost empty. Observations in the upper-right of the graph are particularly interesting. These are high achivement individuals in constrained networks; social support is the key to their achievement (see Burt, 1992a:Chap. 2, on sponsorship and manager achievement).

A second line of analysis is to predict how ego responds to constraint in specific relationships. Dyad data on constraint are written to an output file (2


Rate of Return on Ego's Investments
Constraint of Absent Structural Holes in Ego's Network (C)


Intensity of i's Effort to Control Transactions with j (w_{ij})
Incentive for i to Move i-j Transaction To a More Controlled Setting (c_{ij})

in column 20 of the AUTONOMY command) and can be merged with data on other relations between ego and alter. Let w_{ij} be a measure of i's investment in a strategy to manage the constraint c_{ij} from alter j. Examples are corporate hierarchy ties spanning market transactions with suppliers and customers, or personal obligation ties spanning dependence ties. The prediction is that w_{ij} increases with the severity of constraint (c_{ij}) and decreases with the opportunity (p_{ij}-c_{ij}) that might be lost if the network is changed (see Burt, 1992a, Chap. 7, for elaboration; p_{ij} defined on next page). The w_{ij} are embedding relations (Granovetter, 1985). Here again, the association can show complexities illustrated in the graph at the bottom of the preceding page (Burt, 1992a:Fig 7.3). Three points are illustrated: (1) The bold line shows w_{ij} increasing with c_{ij}, but (2) many authority or obligation relations are created where they aren't needed (shaded area in upper-left), and (3) few constrained relations aren't embedded (empty area in lower-right of graph). Observations in the lower-right are particularly interesting. They are severely constrained relations that seem to have been ignored. Other embedding relations, or other control strategies are being used to manage these (Burt, 1992a:Chap. 7).

RANGE AND BROKERAGE MEASURES

The analysis begins by aggregating input relations across kinds ($z_{ij} = \Sigma_k z_{ijk}$). The aggregate relations are used to build the network of contacts around a specific individual, ego, selected for analysis. You define who gets analyzed as ego in column 12 of the AUTONOMY command as described on page 20. You can analyze the network range and brokerage of each individual in the input system, or analyze a single individual.

Given an individual i to be analyzed as ego, eight variables are produced to describe i's network. The first is ego-network size, N; a count of ego's contacts. This is everyone connected with i (default option), or everyone to whom i sends a relation (1 in column 14 of AUTONOMY command).

Second, nonredundant contacts is a count of the number of independent contacts in ego's network. If ego has any contacts, this varies from 1 (when each contact is strongly tied to each other) to N (when each contact has no connection with any other; see Burt, 1992a:Eq. 2.2, for further details):

$$\text{nonredundant contacts} = \Sigma_j[1 - \Sigma_q p_{iq} m_{jq}], \qquad q \neq i, j$$

where the summation is across all of i's N contacts, p_{iq} is the proportion of i's network time and energy invested in the relationship with q (interaction with q divided by the sum of i's relations);

$$(z_{iq}+z_{qi})/[\Sigma_j(z_{ij}+z_{ji})], \qquad i \neq j$$

and m_{jq} is the marginal strength of the relation from contact j to q (interaction with q divided by the strongest of j's relations with anyone);

$$(z_{jq}+z_{qj})/\max(z_{jk}+z_{kj}), \qquad j \neq k$$

where $\max(z_{jk}+z_{kj})$ is the largest of j's relations with anyone ($0 \leq m_{jq} \leq 1$). Third, contact efficiency is nonredundant contacts divided by total contacts:

$$\text{contact efficiency} = (\text{nonredundant contacts})/\text{contacts}.$$

This is closely related to the two network density measures, the fourth and fifth variables in the analysis (see Burt, 1992a, footnote 5 in Chapter 2, on efficiency and density). The fourth variable is the average marginal strength of relations between contacts:

$$\text{network density} = [\Sigma_j\Sigma_q\, z_{jq}/\max(z_{jk})]/[N(N-1)], \qquad j \neq q$$

where $\max(z_{jk})$ is the largest of j's relations to anyone, so density varies from 0 (no relations between contacts) to 1 (maximum strength relations between all contacts). The fifth variable is the proportion of contact pairs that have some kind of connection with one another:

$$\text{proportional density} = (\sum_j \sum_q \delta_{jq})/N(N-1), \quad j \neq q$$

where δ_{jq} is 1 if z_{jq} is nonzero, otherwise δ_{jq} equals 0. This proportional measure of density varies from 0 (no relations between contacts) to 1 (every pair of contacts is connected). In a network of contacts all connected by weak relations, network density is low and proportional density is high.

The final three variables define constraint. Oligopoly, O_i, varies from 0 to 1 measuring the lack of structural holes around individual i. This is either 1 by default, or read from the input data file (1 in column 16 of AUTONOMY command; see page 20 for illustration, Burt, 1992a, Chap. 2 for alternative measures). Next, contact j is a source of severe constraint on ego to the extent that j has exclusive relations with ego's other contacts and there are no substitutes for j (see Burt, 1992a: Eq. 2.4, 2.6 and 2.7 for elaboration):

$$c_{ij} = (p_{ij} + [\sum_q p_{iq} p_{qj}])^2 O_j, \quad q \neq i, j$$

which is the horizontal axis in the graph at the bottom of page 179. The sum of this dyadic constraint across contacts is the aggregate constraint on ego (the horizontal axis in the graph at the top of page 179):

$$\text{constraint} = C = \sum_j c_{ij}.$$

Constraint varies from a maximum of 1 (when ego has only one contact) to a minimum of 0 (when ego has many, disconnected, easily replaced contacts). Finally, hierarchy describes the distribution of constraint across relations. The following ratio measures the extent to which constraint on the i-j relation is higher than the average across i's relations (cf. Burt, 1992a:Eq. 2.8):

$$c_{ij}/[C/N]$$

and the following index — the Coleman-Theil disorder index — uses the ratio to measure hierarchy:

$$\text{hierarchy} = \frac{\sum_j \left(\frac{c_{ij}}{C/N}\right) \ln \left(\frac{c_{ij}}{C/N}\right)}{N \ln(N)},$$

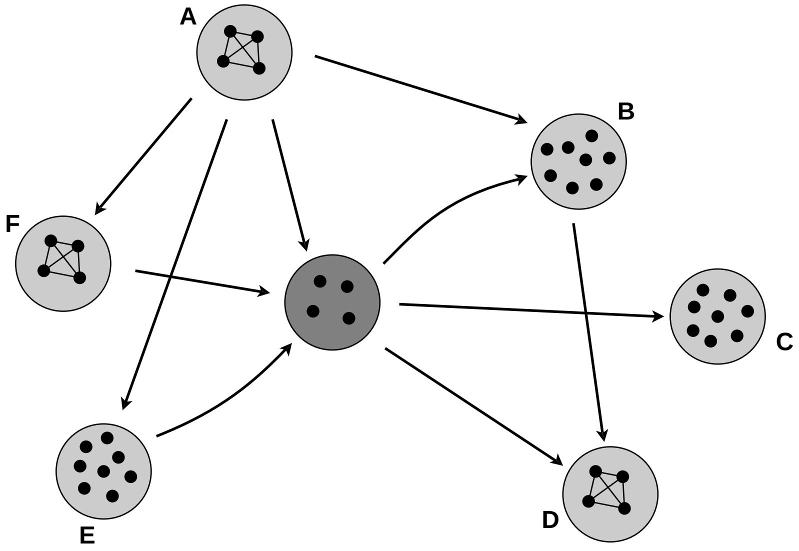
where summation is across all N alters and hierarchy is the extent to which constraint is concentrated in the hands of a dominant contact (cf. Burt, 1992a:Eq. 2.9). Relative constraint equals 1 when j poses constraint equal to the average. Hierarchy varies from 0 (equal constraint from each contact) to 1 (when all constraint comes from a single contact). Hierarchy and density are set to 0 for isolates. They are set to 1 if ego has only one contact. You may wish to handle small networks in some other way.

OUTPUT*

The AUTON example command file produces illustrative results discussed in Burt (1992a, Figures 2.2, 2.5 and 7.4). The sociogram for Figure 2.5 is presented on the next page with the eight variables describing the range and brokerage opportunities of a dot in the dark circle at the center of the network (cf. Burt, Figures 2.2 and 2.5). Gray circles are contact clusters with internal organization indicated by dots and lines.

There are six contacts in the network, but connections among the contacts reduce those to four and two-thirds nonredundant contacts. The ratio of

*An equivalence analysis of role components can be an interesting complement to these measures of brokerage opportunities. The POSITIONS command can be used to generate a count of the times than an individual is ego in triads 3, 5, 7, 8, 9 and 10 on page 130. These are triads in which ego brokers a connection between two alters. These counts are cruder than the range and autonomy measures because they treat all nonzero relations as equally strong and ignore crucial variation in exclusive ties between contacts (see Burt, 1992a, Chap. 2), but they provide interesting details on the manner in which an individual brokers the connection between others (e.g., through mutual ties to the contacts as in triad 7, or being sought out by others as in triad 5, or being an intermediary in a chain between the others as in triad 8). This line of measurement is discussed by Gould and Fernandez (1989).



```
    6  Contacts
4.667  Nonredundant contacts
 .778  Contact efficiency
 .133  Average relation between contacts (network density)
 .133  Portion ties between contacts (proportional density)
 .250  Oligopoly (Oi; lack of structural holes around ego)
 .264 Constraint (sum of Cij; lack of holes around contacts)
 .332  Hierarchy (constraint from dominant contacts)
```

nonredundant to observed contacts is the .778 contact efficiency. The relations are binary, so the two density measures are identical; four of 30 relations between contacts exist (13.3%). Oligopoly is read from the input file, here defined as the maximum proportion of individuals within a cluster connected to others. In the dark gray circle, that is 25%. In cluster A, that is 100%. There is modest constraint, and a high level of hierarchy (cf. Burt, 1992a, Figures 3.9 and 4.8)

The eight variables are the default printout for an analysis. A 1 in column 20 of the AUTONOMY command writes the indices to the output data file so they can be merged with data on correlates of range and structural autonomy for analyses illustrated by the graph at the top of page 179.

For a closer look at who is responsible for the constraint, put a 1 in column 18 of the AUTONOMY command. This generates details on how opportunity and constraint are distributed across ego's relationships. The following come from the analysis on the preceding page:

```
ID      NAME        Cij                Pij                Oj

 1    Contct A    .1512    3.4430    .1667    .2222    1.000
 2    Contct B    .0106     .2421    .1667    .1250     .125
 3    Contct C    .0035     .0790    .1667    .0000     .125
 4    Contct D    .0494    1.1242    .1667    .0556    1.000
 5    Contct E    .0054     .1235    .1667    .0417     .125
 6    Contct F    .0434     .9881    .1667    .0417    1.000
```

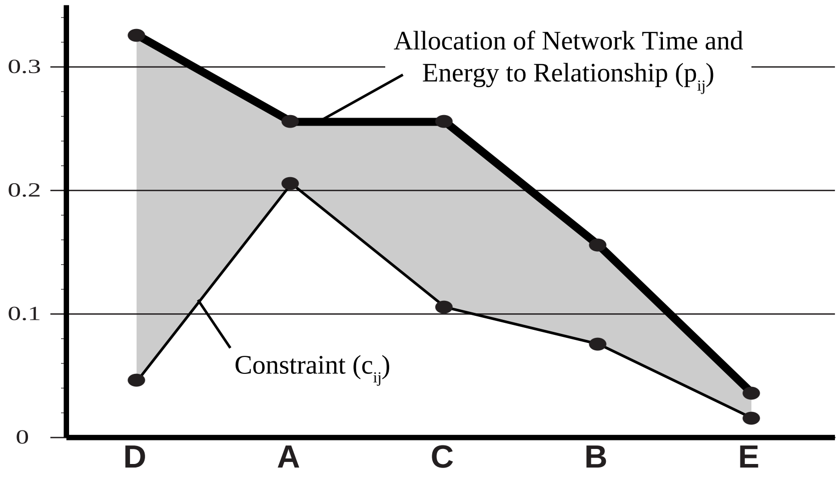
The six contacts are listed by row in order of their appearance in the input data file. The constraint posed by each is listed next to the contact name, followed by the ratio of the constraint to the average per contact (c_{ij}, and c_{ij}/(C/N), respectively). The next two columns give the direct and indirect portion of ego's network time and energy invested in relations that lead back to the row contact. In this example, direct connections are all the same strength (p_{ij} all

1/6). Indirect connections vary with the exclusive relations each contact has with others ($\Sigma_q p_{iq} p_{qj}$). Contact A is best connected. Contact C is not at all connected. The last column is the oligopoly variable from the input file. Everyone in cluster A is connected to everyone else ($O_A = 1.0$). None of the eight individuals in cluster C are connected ($O_C = 1/8 = .125$). The fifth, sixth, and seventh columns are the components that jointly define contact-specific constraint c_{ij} on page 182.

A 2 in column 20 of the AUTONOMY command writes the above dyad data to the output data file with each row preceded by the sequential ID of the individual being analyzed as ego (e.g., fifth person in the input data file), and the oligopoly score for the individual (O_i). These dyad scores can inform research on strategies used to manage constrained relationships. The components of dyad constraint c_{ij} makes it easier to study the extent to which direct contact, indirect contact, and oligopoly contribute individually to the aggregate incentive illustrated on page 179 to embed c_{ij} in w_{ij}.

The hole signature of the network is also printed if dyad results are requested (1 in column 18 of AUTONOMY command). The illustrative hole signature on the next page is taken from Burt (1992a, Figure 2.6). The top bar is defined by individual i's allocation of network time and energy to each relationship (p_{ij}) and the bottom bar is defined by the constraint on each relationship (c_{ij}). Contacts are listed in descending order of the strength of their relationship, then constraint. Contacts A and C (no relation to alphabetic IDs in previous example) represent equal strength relationships, but A poses more severe constraint and so is listed first. This would happen if A has more exclusive relations with the other contacts (higher $\Sigma_q p_{iq} p_{qj}$) or fewer competitors beyond the network (higher O_A). The shaded area is the hole signature of the network. Opportunity relations are indicated by a wide bar in the signature

(p_{ij} high, c_{ij} low; such as contact D). Constraint relations are indicated by a high, narrow bar in the signature (p_{ij} high, c_{ij} high; such as contact A). The hole signature summarizes the distribution of opportunity and constraint across individual relationships in a network. With it, individuals can be identified, studied and compared for their entrepreneurial opportunities (see Burt, 1992a, Figures 2.6, 2.7, 3.7, 4.7, 7.4 and 7.5, for examples and elaboration). To prepare camera-ready hole signatures, put a 3 in column 20 of the AUTONOMY command. Contact sequence in the hole signature, sequence in the input system, name, c_{ij} and p_{ij} will be written to the output data file, from which they can be read by a graphics program.



POWER

Where range and brokerage concern the freedom an individual has to act within the limited sphere of her contacts, power concerns her ability to dominate the whole system across spheres. Where range and brokerage concern the horizontal axis of organization, power concerns the vertical; strong at the top, weak at the bottom. The general idea is demand. An individual who is the object of relations has something of interest to everyone sending the relations. That interest makes the individual prominent and gives her power. Some individuals have a lot of it. Others have very little. This vertical axis of organization is broadly conceived and encompasses the prominence concepts of centrality, prestige, social resources, demand and value. Knoke and Burt (1983) review kinds of measures and answer some practical questions. The POWER command gives you several prominence measures, from the most simple to the sophisticated. The selection is broadened with the NETWORK command options for transforming relations.

ANALYSIS

There are diverse analyses possible with the POWER command, though common practice takes little advantage of what is possible. Two generic lines of analysis are at the ego and dyad levels. The obvious use of power scores is to predict winners. Power increases the probability of winning. Write the prominence measures to the output data file (1 in column 24 of the POWER command), merge with other data on the individuals, and sum scores on each side of a conflict. The side with the higher aggregate score is the more powerful and should win, or should have won, the conflict (e.g., Laumann and Pappi, 1976). At the level of ego-alter dyads, power predicts ego's ability to change relations to suit her interests — an equilibrium relation $\mathbb{Z}_{ij}$ can be derived from the observed relationship z_{ij} and the distribution of power. The observed and equilibrium relations can be written to the output data file (3 in column 24 of the POWER command) to study the association between the equilibrium $\mathbb{Z}_{ij}$ and observed z_{ij}. Summary measures of association are included in the default STRUCTURE printout. Where the equilibrium and observed relations are strongly correlated, observed network structure is at equilibrium. If observed structure is at equilibrium, then descriptions of cliques and positions in the structure are more interesting, and network predictions of opinion and behavior are more credible. Or, alternative equilibria can be derived to see which most closely resembles the observed relations. This search among equilibria is a search for the exchange mechanism most likely responsible for the observed network structure.

PROMINENCE MEASURES

As in the preceding section, the analysis begins by aggregating input relations across kinds ($z_{ij} = \sum_k z_{ijk}$). Choice status, sociometry's early measure of popularity, is the least presumptuous of prominence measures. It is the number of actors who reach ego divided by the number who could have done so:

$$\text{choice status of } i = \sum_j \delta_{ji}/(N\text{-}1), \quad j \neq i$$

where N is the number of actors in the whole system (not just those connected to i), and δ_{ji} equals 1 if j can reach i ($z_{ji} > 0$), otherwise δ_{ji} equals 0. Choice status varies from 0 (when no one reaches i) to 1 (when everyone else reaches i). This can be misleading when two individuals are the object of relations from the same number of others, but one receives weak relations while the other receives strong relations. The next step in sophistication is to weight relations by their strength (where $\max(z_{jk})$ is j's strongest relation to anyone else):

$$\text{extensive relations to } i = \sum_j [z_{ji}/\max(z_{jk})]/(N\text{-}1), \quad j \neq i, k$$

This measure varies from 0 (when i receives no relations) to 1 (when i is the object of a maximum strength relation from every actor). This variable measures extensive prominence in the sense that actor i is the object of strong relations from everyone. This too can be misleading. A member of a completely connected clique has a high score on extensive prominence — but so does everyone else in the clique. Clearer distinction between top and bottom of the social structure is captured by exclusive prominence:

$$\text{exclusive relations to i} = \Sigma_j \, [z_{ji}/\Sigma_k z_{jk}]/(N\text{-}1), \quad j \neq i, k$$

where the bracket now contains the proportion of j's network time and energy allocated to interaction with i. This variable measures the extent to which actor i is the object of exclusive relations from everyone. It varies from 0 (when i receives no relations) to 1 (when i is the only contact for every other actor). This too can be misleading. Power comes more from exclusive relations with powerful players than exclusive relations with weak players.

The final step is to weight exclusive relations by the power of their source such that actor i is powerful to the extent that she is the object of exclusive relations from powerful others:

$$\text{power of i} = p_i = \Sigma_j \, [z_{ji}/\Sigma_k z_{jk})]p_j,$$

where the sum is across all N actors, including ego. The relation in brackets is a proportional strength relation from j to i. It measures the extent to which i has an exclusive relation from j. But these are not the relations in the exclusive relations variable; self-relations are included in this calculation. If self-relations are arbitrary constants, as presumed by default, then z_{jj} is set equal to j's strongest relation to anyone else. If self-relations are measured variables (1 in column 22 of the NETWORK command; see page 38), then z_{jj} is read from the input file and scaled to a proportion, just like j's relations with everyone else. In either case, actors who send no relations to others are treated as keeping all relations to themselves — z_{jj} is set to 1 if all entries in row j are zero.

Thus, the network is a row-stochastic matrix (every row of the matrix sums to 1.0), and I'll refer to these as row-stochastic relations. In such a matrix, the first eigenvalue is 1 and p_j is an element in the corresponding left-hand eigenvector (see the equilibrium model below and Burt, 1982:35-37, for details connecting the model to Hubbell, 1965, and Coleman, 1972). Power here corresponds to price in the general equilibrium model of economic markets. Interaction with i is expensive, valuable, to the extent that i is the object of exclusive relations from people whose interaction is highly valued. This is the metaphor used below to predict what the network would look like at equilibrium if everyone were able to exchange the relationships they have for the relationships they want. Katz (1953) first used the eigenvector solution in measuring network prominence, but Hubbell (1965) provides the first clear analogy to the general equilibrium model applied in input-output analysis, and Bonacich (1972) provides the most clear exposition of computing eigenvector prominence scores from network data. Coleman (1966, 1972, 1973, 1986) provides the most articulate theoretical analogy to market processes. Coleman's exchange model is the basis for the equilibrium calculations below. Power, prestige, or price, p_i increases with the demand for i's network time and energy.

The fifth prominence variable in the POWER output is Mizruchi et al.'s (1986) distinction between derived and reflected components in the basic model (see Mintz and Schwartz, 1985, for detailed substantive application). Power is of two parts: (1) Part comes from i receiving relations she doesn't reciprocate. (2) The rest comes from mutual relations; alters she contacts who in turn contact her. The latter part is reflected power in the sense that the mutual relation between ego and alter reflects power on one another. The two parts sum to p_i, so only one is in the output. The following is the percentage of ego's power that is reflected:

$$100*\sum_j z_{ij} z_{ji}, \quad j \neq i$$

where z_{ij} is the row-stochastic relation from i to j used to compute power on the preceding page. The product is the portion of ego's total interaction that is allocated to alter j and comes back through j's strong relation to ego. One hundred minus this amount is the percentage of ego's power that is derived from relations she receives but doesn't reciprocate. Note that this is the percentage of ego's power that is reflected; not the level of reflected power (the latter, if you want it, is p_i times the percentage reflected, divided by 100).*

I have two final notes on the power variable. There are options for scaling it, and the iterative computation can fail to converge.

The power variable is an eigenvector. It has a unique solution (up to a scalar) and all elements in the eigenvector will be positive (under conditions discussed in the next paragraph). The program uses the most powerful actor as a numeraire for scaling levels of power. The power of each individual is the ratio of p_i over the power of the most powerful actor as numeraire, p_n;

$$p_i = p_i/p_n.$$

Thus, the power scores vary by default from a maximum of 1, for the numeraire and equally powerful actors, toward 0 for the weakest actors. The scores are equally valid if expressed with respect to some other actor as numeraire. For example, if the fifth actor in the input data file had special substantive significance as a reference point for discussing power, you could rescale the power variable by dividing by p_5. On the rescaled power variable, a score of 2 indicates someone with twice the power of actor 5, and a score of .5 indicates someone with half the power of actor 5.

*Note also that this simple expression for reflected power depends on the first eigenvalue being 1, which it is in the row-stochastic matrix used here. Because it is always 1, the percent reflected power can be computed (and is presented in the printout) even if the eigenvalue for power cannot be computed. More generally, the eigenvalue need not be 1, and a more complex equation is required (see Mizruchi et al., 1986).

Computational problems can arise if the network contains disconnected subsystems (e.g., uncited individuals). If the eigenvector contains negative elements, the program adds the magnitude of the most negative element to every element before scaling elements relative to the numeraire. A warning message is printed and the original eigenvector elements also appear in the printout. This destroys the relative magnitude of the eigenvector elements, so no stability results are presented (as described in the equilibrium section below). Also, the iterative search for a solution to the eigenvector equation might not converge to within the allowed tolerance in the allowed number of steps (see the DEFAULTS command on pages 31-32 for changing the steps or tolerance). If any convergence problem occurs, a warning message is printed and all power scores are set to zero.

OUTPUT*

The POWER example command file requests three analyses. The second and third are large analyses of elite experts in sociological methodology and American markets. The first is an analysis of prominence in the simple figure on page 184 ignoring structure within clusters and making all ties symmetric between the gray dots (just like the first analysis in the AUTON example command file).

The results of the first analysis are presented at the top of the next page. The eigenvalue is 1.0, which it should be. Choice status and extensive relations are the same here because all relations have the same weight. Contact A is connected with two-thirds of the network, contact B with half. Scores under

*Again, an equivalence analysis of role components can be an interesting complement to the prominence measures. The POSITIONS command can be used to generate a count of the times that an individual is in the triads on page 130. For example, if power is correlated with the frequency of ego being in triad 5, then power is associated with having a disorganized constituency. Or, power increasing with triad 32 and decreasing with triad 22 means that power is correlated with being at the top of a chain of command.

```
PROMINENCE ANALYSIS OF INPUT RELATIONS

(MAXIMUM EIGENVALUE:  1.000 IN 10 ITERATIONS)

   X1 = CHOICE STATUS (portion others reaching ego)
   X2 = EXTENSIVE RELATIONS (mean marginal strength received)
   X3 = EXCLUSIVE RELATIONS (mean proportional strength received)
   X4 = POWER (eigenvector; numeraire is most powerful actor)
   X5 = PERCENT POWER REFLECTED (interaction out that comes back)

ID      NAME        X1         X2        X3        X4       X5

 1    Contct A    .6667      .6667     .2500     .7140    21.19%
 2    Contct B    .5000      .5000     .1528     .5712    16.90%
 3    Contct C    .1667      .1667     .0278     .2861     7.14%
 4    Contct D    .3333      .3333     .0833     .4285    13.10%
 5    Contct E    .3333      .3333     .0694     .4284    11.43%
 6    Contct F    .3333      .3333     .0694     .4284    11.43%
 7         Ego   1.0000     1.0000     .5139    1.0000    27.86%
```

X3 are lower than under X2 because relations are not equally exclusive in the network. For example, contacts D and E have equal choice status, but D is slightly higher on the exclusive relations variable because his relation from B is more exclusive than E's relation from A. Power is correlated .99 with choice status in this network. Though the prominence measures can vary substantially, my experience has been that they yield very similar correlations with criterion variables in substantive research. More dramatic than differences between them when computed from the same relations are differences between them when they are computed from different transformations of the relations (e.g., making relations symmetric with the NETWORK command, see Knoke and Burt, 1983).

The five prominence indices are the default printout for an analysis. A 1 in column 24 of the POWER command writes the scores to the output data file so they can be merged with data on correlates of power.

EQUILIBRIUM

Will the relations observed today be there tomorrow? Network analysts often presume that social structure endures. Network equilibrium models turn the presumption into an empirical question. The models use the distribution of power to predict how relations would change if powerful people could get the interaction they wanted. Then, the predicted equilibrium relations are compared with observed relations to determine aggregate stability and locate the most unstable classes of relations. Equilibrium under independence is generated automatically since the computational work is largely finished once power is computed.

Equilibrium under Independence

Independence equilibrium is a direct application of James Coleman's exchange model (Coleman, 1966, 1972, 1973, 1986; cf. Taylor and Coleman, 1979). Equilibrium relations are derived by imagining that actors can buy the interaction in which they are interested by selling their own interaction to anyone who is interested. Actor i enters the market with interest in getting interaction from certain actors and a capacity to buy that interaction with her own network time and energy that has a certain value or worth, w_i. I'll take the proportional strength of i's observed relation to j, the row-stochastic relation z_{ij} ($0 \leq z_{ij} \leq 1$, $\sum_j z_{ij} = 1$), as a measure of i's interest in j. I'll further assume that z_{ij} indicates the proportion of i's resources she will expend to get interaction from j (illustrative evidence of allocation proportional to interest can be seen in Michener et al., 1975, 1977). The portion of her resources that i will expend on getting interaction from j is then:

$$w_i z_{ij}.$$

It costs something to get j to interact with i. When she enters the market, ego controls her interaction interests; the z_{ij} are row-stochastic. At market equilibrium, however, the market controls who gets how much of her interaction; the equilibrium relations are column stochastic. Let p_j be the price of getting all j’s network time and energy and let $\mathbb{Z}_{ji}$ be the portion i gets when the system is in equilibrium; $\mathbb{Z}_{ji}$ is a fraction ($0 \leq \mathbb{Z}_{ji} \leq 1$), and the fractions sum across buyers to the total of j's network time and energy ($\Sigma_j \mathbb{Z}_{ji} = 1$). This is j's equilibrium relation to i. The cost of buying the $\mathbb{Z}_{ji}$ amount of interaction is then:

$$p_j \mathbb{Z}_{ji},$$

which, at equilibrium, equals the resources i was willing to spend on interaction from j:

$$w_i z_{ij} = p_j \mathbb{Z}_{ji}.$$

I have to make assumptions to be able to do anything with this. Assume that interaction is bought and sold in a perfectly competitive market such that the value of the network time and energy with which an actor enters the market, w_i, equals the equilibrium price, p_i, of purchasing all her network time and energy. Then the above equilibrium equation is:

$$p_i z_{ij} = p_j \mathbb{Z}_{ji},$$

and summing across actors i yields:

$$\Sigma_i \, p_i z_{ij} = \Sigma_i \, p_j \mathbb{Z}_{ji} = p_j(\Sigma_i \mathbb{Z}_{ji}).$$

The portions of j’s interaction (in parentheses) sum to 1, so the price of j’s network time and energy, p_j, is the sum of interest in getting his interaction weighted by the price of those interested in it ($\Sigma_i \, p_i z_{ij}$):

$$p_j = \Sigma_i \, p_i z_{ij}.$$

Allowing for the switch in subscripts, this is the equation used on page 190 to define power; z_{ij} is one of the row-stochastic relations in brackets in the power equation. In other words, the j's power in the network equals the price of buying his network time and energy. The power scores presented in the prominence analysis, inserted in the equilibrium equation, define the equilibrium relationship from i to j:

$$\mathbb{Z}_{ij} = z_{ij}\left(\frac{p_i}{p_j}\right).$$

The equilibrium relationship from i to j increases with the extent to which i is observed to send a relation to j, z_{ij}, and i’s power is greater than j’s so i can get what she wants from j, (p_i/p_j). The equilibrium relation is undefined if j has zero power. This can happen if there are minority subsystems in the network (such as a person who receives no relations). All power is given to the majority subsystem. If any actor has zero power, equilibrium relations are not computed.

Output

The default output is two things. First, ordinary least-squares estimates for a linear regression equation predicting equilibrium relations from observed relations are presented as a summary statement of stability of the input network. High correlation means that the equilibrium relations look just like the observed relations, implying that there is little pressure for change in the observed network. Second, a turnover table is presented showing the association between levels of observed relations (rows) and resulting levels of equilibrium relations (columns).

Optionally, the equilibrium relations can be printed (1 in column 18 of the POWER command). Three lines are printed for each row of the network; first the equilibrium $\mathbb{Z}_{ij}$, then the observed z_{ij}, then change to equilibrium ($\mathbb{Z}_{ij}$- z_{ij}). The equilibrium relations can be written as a square matrix to the output data file (2 in column 24), or written by dyad (3 in column 24). The dyad file contains ego-alter identification, followed by p_i, p_j, $\mathbb{Z}_{ij}$, and z_{ij}. The first N lines are self-relations (z_{ii}). The stability correlation and turnover table in the default printout are crude illustrations of the dyad analyses possible.

The first analysis in the POWER example command file shows the network already in an equilibrium state (r = 1.00). The third analysis shows little correlation between observed and equilibrium market relations (r = .10). The second analysis is more interesting. These are advice relations among 52 people prominent in sociological methodology during the mid-1970s (Burt, 1982:Chaps. 3 and 6). The four statistics elite are like a value-sink in the system; they only acknowledge advice from one another and their influence is widely acknowledged by others. Demand for their interaction comes in, but doesn't return to the system. The printout below shows 22% of equilibrium variance predicted from observed relations. Change is primarily zero strength relations becoming slightly nonzero, and vice versa. A few of the strongest

```
STABILITY COEFFICIENT: Here are ordinary least-squares estimates for predicting
the strength of equilibrium relations from the strength of observed relations:

   EQUILIBRIUM Z = .001 + .276*(OBSERVED Z) + RESIDUAL,

which predicts 22.1% of the variation in equilibrium relations (r = .470).  A
high correlation implies that there is little structural pressure for change in
the observed network. Estimates are based on N(N-1) observations.

TURNOVER TABLE TO EQUILIBRIUM: Observed relations are assigned to rows by their
strength, and equilibrium relations to columns.


                    NONE        WEAK       MORE      STRONG        total

   (z = 0) NONE     1326         549          0           0         1875

  (z < .1) WEAK      516         203         11           0          730

          MORE       33           0         14           0           47

(z > .5) STRONG        0           0          0           0            0

         total     1875         752         25           0         2652
```

equilibrium relations (11) are weak in the observed network. A few of the strongest observed relations (33) weaken to zero or near-zero at equilibrium. I wrote dyads to the output data file to find out who is in these least stable relationships. The pattern is that relations among the four elite are the strong relations that survive to equilibrium, relations to the four elite from others weaken because the less powerful cannot command the attention of the more powerful, and the remaining changes are scattered among other experts.

Equilibrium with Substitutions

Power and equilibrium up to this point are all of a kind; actors are presumed to be independent sources of relations. More generally, they are not. For any one person in whom you are interested, there is someone else who will have to do if the first is not available; for example, one doctor substituting in an emergency for your usual doctor. Power and equilibrium can be modelled in this more general situation by allowing substitutions between actors (Burt, 1979). A 1 in column 12 of the POWER command requests prominence measures and equilibrium relations derived with substitutions.

Let ω_{jq} be a proportion measuring the extent to which someone seeking actor q's interaction would accept j as a substitute ($0 \leq \omega_{jq} \leq 1$, $\sum_j \omega_{jq} = 1$). This makes i's interests more complex. Her observed interest in j, the proportion z_{ij}, is distinct from the interest in j — call it x_{ij} — that guides her market behavior. Her market interest seeks out j in lieu of people in whom she is actually interested. Interest in buying j's interaction increases with observed interest in actors q, z_{iq}, for whom j could substitute, ω_{jq}:

$$x_{ij} = \sum_q z_{iq}\omega_{jq}.$$

Consider a quick numerical illustration. Actor i expresses interest in three people; z_{ij} = .2 for A, .2 for B, and .6 for C. The three people are variably substitutable for one another:

$$\{\omega_{jq}\} = \begin{bmatrix} 1.0 & .0 & .0 \\ .0 & .9 & .5 \\ .0 & .1 & .5 \end{bmatrix},$$

which shows that no one substitutes for A, C substitutes to a small extent (.1) for B, and B is a perfect substitute for C. Actor i's interests in the three people allowing for substitutions are; x_{ij} = .2 for A, .48 for B, and .32 for C.

Four points are illustrated: First, interest allowing substitutions is a proportion like observed interest ($0 \leq x_{ij} \leq 1$, $\sum j\ x_{ij} = 1$). Second, market interest equals observed interest if the object of interest doesn't substitute for any other (person A). Third, market interest is greater than observed interest if the object of interest could substitute for someone of greater interest (person B). Third, market interest is less than observed interest if the object of interest could substitute for someone of less interest (person C). In essence, allowing substitutions homogenizes demand for substitutable actors.

That's it. Replace z_{ij} with x_{ij} in the equilibrium equations on pages 196-197 to define power and equilibrium relations:

$$p_j = \Sigma_i\ p_i x_{ij}, \quad \text{and} \quad \mathbb{Z}_{ji} = x_{ij}(p_i/p_j).$$

Note that power has to be recomputed to allow for substitutions between actors. It's based on the x_{ij} instead of the z_{ij}. Power is now a function of being the object of strong relations — or a substitute for someone who is the object of strong relations — from powerful actors.

Second, the power scores and equilibrium relations under independence are a very special case of the more general model. They describe social systems in which no substitutes are accepted; all ω_{jq} are zero except w_{jj} which equals 1. When the ω_{jq} are so defined, x_{ij} equals z_{ij} and the above equations reduce to the equations on pages 196-197. But when one or more of the off-diagonal ω_{jq} are nonzero, indicating that some actors can substitute for others, the independence model is incorrect.

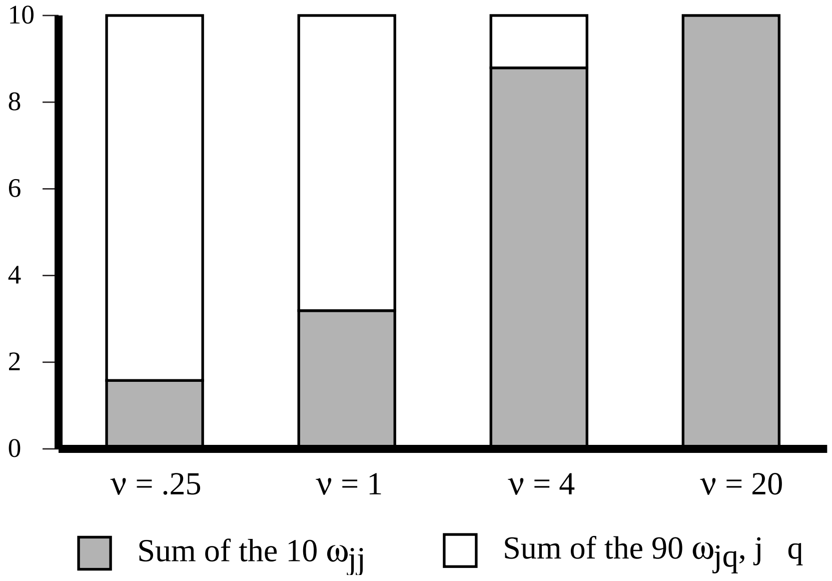
The substitution weights are the equivalence weights defined on page 158 for contagion analysis (the substitution weights ω, not the network correlation weights w):

$$\omega_{jq} = \frac{(\mathrm{dmaxq} - d_{jq})^{\nu}}{\sum_k (\mathrm{dmaxq} - d_{kq})^{\nu}},$$

where d_{jq} is the equivalence distance between actors j and q. Any distance data can be input to define equivalence, or any of the equivalence options can be selected with a POSITIONS command (see page 40). If no POSITIONS command is included in the command file, the default is to define substitutions by structural equivalence (the default POSITIONS command). Structural equivalence is defined by the similarity of j's and q's observed relations, demand for each is defined by those relations, and allowing substitutions between actors homogenizes demand for substitutable actors. So, substitutability defined by structural equivalence operates like a smoothing function on observed demand; homogenizing demand for individuals who are already similar in being sought out to the same extent by the same individuals.*

The exponent ν transforms observed into perceived equivalence. The exponent is read from columns 13-17 of the POWER command in the same way that it is read from the CONTAGION command (see pages 160-161).

*The POWER command, unlike the CONTAGION command, doesn't have options for substitution between cohesive actors; the substantive evidence of cohesion contagion is weak relative to equivalence contagion (Burt, 1992b). If you want to define substitutions between cohesive actors, however, define a distance matrix where distance measures the lack of cohesion and read it in through option 7 in column 12 of the POSITIONS command (as in the first analysis in the EQUI example command file).



The exponent ν operates here as in the contagion models. It is a positive real number that defines a domain of substitution equilibria ranging from interdependence to independence. Toward the extreme of zero, everyone can be viewed as a substitute for anyone else (0 < ν < 1; dotted line in the graph on page 159). A value of ν equal to 1.0 eliminates the exponent; substitutions are defined by observed levels of equivalence between actors (solid line in the graph on page 159). These values of ν are usually too broad a definition of substitution. Greater differentiation exists in who can substitute for whom. At the other extreme, ν equal to a large number can eliminate substitutions. As ν gets large, sources of interaction become increasingly nonsubstitutable (the ω_{jj} approach 1, and the ω_{jq} approach 0, $j \neq q$ — the independence model).

The point is illustrated in the above graph. The bars show the sum of self (shaded areas) and other (white areas) substitution weights for a random network of 10 people. The weights sum to 1 for each person, and so sum to

10 across the network. When ν is a fraction or one, most substitution is between people. When ν is 4, there is little substitution between people. When it is 20, the independence model is reached; all ω_{jj} equal 1, allowing no substitution between people. The point is that the substitution weights progress from interdependence to independence with increasing ν.*

Progress varies with network structure. Stronger equivalence between actors makes it more difficult to eliminate substitution between equivalent actors. At the equivalence extreme of identical relations, substitution weights never reach independence. Structurally equivalent actors j and q are as proximate to each other as to themselves ($d_{jj} = d_{qq} = d_{jq} = 0$), so they are perfectly substitutable ($\omega_{jq} = \omega_{qq}$), and no increase in ν can make ω_{jq} go to zero.

There are two options for exploring alternative values of ν with the POWER command. The first option is to search for a proper value (as in a contagion analysis). If columns 13-17 and columns 21-22 of the POWER command are blank or zero, the program begins with $\nu = 1$. The program searches across successive integer values of ν for the strong correlation between observed and equilibrium relations. The idea is to find the level of substitution most likely responsible for the observed network structure. The default is to stop when the correlation changes less than .05 from ν to $\nu+1$. As in a contagion analysis, you can change the default by entering a negative fraction in place of ν on the POWER command (see page 161).

The second option is to explore a range of ν values regardless of the correlation between observed and equilibrium relations. An integer K is read

*Results for the graph were obtained with MONTE and CONTAGION commands. The 10 people were allowed 1 sociometric choice (with a seed number of 839) and relations were derived from direct and two-step indirect path distances. I then asked for a contagion analysis of random response data with ego results written to the output data file. The ω_{jj} are included in the written data (see pages 166-167).

from columns 21-22 of the POWER command, and equilibrium relations are computed for substitution weights defined by K different values of ν ranging from 1 to K. A summary table is printed describing for each value of ν the extent to which the substitution weights approximate the independence model and the correlation between observed and equilibrium relations.

Output

The output is the same as for equilibrium under independence, with two exceptions: (a) market interests x_{ij} are printed with equilibrium relations (1 in column 18 of POWER command), and (b) a history of stability correlations is printed if alternative values of ν have been tried. The EQUI example command file requests four analyses. The first is the following three-person network of z_{ij}:

$$\left\{z_{ij}\right\} = \begin{bmatrix} .2 & .2 & .6 \\ .3 & .5 & .2 \\ .1 & .1 & .8 \end{bmatrix},$$

with the substitution weights ω_{jq} on page 200. The observed relations and substitution weights combine to define market interests (the x_{ij}):

$$\left\{x_{ij}\right\} = \begin{bmatrix} .20 & .48 & .32 \\ .30 & .55 & .15 \\ .10 & .49 & .41 \end{bmatrix}.$$

Power in the observed network favors the third actor (p_i = .24, .29, and 1.00). This is apparent from the large entries in the third column of z_{ij}. But the second actor can substitute for the third, and not vice versa, so much of the demand for the third actor gets redistributed in the market as demand for the second actor. This is apparent from the large entries in the second column of x_{ij}. Power derived from the x_{ij} favors the second actor (p_i = .44, 1.00, and .49).

```
This is the history of stability correlations across increasing values of
nu. Iter is the number of iterations required to converge on eigenvector
elements. Percent self is the percentage of substitution weight in self-
relations (100% is the independence model).

      NU    ITER    % Self    Stability Correlation
     ----  ------  --------  -------------------------------
        1     5      38.9%      .843
        2     7      59.0%      .954
        3     8      71.3%      .968
        4     8      78.0%      .970
   ----------------------------------------------------------
```

The other three analyses in the EQUI example command file are the three systems analyzed in the POWER example command file. Substitution improves the fit between observed and equilibrium relations among the elite social scientists (third analysis in EQUI). Relative to the .470 stability correlation under independence (page 198), allowing substitutions between structurally equivalent scientists generates a stability correlation of .555 for a ν of 1, .633 for 2 and .650 for a ν of 3.

In contrast, independence is a more likely equilibrium model for the illustrative six-person network on page 184 (second analysis in EQUI). The summary table above shows how similarity between observed and equilibrium relations increases as substitution approaches the independence model (ν is spelled out as nu in the printout). For ν equal 1 to 4, the correlation between observed and equilibrium relations increases from .843 to .970 and substitution weight is increasingly concentrated in self-weights (38.9% up to 78.0%). As substitution approaches independence (100% self-weight), the correlation approaches the 1.0 correlation between observed relations and independence equilibrium relations (page 198). The 78% substitution weight in self-relations is short of the 100% that defines the independence model. The percentage never gets higher than 85.7%. Actors E and F have identical relations, and so are perfectly substitutable. To see the history of correlation across increasing ν, re-run the second analysis in the EQUI example command file with a 20 in columns 21-22 of the POWER command. Independence equilibrium in this network is only reached by explicitly excluding substitution between actors (by putting a 1 in column 12 of the POWER command so that all $\omega_{jj} = 1$ and all $\omega_{jq} = 0$, $j \neq q$).

FORMAT STATEMENTS

STRUCTURE uses standard FORTRAN format statements to read and write data. Such statements are a routine topic in introductory FORTRAN books and were once a routine topic in the manuals for statistics software. With the increasing diversity of format procedures used in different statistical packages, however, it is not as reasonable as it once was to presume that a research literate social scientist knows standard FORTRAN format statements. This Appendix is provided as a convenient reference. Examples are available in the many example command and data files included with the STRUCTURE program disk. You might also find it valuable to see the examples on pages 28-30, 34-35, and 48-49. The following only ensures competence to write basic format statements. The standard FORTRAN format elements and their combinations in format statements are much more diverse than I can describe here (e.g., see Dyck et al., 1984:596ff).

A format statement is a list of format elements — separated from one another by commas with the entire list contained in parentheses — with which you tell a computer how to read variables in a data file. The data file is organized as a large table with each record (respondent, actor, event) beginning on a new row and variables across the columns. A format statement tells the computer which columns belong to which variables. The computer reads the elements sequentially from left to right.

I'll explain format elements in just a minute, but for immediate illustration, consider the example data file to the right containing three variables on four individuals, with a format statement on the first line defining the location of variables in each row of the data file.

```
(5X,I2,A2,F3.0)
ACTOR 1 M 30
ACTOR 2 F 45
ACTOR 3 F 19
ACTOR 4 M 74
```

Each row of the data file contains the data on one actor. There are four columns of information in the file; a text label “ACTOR” (which is of no analytical value but emphasizes the point that each row is a new actor identified by the subsequent number), a sequential identification number (1 to 4), sex (Male or Female), and age (19 to 74). The format statement contains four format elements contained in parentheses. Reading from left to right, the computer first sees “5X” which tells it to skip columns 1 through 5. This skips over the five column “ACTOR” label. The computer next sees “I2” which tells it to read a two digit integer variable from columns 6 and 7. The computer next sees “A2” which tells it to read a two letter alphanumeric variable from columns 8 and 9. Finally, the computer sees “F3.0” which tells it to read a three digit real number variable from columns 10 through 12. The format statement ends with the closing parenthesis. To read data on the next actor, the computer begins on the next row of the file. After it has read the fourth row of data, the computer will contain the three variable data matrix displayed to the right.

```
1 M 30.
2 F 45.
3 F 19.
4 M 74.
```

Now let me offer more specific details on kinds of format elements. Of the many valid format elements, five are sufficient to run STRUCTURE, and you don’t even need to know all of these unless you have complex data files and every kind of network data that can be analyzed by the program.

SKIP ELEMENTS

The two format elements displayed in the shaded box to the right are basic to skipping over information in a file. One skips columns and the other skips lines. There are some examples on page 30.

wX Skip “w” columns. In the example format statement on page 207, format element “5X" tells the computer to skip five columns.

/ Skip to the next line.

INTEGER FORMAT

Iw Define an integer number variable that is “w” columns long. In the example format statement on page 207, element “I2” tells the computer to read actor ID as a two digit integer.

The format element displayed in the box to the right is used in STRUCTURE to read binary and rank order sociometric choice data. Integer data are numbers without fractions. Valid integer data include the numeric digits 0 to 9, blank (read as 0 by STRUCTURE), and a positive or negative sign (+ or -). Any other letters or symbols read as integer data will generate an error. Blanks to the right of the input number will be treated as zeros (in other words, input is “right-justified”). Here are some examples of I5 input with the data displayed as they appear in the input data file (to the left) and displayed as they will be interpreted by the program (to the right):

```
12345     12345
   -1     -1
-1        -1000
 2 34     2034
```

REAL NUMBER FORMAT

Fw.d Define a real number variable that is “w” columns long of which up to “d” are to the right of the decimal point. In the example statement on page 207, the “F3.0” format element tells the computer to read age as a three digit real number variable with zero decimal places.

The format element displayed in the box to the right is used in STRUCTURE to read all response and network data except choice data and event data. Real number data include fractions as well as all integer data. Valid real number data include the numeric digits 0 through 9, blank (read as a 0 by STRUCTURE), a decimal point (.), and a positive or negative sign (+ or -). Any other letters or symbols read as real number data will generate an error. As with integer variables, blanks to the right of the input number will be treated as zeros. The decimal point can be explicit or implicit. Explicitly, if a decimal appears anywhere in the w columns defining a real number, the value of d is ignored. In other words, a decimal point in the input data over-rides the format element used to read the data. Implicitly, if no decimal point appears in the w columns defining a real number variable, a decimal point is inserted between columns d+1 and d from the right end of the variable. Here are some examples of real number input with the data displayed as they appear in the input data file (to the left), the F format element used to read the data (in the middle), and the data displayed as they will be interpreted by the program (to the right):

```
12345          F5.0      12345.0
   -1          F5.0      -1.0
-1.       F5.0      -1.0
 2 34          F5.0      2034
 2 34          F5.1      203.40
 2 34          F5.2      20.34
123.4          F5.2      123.4
```

ALPHANUMERIC FORMAT

The format element displayed in the box to the right is used in STRUCTURE to read actor identification codes and identify the actors involved in events. Alphanumeric data include all integer and real data as well as letters and ASCII standard keyboard symbols. The two lists below give some examples of A8 alphanumeric input with the data displayed as they appear in the input data file

Aw Define an alphanumeric variable that is “w” columns long. In the example statement on page 207, the “A2” format element tells the computer to read sex as a two letter variable.

(to the left) and displayed as they will be interpreted in the program (to the right):

```
12345678       12345678
!@#%&*ed       !@#%&*ed
James               James
   James          James
```

Once read by STRUCTURE or ASSISTANT (or JEDIT, the utility program for joint involvement data), the alphanumeric identification codes assigned to actors are right justified to make them consistent (e.g., input label “James   ” becomes “   James”). This consistency is imposed to improve the reliability of joint involvement data. Without it, James in the above two examples would be treated as two different people; a left-justified James and a right-justified James.

REFERENCES

Alba, R. (1973) "A graph-theoretic definition of a sociometric clique," *Journal of Mathematical Sociology* 3:113-126.

Alba, R. and C. Kadushin (1976) "The intersection of social circles," *Sociological Methods & Research* 1976 5:77-102.

Arabie, P. and S. A. Boorman (1982) "Blockmodels: developments and prospects," In *Classifying Social Data*, edited by H. C. Hudson. San Francisco: Jossey–Bass.

Berelson, B. R., P. F. Lazarsfeld and W. N. McPhee (1954) *Voting*. Chicago: University of Chicago Press.

Berkman, L. F. and S. L. Syme (1979) "Social networks, host resistance, and mortality: a nine-year follow-up study of Alameda County residents," *American Journal of Epidemiology* 109:186-204.

Bernard, H. R., P. D. Killworth and L. D. Sailer (1980) "Informant accuracy in social network data IV: a comparison of clique–level structure in behavioral and cognitive network data," *Social Networks* 2:191–218.

Blau, P. M. (1977) *Inequality and Heterogeneity*. New York: Free Press.

Bonacich, P. (1972) "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology* 2:113-120.

______ (1987) "Power and centrality: a family of measures," *American Journal of Sociology* 92:1170-1182.

Breiger, R. L. (1974) "The duality of persons and groups," *Social Forces* 53: 181-190.

______ (1976) "Career attributes and network structure: a blockmodel study of a biomedical research speciality," *American Sociological Review* 41:117–135.

Breiger, R. L., S. A. Boorman and P. Arabie (1975) "An algorithm for clustering relational data with application to social network analysis and comparison with multidimensional scaling," *Journal of Mathematical Psychology* 12:328–383.

Burt, R. S. (1976) "Positions in networks," *Social Forces* 55:93–122.

______ (1977) "Positions in multiple network systems, part one," *Social Forces* 56:106–131.

______ (1979) "Relational equilibrium in a social topology," *Journal of Mathematical Sociology* 6:211-252.

______ (1980) "Autonomy in a social topology," *American Journal of Sociology* 85:892-925.

______ (1982) *Toward a Structural Theory of Action.* New York: Academic Press.

______ (1983a) "Cohesion versus structural equivalence as a basis for subgroups," In *Applied Network Analysis,* edited by R.S. Burt and M.J. Minor. Beverly Hills: Sage.

______ (1983b) *Corporate Profits and Cooptation*. New York: Academic Press.

______ (1983c) "Range," In *Applied Network Analysis,* edited by R.S. Burt and M.J. Minor. Beverly Hills: Sage.

______ (1986) "A cautionary note," *Social Networks* 8:205–211.

______ (1987) "Social contagion and innovation: cohesion versus structural equivalence," *American Journal of Sociology* 92:1287-1335.

______ (1988a) "Some properties of structural equivalence measures derived from sociometric choice data," *Social Networks* 10:1-28.

______ (1988b) "The stability of American markets," *American Journal of Sociology* 94:356-395.

______ (1990a) "Detecting role equivalence," *Social Networks* 12:83-97.

______ (1990b) "Kinds of relations in American discussion networks," In

Structures of Power and Constraint, edited by C. J. Calhoun, M. W. Meyer and W. R. Scott. New York: Cambridge University Press.

______ (1992a) *Structural Holes*. Cambridge: Harvard University Press.

______ (1992b) *Social Contagion*. New York: Columbia University Press.

Burt, R. S. and W. M. Bittner (1981) “A note on inferences regarding network subgroups,” *Social Networks* 3:71–88.

Burt, R. S. and D. S. Carlton (1989) “Another look at the network boundaries of American markets,” *American Journal of Sociology* 95:723-753.

Burt, R. S. and P. Doreian (1982) “Testing a structural model of perception: conformity and deviance with respect to journal norms in elite sociological methodology,” *Quality and Quantity* 16:109-150.

Burt, R. S. and M. J. Minor (1983, eds.) *Applied Network Analysis*. Beverly Hills: Sage Publications.

Burt, R. S. and T. Uchiyama (1989) “The conditional significance of communication,” In *The Small World*, edited by M. Kochen. Norwood, N.J.: Ablex.

Coleman, J. S. (1958) “Relational analysis: the study of social organization with survey methods,” *Human Organization* 16:28-36.

______ (1966) “Foundations for a theory of collective action,” *American Journal of Sociology* 71:615-627.

______ (1972) “Systems of social exchange,” *Journal of Mathematical Sociology* 2:145-163.

______ (1986) “Social action systems,” In *Individual Interests and Collective Action*. New York: Cambridge University Press.

Coleman, J. S., E. Katz and H. Menzel (1957) “The diffusion of an innovation among physicians,” *Sociometry* 20:253-270.

______ (1966) *Medical Innovation*. New York: Bobbs-Merrill.

Cook, K. S. and R. M. Emerson (1978) “Power, equity, and commitment in exchange networks,” *American Sociological Review* 43:721-739.

Cook, K. S., R. M. Emerson, M. Gillmore and T. Yamagishi (1983) “The distribution of power in exchange networks: theory and experimental results,” *American Journal of Sociology* 89:275-305.

Coser, R. L. (1975) “The complexity of roles as a seedbed of individual autonomy,” In *The Idea of Social Structure*, edited by L.A.Coser. New York: Harcourt, Brace, Jovanovich.

Cronbach, L. J. and G. C. Gleser (1953) “Assessing similarity between profiles,” *Psychological Bulletin* 50:456–473.

Davis, A., B. B. Gardner and M. R. Gardner (1941) *Deep South*. Chicago: University of Chicago Press.

De Graaf, N. D., and H. D. Flap (1988) “With a little help from my friends,” *Social Forces* 67:453-472.

Doreian, P. (1980) “Linear models with spatially distributed data: spatial disturbances or spatial effects?” *Sociological Methods & Research* 9:29-60.

______ (1988) “Equivalence in a social network,” *Journal of Mathematical Sociology* 13:243-282.

Dow, M. M., M. L. Burton and D. R. White (1982) “Network autocorrelation: a simulation study of a foundational problem in regression and survey research,” *Social Networks* 4:169-200.

Duncan, O. D., A. O. Haller and A. Portes (1968) [1971] “Peer influences on aspirations: a reinterpretation,” In *Causal Models in the Social Sciences*, edited by H. M. Blalock. Chicago: Aldine.

Dyck, V. A., J. D. Lawson and J. A. Smith (1984) *FORTRAN/77: An Introduction to Structured Problem Solving.* Englewood-Cliffs: Prentice-Hall.

Efron, B. (1982) *The Jackknife, the Bootstrap and Other Resampling Plans*. Philadelphia: SIAM.

Everitt, B. 1974. *Cluster Analysis*. New York: Wiley.

Festinger, L., S. Schachter and K. W. Back (1950) *Social Pressures in Informal Groups*. Reissued by Sanford: Stanford University Press.

Finifter, B. M. (1972) "The generation of confidence: evaluating research findings by random subsample replication," In *Sociological Methodology 1972*, edited by H. L. Costner. San Fransisco: Jossey-Bass.

Flament, C. (1963) *Applications of Graph Theory to Group Structure*. Englewood Cliffs: Prentice-Hall.

Freeman, L. C. (1977) "A set of measures of centrality based on betweenness," *Sociometry* 40: 35-41.

Freeman, L. C., D. Roeder, and R. R. Mulholland (1980) "Centrality in social networks: II. experimental results," *Social Networks* 2:119-141.

Friedkin, N. E. (1984) "Structural cohesion and equivalence explanations of social homogeneity," *Sociological Methods & Research* 12:235–261.

Friedkin, N. E. and K. Cook (1990) "Peer group influence," *Sociological Methods and Research*, 19:122-143.

Galaskiewicz, J. and R. S. Burt (1991) "Interorganization contagion in corporate philanthropy," *Administrative Science Quarterly* 36:88-105.

Goode, W.J. (1960) "A theory of role strain," *American Sociological Review* 25:483-496.

Gould, R. V. and R. Fernandez (1989) "Structures of mediation: A formal approach to brokerage in transaction networks," In *Sociological Methodology*. Vol.19., edited by C. Clogg. Oxford: Basil Blackwell.

Granovetter, M. (1973) "The strength of weak ties," *American Journal of Sociology* 78:1360-1380.

______ (1983) “The strength of weak ties: a network theory revisited,” In *Sociological Theory 1983*, edited by R. Collins. San Francisco: Jossey-Bass.

______ (1985) “Economic action and social structure: the problem of embeddedness,” *American Journal of Sociology* 91:481-510.

Guilarte, M. (1990) *Delegitimation of the Colonial System after World War Two: A Structural Theory of Influence for Peripheral Countries*. Department of Sociology Ph.D. Dissertation, Columbia University.

Hallinan, M. T. (1974) *The Structure of Positive Sentiment*. New York: Elsevier.

Harary, F., R. Norman and D. Cartwright (1965) *Structural Models: An Introduction to the Theory of Directed Graphs*. New York: John Wiley.

Heinz, J. P. and E. O. Laumann (1982) *Chicago Lawyers*. New York: Russell Sage Foundation.

Hinkley, D. V. (1978) “Improving the jackknife with special reference to correlation estimates,” *Biometrika* 65:13-21.

Holland, P. W. and S. Leinhardt (1970) “A method for detecting structure in sociometric data,” *American Journal of Sociology* 70:492-513.

______ (1973) “The structural implications of measurement error in sociometry,” *Journal of Mathematical Sociology* 3:85–111.

Hubbell, C. H. (1965) “An input-output approach to clique detection,” *Sociometry* 28:377-399.

Hummell, H. J. and W. Sodeur (1987) “Strukturbeschreibung von positionen in sozialen beziehungsnetzen,” In *Methoden der Netzwerkanalyze*, edited by F. Pappi. Munich: Oldenbourg.

Johnson, J. C. (1986) “Social networks and innovation adoption: a look at Burt’s use of structural equivalence,” *Social Networks* 8:343-364.

Katz, E. and P. F. Lazarsfeld (1955) *Personal Influenc*e. New York: Free Press.

Katz, L. (1953) “A new status index derived from sociometric analysis,” *Psychometrika* 18:39–43.

Killworth, P. D. and H. R. Bernard (1974) “Catij: a new sociometric and its application to a prison living unit,” *Human Organization* 33:335-350.

Knoke, D. and R. S. Burt (1983) “Prominence,” In *Applied Network Analysis*:, edited by R. S. Burt and M. J. Minor. Beverly Hills: Sage.

Knoke, D. and J. H. Kuklinski (1982) *Network Analysis*. Beverly Hills: Sage.

Laumann, E. O. (1966) *Prestige and Association in an Urban Community*. New York: Bobbs-Merrill.

Laumann, E. O. and D. Knoke (1987) *The Organizational State*. Madison, WI: University of Wisconsin Press.

Laumann, E. O. and F. U. Pappi (1976) *Networks of Collective Action*. New York: Academic Press.

Lazarsfeld, P. F., B. Berelson and H. Gaudet (1944) *The People’s Choice*. New York: Columbia University Press.

Leinhardt, S. (ed., 1977) *Social Networks*. New York: Academic Press.

Lin, N. (1982) “Social resources and instrumental action,” In *Social Structure and Network Analysis*, edited by P. V. Marsden and N. Lin. Beverly Hills: Sage.

Lin, N., W. Ensel and J. Vaughn (1981) “Social resources and strength of ties: structural factors in occupational attainment,” *American Sociological Review* 46:393-405.

Lindzey, G. and D. Byrne (1968) “Measurement of social choice and interpersonal attractiveness,” In *Handbook of Social Psychology*, Vol. 3, edited by G. Lindzey and E. Aronson. Reading, MA: Addison-Wesley.

Linton, R. (1936) *The Study of Man*. New York: D. Appleton-Century.

Lorrain, F. P. and H. C. White (1971) “Structural equivalence of individuals in social networks,” *Journal of Mathematical Sociology* 1:49–80.

Luce, R. D. and A. D. Perry (1949) “A method of matrix analysis of group structure,” *Psychometrika* 14:95-116.

Markovsky, B., D. Willer and T. Patton (1988) “Power relations in exchange networks,” *American Sociological Review* 53:220-236.

Marsden, P.V. (1982) “Brokerage behavior in restricted exchange networks,” In *Social Structure and Network Analysis*, edited by P. V. Marsden and N. Lin. Beverly Hills: Sage.

______ (1984) “Restricted access in networks and models of power,” *American Journal of Sociology* 88:686-717.

______ (1990) “Network data and measurement,” *Annual Review of Sociology* 16:435-463.

Marsden, P. V. and N. Lin (eds., 1983) *Social Structure and Network Analysis*. Beverly Hills: Sage.

Merton, R. K. (1957) “The role-set: problems in sociological theory,” *British Journal of Sociology* 8:106-120.

Michener, H. A., E. D. Cohen and A. B. Sorensen (1977) “Social exchange: predicting transactional outcomes in 4-event, 3-person systems,” *Journal of Personality and Social Psychology* 32:283- 293.

______ (1977) “Social exchange: predicting transactional outcomes in five-event, four- person systems,” *American Sociological Review* 42:522-535.

Miller, G. S. (1956) “The magical number seven, plus or minus two: some limits on our capacity for processing information,” *Psychological Review* 63:81–97.

Miller, R. G. (1974) "The jackknife — a review," *Biometrika* 61:1-15.

Mintz, B. and M. Schwartz (1985) *The Power Structure of American Business*. Chicago: University of Chicago Press.

Mizruchi, M. S. (1982) *The American Corporate Network: 1904-1974*. Beverly Hills: Sage.

Mizruchi, M. S., P. Mariolis, M. Schwartz and B. Mintz (1986) "Techniques for disaggregating centrality scores in social networks," In *Sociological Methodology 1986*, edited by N. Tuma. San Francisco: Jossey-Bass.

Moscovici, S. (1985) "Social influence and conformity," In *Handbook of Social Psychology*, edited by G. Lindsey and E. Aronson. New York: Random House.

Mosteller, F. and J. W. Tukey (1968) "Data analysis, including statistics," In *Handbook of Social Psychology*, Vol. II, edited by G. Lindzey and E. Aronson. Reading, MA: Addison-Wesley.

Nadel, S. F. (1957) *The Theory of Social Structure*. London: Cohen and West.

Nordlie, P. G. (1958) *A Longitudinal Study of Interpersonal Attraction in a Natural Group Setting*. Department of Psychology Ph.D. Dissertation, University of Michigan.

Ord, K. (1975) "Estimation methods for models of spatial interaction," *Journal of the American Statistical Association* 70:120-126.

Pappi, F. U. (ed., 1987) *Methoden der Netzwerkanalyze*. Munich: Oldenbourg.

Pidd, M. (1984) *Computer Simulation in Management Science*. New York: John Wiley.

Sailer, L. D. (1978) "Structural equivalence: meaning and definition, computation and application," *Social Networks* 1:73–90.

Schott, T. and J. P. Morrissey (1989) "Environments of an organization making for isomorphism: cooperators versus competitors," Paper presented at the Annual Meetings of the American Sociological Association, San Francisco.

Schwartz, J. E. (1977) "An examination of CONCOR and related methods for blocking sociometric data," In *Sociological Methodology 1977,* edited by D. R. Heise. San Francisco: Jossey-Bass.

Snyder, D. and E. L. Kick (1979) "Structural position in the world system and economic growth," *American Journal of Sociology* 84:1096–1126.

Taylor, D. G. and J. S. Coleman (1979) "Equilibrating processes in social networks: a model for conceptualization and analysis," In *Perspectives on Social Network Analysis,* edited by P. W. Holland and S. Leinhardt. New York: Academic Press.

Wellman, B. and S. D. Berkowitz (eds., 1988) *Social Structures*. New York: Cambridge.

Williams, R. M. (1975) "Relative deprivation," In *The Idea of Social Structure,* edited by L. A. Coser. New York: Harcourt, Brace, Jovanovich.

Winship, C. (1988) "Thoughts about roles and relations: an old document revisited," *Social Networks* 10:209-231.

Winship, C. and M. Mandel (1983) "Roles and positions: a clique and extension of the blockmodeling approach," in *Sociological Methodology 1983/84,* edited by S. Leinhardt. San Francisco: Jossey–Bass.

White, D. R., M. L. Burton and M. M. Dow (1981) "Sexual division of labor in African agriculture: a network autocorrelation analysis," *American Anthropologist* 83:824-849.

White, D. R. and K. P. Reitz (1983) "Graph and semigroup homomorphisms on networks of relations," *Social Networks* 5:193–234.

White, H. C., S. A. Boorman and R. L. Breiger (1976) "Social structure from multiple networks, I. blockmodels of roles and positions," *American Journal of Sociology* 81:730–780.

Xia, Fang (1991) *Research Design and Evidence of Social Contagion*. Department of Sociology Ph.D. Dissertation, Columbia University.

Yamagishi, T., M. R. Gillmore and K. S. Cook (1988) “Network connections and the distribution of power in exchange networks,” *American Journal of Sociology* 93:833-851.

Ziegler, R. (1987) “Positionen in sozialen raumen, die multivariate analyse multipler netzwerke,” In *Methoden der Netzwerkanalyse*, edited by F. U. Pappi. Munich: Oldenbourg.

INDEX