SNAP (Social Network Analysis Procedures)
for **GAUSS** Version 2.5*

Friedkin, Noah E. 2001. *SNAP: Social Network Analysis Procedures,* Version 2.5. Maple Valley, WA: Aptech Systems, Inc.

Developer

Professor Noah E. Friedkin
Department of Sociology
University of California
Santa Barbara, CA 93106

URL: http://www.soc.ucsb.edu/faculty/friedkin/

Purchasing Information

Aptech Systems, Inc.
23804 S. E. Kent-Kangley Road
Maple Valley, WA 98038

Tel: (425) 432-7855
Fax: (425) 432-7832
URL: http://www.aptech.com/

SNAP: Social Network Analysis Procedures

Abstract. SNAP provides a suite of procedures for social network analysis. Database procedures facilitate the input and output of network data. Analysis procedures perform symmetrization and other basic operations or return information about a network including its density, connectivity category, geodesics, and point centralities. Other procedures permit specialized analyses on structural equivalence, blockmodels, balance models, and social influence systems.

INTRODUCTION

Because statistical software programs do not include procedures for the analysis of networks, network analysts must write their own programs or rely on one or more of the available programs for network analysis — GRADAP, NEOGOPY, SONNET, STRUCTURE, and UCINET. The field of network analysis includes a variety of substantive areas, and the available programs for network analysis more or less strongly reflect the particular theoretical interests of their developers. Even UCINET and GRADAP, the least theoretically specialized of the network analysis programs, do not satisfy the theoretical interests of all investigators. None of the network programs aim to provide an integrated environment for statistical/network analyses; therefore, these programs are often employed in tandem with other statistical software programs such as BMDP, SAS, SPSS, or SYSTAT. For instance, an investigator might employ UCINET and STRUCTURE to obtain certain conventional network measures, FORTRAN to construct other network measures, and SPSS to analyze the file that includes these measures.

The goal of SNAP is to provide an integrated environment in which to conduct general mathematical/statistical investigations and social network analyses. It is annoying to have to employ a variety of software programs in order to conduct research on networks. It is even more annoying to have to employ a different combination of programs for each new research project that is pursued. SNAP is a response to these annoyances. Clearly, it would be useful to bundle a set of social network procedures together with an established system for statistical analysis. Since no set of "canned" network procedures will serve the theoretical interests of all investigators, the statistical system ought to provide for basic programming. Readers who are familiar with the GAUSS System will readily understand why I chose GAUSS to host SNAP.

Overview of GAUSS

GAUSS is a highly acclaimed mathematical and statistical programming language in which the basic unit of analysis is a matrix. Writing programs in GAUSS is efficient because GAUSS uses a high level programming language; nevertheless, GAUSS programs execute rapidly. This unusual combination of easy programming and fast execution occurs because the numerical algorithms in GAUSS were optimized to deal with matrices and matrix operations. Furthermore, with the matrix as the basic unit of analysis, the possible statistical applications of GAUSS encompass most of modern applied mathematics. Noteworthy among these applications are regression and linear statistical models, maximum likelihood and non-linear estimation, time series, and stochastic simulations. State-of-the-art developments in statistical analysis, if not provided by GAUSS, usually are available in the form of supplemental programs to GAUSS**.** Moreover, access to a huge domain of applications is made possible by a foreign language interface that allows the use of compiled FORTRAN, C, or ASSEMBLY programs from within GAUSS.

GAUSS allows an easy interdigitation of programming and analysis. It is simple in GAUSS to construct, debug, and modify a program that draws on GAUSS procedures, one's own or imported procedures, and lines of ad hoc code. It also is simple to format and direct program outputs to ASCII files. Finally, GAUSS contains excellent publication quality graphics.

Overview of SNAP

The procedures in SNAP may be called up at any point in your GAUSS program. For example:

> M=rndu(30,30); @ is a GAUSS procedure that creates a 30x30 matrix of uniform random numbers. @
>
> A=M .ge .50; @ is a GAUSS operation that creates a matrix of ones and zeros. @
>
> C=ccat(A,1); @ is a SNAP procedure that determines the connectivity category of A; A is either strong unilateral, weak, or disconnected. @

If you prefer to conduct most of your statistical analysis in SPSS, SAS, or some other language, then you may use GAUSS to construct the data files for the analysis. For example:

> load M[30,30] = a:\net.asc; @ is a GAUSS procedure that imports a ASCII file containing a square 30 x 30 matrix. @
>
> {TEC,IEC,MEC}=pcennf(M,0); @ is a SNAP procedure that computes three measures of point centrality. @
>
> output file = a:\cen.out reset;

TEC~IEC~MEC;
end; @ are GAUSS operations that create an ASCII file (i.e., a 30 x 3 matrix) of the three measures of point centrality for each of the 30 cases. @.

MACHINE REQUIREMENTS AND INSTALLATION

Although SNAP is likely to run on earlier versions of GAUSS, GAUSS-386 v3.0 or above is recommended. GAUSS-386 requires a 386 or 486 compatible machine with an Intel 387 or compatible math co-processor, a minimum of 4 MB of RAM and a hard disk with sufficient free space. DOS 3.3 or above is required.

To install SNAP, copy the procedures to a subdirectory C:\GAUSS\SNAP25 and append **;$(GAUSSDIR)\SNAP25** to SRC_PATH in your GAUSSI.CFG file:

SRC_PATH = $(GAUSSDIR)\SRC;$(GAUSSDIR)\SNAP25

COMMAND REFERENCE

DATABASE

FTOM	returns a matrix from a dyad file (ASCII or GAUSS).
FTOM2	returns the matrix of a network from a delimited ASCII file in node list (geographic algorithm) format.
LTOM	returns the matrix of a network from a delimited ASCII file or a memory resident matrix containing the first point, the second point, and line value for all nonzero lines in the network.
MTOL	returns the lines and line values of a network.
RESP1	sorts and expands a survey data matrix by inserting rows of missing values in locations corresponding to the nonrespondents.

CREATE

ADDP	adds a point to a network
ALLNET	returns a matrix containing all possible subsets of a number of directed lines, where each row in the matrix contains one such subset
DELP	deletes a point from a network
MINSERT	inserts rows of X, an $n \times k$ matrix, into a matrix that has a larger number of rows and the same number of columns as X
MISSET	sets parts of an $n \times n$ matrix to missing values.
NID3	retrieves a network from the population of nonisomorphic digraphs with three points.
NID4S	retrieves a network from the population of nonisomorphic symmetric digraphs with four points.
NID5S	retrieves a network from the population of nonisomorphic symmetric digraphs with five points.
RANA	returns an adjacency matrix with N points and L randomly placed lines.
RANA2	returns a symmetric adjacency matrix with N points and L randomly placed lines.
RANW	returns a random (Markov) chain.
TREE	returns a tree with N points and random outdegrees with an optional constraint on maximum outdegree.

TRANSFORM

ADJM	returns an adjacency matrix with options for symmetry and the diagonal.
EXPW	returns a matrix of weights from an exponential function of a distance matrix.
NORM	normalizes a matrix.
STAND	standardize a matrix.
STAND2	standardize a matrix by ranging.
SYM	symmetrize a $n \times n$ matrix.

CLASSIFY

ANPAIR	returns the associated number pair for each point in a strongly connected network.
CCAT	returns the connectedness category of a network or the connectedness category of each pair of points in a network.
CHAIN	classifies a chain as ergodic (irreducible aperiodic) or nonergodic.
PCLASS	classifies the points of a network into the categories of isolate, ordinary, transmitter, receiver, and carrier.
REMOVL	returns a classification, c_ic_j, of the lines of a network where c_i is the connectedness category of the network and c_j is the connectedness category of the network with the line removed.
REMOVP	returns a classification, c_ic_j, of the points of a network where c_i is the connectedness category of the network and c_j is the connectedness category of the network with the point removed.

CONNECTIONS

BLOKMOD	returns a blockmodel.
BUNDLES	returns three measures of the point bundles in a digraph.
BUNDLES2	returns four measures of the point bundles in a digraph, excluding those points in a bundle that have no direct lines to or from any other point in the bundle.
DENSITY	returns the density and average degree of a network.
DENSITY2	returns the density of a network adjusted for nonrespondents.
GEOS	returns the length and number of the shortest paths (geodesics) from i to j, for all i and j in a network.
IFLOW	computes the probability that information has flowed from j to i through one or two step communication paths.

MCT_M	computes the mean first passage matrix for a regular Markov chain.
MCT_N	computes the mean number of times a Markov process, starting from state i, passes through state j before reaching state k for the first time.
MCT_Z	computes the fundamental matrix for a regular Markov chain.
NETSTREE	returns all paths from each point in a network.
NSEQ	returns the number of sequences of length X in a network from *i* to *j* for all *i* and *j*.
PAIRS	returns one of two indices of pair dependency
PATHS3	returns the number of nonredundant paths of length three connecting all pairs of points in a network.
REACH	returns the reachability matrix of a network.
REACH2	returns a limited reachability matrix.
RSET	returns the points in a network that *i* can reach via paths of length dmax or less.
SA1	computes an index of structural access.
TRIADS	returns a triad census.
TWOS	returns the frequency of one of five types of two-step sequences that may join pairs of points in a network.

DISTANCE/PROXIMITY

CONCOR	returns a matrix of profile similarities from a correlation of the column vectors of a matrix.
CONCOR2	returns an hierarchical cluster analysis of units based on their profile similarities.
DPOS	returns the Euclidean distance between the positions of *i* and *j* for all *i* and *j* in a network.
DPOS2	returns a measure of profile dissimilarity (structural equivalence).
DPOS3	returns a measure of profile dissimilarity (structural equivalence).
DPOS4	returns a measure of profile dissimilarity (structural equivalence).
HCLUS	returns an hierarchical cluster analysis of a memory resident matrix of distances.
MINK	returns a distance matrix of Minkowski metrics from an k-dimensional space.
PSIM	returns measures of association (profile similarity) among the rows of a binary (0,1) matrix.
QPLAC	returns a one or two dimensional quadratic placement.
VTOD	returns a distance matrix from a vector.

ASSOCIATION

MCORR computes Pearson's correlation for the entries of two matrices.

PVEC returns relative frequency of adjacencies in A for each subpopulation with a unique value given in B, i.e., $P(A|B_i)$, where A and B are $n \times n$ matrices.

QAP computes and tests the significance of the association between two square matrices based on the quadratic assignment paradigm.

QAPREG returns QAP Monte Carlo tests of significance on the partial correlations between a dependent and set of independent variables.

QAPREG2 tests the significance of regression coefficients and r-square by repetitively permuting the $n \times n$ matrix for the dependent variable calculating the regression coefficients and r-square each time.

RTEST returns the significance level for an approximate randomization test.

TAU returns Kendall's rank correlation coefficient and significance test.

TAUP returns Kendall's partial rank correlation coefficient.

CENTRALITY

PCENLF returns one of three measures of point centrality: degree, closeness, or betweeness.

PCENNF returns Friedkin's (1991) three measures of point centrality: total, immediate, and mediative.

PCENPB returns Bonacich's (1972) measure of point centrality.

PCENPB2 returns Bonacich's (1987) measure of point centrality.

SOCIAL INFLUENCE NETWORK THEORY

MRAR returns maximum likelihood estimates of a and b in the mixed regressive-autoregressive model $\mathbf{y} = a\mathbf{W}\mathbf{y} + \mathbf{X}b + \mathbf{u}$.

PCENNF returns Friedkin's (1991) three measures of point centrality: total, immediate, and mediative.

SINA returns the solution for **A** in $\mathbf{Y}^{(\infty)} = \mathbf{A}\mathbf{W}\mathbf{Y}^{(\infty)} + (\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$

SINSEQ returns the result of the recursion $\mathbf{Y}^{(t+1)} = \mathbf{A}\mathbf{W}\mathbf{Y}^{(t)} + (\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$ where for $t = 1, 2, \ldots, p$.

SINSEQ2 returns the result of the recursion where $\mathbf{Y}^{(t+1)} = \mathbf{A}\mathbf{W}^{(t)}\mathbf{Y}^{(t)} + (\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$ for $t = 1, 2, \ldots, p$.

SINV computes $\mathbf{V} = (\mathbf{I} - \mathbf{A}\mathbf{W})^{-1}(\mathbf{I} - \mathbf{A})$

SINVT returns the total effects matrix at time *t*-1 and predicted outcome scores at time t for $\mathbf{Y}^{(t)} = \mathbf{AWY}^{(t-1)} + (\mathbf{I}-\mathbf{A})\mathbf{Y}^{(1)}$

SINY computes $\mathbf{Y} = (\mathbf{I}-\mathbf{AW})^{-1}(\mathbf{I}-\mathbf{A})\mathbf{Y}^{(1)}$

SINYTOY1 computes $\mathbf{Y}^{(1)} = (\mathbf{I}-\mathbf{A})^{-1}(\mathbf{I}-\mathbf{AW})\mathbf{Y}$

SOCIAL EXCHANGE THEORY

EVM returns predictions of the Expected Value Model of Social Power.

PROCEDURES

ADDP

PURPOSE
Adds a point to a network.

FORMAT
Mk=ADDP(M,r,c,k);

INPUTS

M	$n \times n$ matrix
r	1 × (n+1) vector of row values for point-k
c	(n+1) × 1 vector of column values for point-k
k	scalar position of new point in network

OUTPUTS

Mk	(n+1) × (n+1) network

REMARKS
The row diagonal entry takes precedence over the column diagonal entry.

EXAMPLE
```
m={1 1 0 0,
   1 1 0 0,
   0 0 1 1,
   0 0 1 1};
r=zeros(1,5);
c=ones(5,1);
z=addp(m,r,c,3);z;

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

ALSO SEE
DELP

ADJM

PURPOSE
Returns an adjacency matrix with options for symmetry and the diagonal.

FORMAT
A=ADJM(M,flag);

INPUTS

M	$n \times n$ matrix	
flag	scalar	
	1	simple conversion (unsymmetrized) if $m_{ij} \neq 0,$ then $a_{ij} = 1;$ otherwise $a_{ij} = 0$
	2	convert and symmetrize if $m_{ij} \neq 0$ or $m_{ji} \neq 0,$ then $a_{ij} = 1;$ otherwise $a_{ij} = 0$
	10	unsymmetrized with zero diagonal $a_{ii} = 0;$ if $m_{ij} \neq 0,$ then $a_{ij} = 1;$ otherwise $a_{ij} = 0$
	11	unsymmetrized with ones on diagonal $a_{ii} = 1;$ if $m_{ij} \neq 0,$ then $a_{ij} = 1;$ otherwise $a_{ij} = 0$
	20	symmetrized with zero diagonal $a_{ii} = 0;$ if $m_{ij} \neq 0$ or $m_{ji} \neq 0,$ then $a_{ij} = 1;$ otherwise $a_{ij} = 0$
	21	symmetrized with ones on diagonal $a_{ii} = 1;$ if $m_{ij} \neq 0$ or $m_{ji} \neq 0$ then $a_{ij} = 1;$ otherwise $a_{ij} = 0$

OUTPUTS

A	$n \times n$ adjacency matrix

ALSO SEE
SYM

ALLNET

PURPOSE
Returns a matrix containing all possible subsets of a number (r) of directed lines, where each row in the matrix contains one such subset.

FORMAT
M=ALLNET(r);

INPUTS
r is the number of directed lines

OUTPUTS
M is an $2^r \times r$ matrix, where each row indicates a possible subset of the *r* lines

REMARKS
This procedure will report 2^r and request permission to generate M. To retrieve a subnet from M use the ALLNET2 procedure.

EXAMPLE
m=allnet(4); m;

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

ALSO SEE
ALLNET2

ALLNET2

PURPOSE
Retrieves a subnet from the matrix returned by the ALLNET procedure.

FORMAT
S=ALLNET2(M,P,i);

INPUTS

M	$2^r \times r$ matrix from ALLNET
i	scalar, *i*th row in M
P	$r \times 2$ matrix of assignments of point identifiers to the *r* lines, where p_{j1} is the first point of the *j*th line, p_{j2} is the second point of the *j*th line, and $j = 1,2,\ldots,r$.

OUTPUTS

S	$n \times n$ matrix containing the subset of lines indicated in the *i*th row of M and *n* is the maximum value in P

EXAMPLE
```
m=allnet(4); p={1 2, 1 3, 2 3, 3 4};
s=allnet2(m,p,1); s;
s=allnet2(m,p,8); s;
s=allnet2(m,p,10);s;

   0   0   0   0
   0   0   0   0
   0   0   0   0
   0   0   0   0

   0   0   1   0
   0   0   1   0
   0   0   0   1
   0   0   0   0

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

ALSO SEE

ANPAIR

PURPOSE
Returns the associated number pair for each point in a strongly connected network.

FORMAT
R=ANPAIR(M);

INPUTS

M	$n \times n$ matrix

OUTPUTS

R	$n \times 2$ vector, where R[.,1] contains the outnumbers and R[.,2] contains the innumbers

REMARKS
A network is strongly connected if all points are joined by paths in both directions (see CCAT).

The outnumber of a point is the length of the longest path from the point to the other points in the network. The innumber of a point is the length of the longest path from the other points in the network to the point.

The diameter of the network is maxc(R[.,1])=maxc(R[.,2]).

EXAMPLE
m; anpair(m);

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

2  2
3  3
2  2
3  3
```

REFERENCE
Pp. 160-169 in Harary, Frank, Robert Z. Norman, and Dorwin Cartwright (1965) *Structural models: an introduction to the theory of directed graphs*. New York: Wiley.

BLOKMOD

PURPOSE

Returns a blockmodel.

FORMAT

BM=BLOKMOD(M,nM,BLOCKS,flag,type);

INPUTS

M	$k \times n$ (stacked) matrix	
nM	scalar, number of matrices stacked in M	
BLOCKS	$n \times 2$ matrix, point indices (any order) in first column and block indices (0,1, 2, …) in second column; set indice to zero for no block assignment.	
flag	scalar	
	0	exclude m_{ii} for each matrix in M
	1	include all entries
type	scalar, type of blockmodel	
	1	permute each matrix in M according to block assigments
	2	mean values
	3	density of nonzero values
	4	zeroblock; 0 if all ties are absent, 1 otherwise
	5	oneblock; 1 if all ties are present, 0 otherwise

OUTPUTS

BM — for type=1, $\left(\frac{k}{nM}\right)$, $n \times n$ permuted matrices

$v \times v$ matrices, otherwise, where v is the number of blocks

REMARKS

Output is stored in an ASCII file, BLOKMOD.OUT.

ALSO SEE

CONCOR, CONCOR2

BUNDLES

PURPOSE
Returns three measures of the point bundles in a digraph.

FORMAT
{od,id,dp}=BUNDLES(M);

INPUTS

M	$n \times n$ matrix, where $m_{ii} = 0$

OUTPUTS

od	$n \times 1$ vector, outdegree
id	$n \times 1$ vector, indegree
dp	$n \times 1$ vector, number of distinct points

REMARKS
Outdegree (od) is the number of lines from a point. Indegree (id) is the number of lines to a point.

EXAMPLE

```
m;
0  1  0  1  0  1  0
1  0  1  0  1  0  1
0  1  0  1  0  1  0
1  0  1  0  1  0  1
0  1  0  1  0  1  0
1  0  1  0  1  0  0
0  0  0  0  0  0  0

{od,id,dp}=bundles(m);
od~id~dp;
3  3  3
4  3  4
3  3  3
4  3  4
3  3  3
3  3  3
0  2  2
```

ALSO SEE
BUNDLES2, PCLASS

BUNDLES2

PURPOSE
Returns four measures of the point bundles in a digraph, excluding those points in a bundle that have no direct lines to or from any other point in the bundle.

FORMAT
{od,id,dp,den}=BUNDLES(M);

INPUTS

M	$n \times n$ matrix, where $m_{ii} = 0$

OUTPUTS

od	$n \times 1$ vector, outdegree
id	$n \times 1$ vector, indegree
dp	$n \times 1$ vector, number of distinct points
den	scalar, density of lines among the points in the adjusted bundle

REMARKS
Outdegree (od) is the number of lines from a point. Indegree (id) is the number of lines to a point. The bundle of a point consists of all the lines to and from the point; however, here the bundle of a point is restricted to those points that either send or receive at least one line from another point in the bundle.

ALSO SEE
BUNDLES

CCAT

PURPOSE
Returns the connectedness category of a network or the connectedness category of each pair of points in a network.

FORMAT
C=CCAT(M,flag);

INPUTS

M	$n \times n$ matrix	
flag	scalar	
	1	for connectedness of network
	2	for connectedness of pairs

OUTPUTS

C scalar or $n \times n$ matrix

scalar, connectedness of networks (flag=1)
0 M is disconnected
1 M is strictly weak
2 M is strictly unilateral
3 M is strong

$n \times n$ matrix, connectedness of pairs (flag=2)
0 i and j are 0-connected
1 i and j are 1-connected
2 i and j are 2-connected
3 i and j are 3-connected

REMARKS
Two points are disconnected (0-connected) if they are not joined by a semipath, weakly connected (1-connected) if they are joined by a semipath but not a path, unilaterally connected (2-connected) if they are joined by a path in one direction but not in the other, and strongly connected (3-connected) if they are joined by paths in both directions. The connectedness category of a network is the minimum connectedness category among the pairs of points.

EXAMPLE

m;
0 0 0 1
1 0 0 0
1 0 0 1
1 0 0 0

ccat(m,1);
1

ccat(m,2);
3 2 2 3
2 3 1 2
2 1 3 2
3 2 2 3

ALSO SEE

REMOVL REMOVP

REFERENCE

Pp. 50-84 in Harary, Frank, Robert Z. Norman, and Dorwin Cartwright (1965) *Structural models: an introduction to the theory of directed graphs*. New York: Wiley.

CHAIN

PURPOSE
Classifies a chain as ergodic (irreducible aperiodic) or nonergodic.

FORMAT
c=CHAIN(M);

INPUTS

M	$n \times n$ row stochastic matrix

OUTPUTS

c	scalar 0 M is nonergodic 1 M is ergodic (regular)

EXAMPLE
```
z1;
 0  1  0  0
 0  0  1  0
 0  0  0  1
 1  0  0  0
chain(z1);
 0
z2;
 0   0  1  0
 0   0  1  0
.5   0 .5  0
 1   1  0  0
chain(z2);
 1
```

REFERENCE
Pp. 197-204 in Isaacson, Dean L. and Richard W. Madsen (1976) *Markov Chains: Theory and Applications*. New York: John Wiley.

CONCOR

PURPOSE
Returns a matrix of profile similarities from a correlation of the column vectors of a matrix.

FORMAT
{CM,S,b1,b2,v}=CONCOR(M,nM,b,flag,iter);

INPUTS

M	$k\times n$ (stacked) matrix	
nM	scalar, number of matrices stacked in M	
b	$n\times 1$ vector of column indices in M	
flag	scalar	
	0	exclude m_{ii} for each matrix in M
	1	include all entries
iter	scalar, number of iterations	

OUTPUTS

CM	$n \times n$ matrix of correlations among the columns of M
S	$n \times n$ matrix of the iterated correlations
b1	vector of Block1 indices
b2	vector of Block2 indices
v	vector of indices with zero column variance

REMARKS
M may contain missing values. The analysis is performed on that subset of columns in M indicated by the indices in b. Set b=0 for an analysis of all the columns. Columns in M with zero variance are excluded automatically from the analysis.

Pearson product-moment correlations of the columns of M are computed and returned in CM. A correlation matrix of this correlation matrix is returned for iter=2 (and so on for higher values of iter).

REFERENCES
Brieger, Ronald L., Scott A. Boorman and Phipps Arabie (1975) "An algorithm for clustering relational data with application to social network analysis and comparison with multidimensional scaling." *Journal of Mathematical Psychology* 12: 328-83.

ALSO SEE
CONCOR2, BLOKMOD

CONCOR2

PURPOSE
Returns an hierarchical cluster analysis of units based on their profile similarities.

FORMAT
call CONCOR2(M,nM,flag,iter);

INPUTS

M	$k \times n$ matrix, items by units	
nM	scalar, number of matrices stacked in M	
flag	scalar	
	0	exclude m_{ii} for each matrix in M
	1	include all entries
iter	scalar, number of iterations	

OUTPUTS
Screen output describing the hierarchical partitioning of the units; this output is automatically stored in an ASCII file, CONCOR.OUT.

REMARKS
M may contain missing values. Columns in M with zero variance are excluded automatically from the analysis. The units are partioned into two blocks; then each of these two blocks is partioned if possible; and finally, each of the four (second level) blocks is partioned if possible.

If M is an (item by units) hypergraph, then set nM=1 and flag=1.

ALSO SEE
CONCOR, BLOKMOD

DELP

PURPOSE
Deletes a point from a network

FORMAT
Mk=DELP(M,k);

INPUTS

M	$n \times n$ matrix
k	indice of point to delete

OUTPUTS

Mk	(n-1)×(n-1) matrix

EXAMPLE

```
m;
1 1 1 0 0
1 1 1 0 0
1 1 1 1 1
0 0 1 1 1
0 0 1 1 1

z=delp(m,3);z;
1  1  0  0
1  1  0  0
0  0  1  1
0  0  1  1
```

ALSO SEE
ADDP

DENSITY

PURPOSE
Returns the density and average degree of a network.

FORMAT
{den,avdeg}=DENSITY(M);

INPUTS

M	$n \times n$ matrix

OUTPUTS

den	scalar
avdeg	scalar

REMARKS

$$\text{den} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}}{n(n-1)} \qquad \text{avdeg} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}}{n}$$

where $a_{ii} = 0,\ a_{ij=1}$ if $m_{ij} \neq 0,$ and $a_{ij} = 0$ otherwise.

EXAMPLE
m;

0.00	0.33	0.78	0.10	0.02
0.16	0.00	0.67	0.00	0.00
0.28	0.03	0.00	0.00	0.00
0.28	0.00	0.00	0.00	0.42
0.20	0.00	0.00	0.09	0.00

{den,avdeg}=density(m);
den; avdeg;
0.60 2.40

ALSO SEE
BUNDLES

DENSITY2

PURPOSE
Returns the density of a network adjusted for nonrespondents.

FORMAT
den=DENSITY2(M,r,flag);

INPUTS

M	$n \times n$ matrix	
r	k×1 vector of respondent indices	
flag	scalar	
	0	no exclusions
	1	exclude nonrespondent rows and cols
	2	exclude nonrespondent rows

OUTPUTS

den	scalar

REMARKS

$$\text{den} = \frac{\sum_{i=1}^{r} \sum_{j=1}^{c} a_{ij}}{n^*}$$

where $\mathbf{A} = \left[a_{ij}\right]$ is a submatrix of M, $a_{ii} = 0,\ a_{ij} = 1$ if $m_{ij} \neq 0,\ \ a_{ij} = 0$ if $m_{ij} = 0,$ and n^* is the number of nonexcluded entries in $\mathbf{A}$.

EXAMPLE

```
m;
0 2 3 2 1
0 0 0 0 0
4 0 0 6 4
0 0 0 0 0
9 0 0 8 0
let r = 1 3 5;
den=density2(m,r,0);          den=density2(m,r,1);          den=density2(m,r,2);
den;                          den;                          den;
0.450                         0.833                         0.750
```

ALSO SEE
DENSITY

DPOS

PURPOSE
Returns the Euclidean distance between the positions of i and j for all i and j in a network.

FORMAT
D=DPOS(M,flag);

INPUTS

M	$n \times n$ matrix	
flag	scalar	
	0	exclude m_{ii} values
	1	include m_{ii} values

OUTPUTS
D $n \times n$ matrix where

$$d_{ij} = \left[\sum_{k=1}^{n} \left(m_{ik} - m_{jk} \right)^2 \right]^{\frac{1}{2}}$$

EXAMPLE
m;
2.000 7.000 9.000 6.000
4.000 0.000 9.000 3.000
5.000 9.000 2.000 8.000
5.000 9.000 2.000 3.000

dpos(m,0);
0.000 7.874 8.124 8.426
7.874 0.000 12.490 11.446
8.124 12.490 0.000 5.000
8.426 11.446 5.000 0.000

ALSO SEE
DPOS2, DPOS3,DPOS4

REFERENCE
Cronbach, Lee J. and Goldine C. Gleser (1953) "Assessing similarity between profiles." *The Psychological Bulletin* 50: 456-473.

DPOS2

PURPOSE
Returns a measure of profile dissimilarity (structural equivalence).

FORMAT
D=DPOS2(M,flag);

INPUTS

M	$n \times n$ matrix		
flag	scalar		
		0	exclude m_{ii} values
		1	include m_{ii} values

OUTPUTS
D $n \times n$ matrix where

$$d_{ij} = \left[\sum_{k=1}^{n} \left\{ \left(m_{ik} - m_{jk} \right)^2 + \left(m_{ki} - m_{kj} \right)^2 \right\} \right]^{\frac{1}{2}}$$

ALSO SEE
DPOS,DPOS1, DPOS3,DPOS4

REFERENCE
Burt, Ronald S. (1976) "Positions in networks." *Social Forces* 55:93-122.

DPOS3

PURPOSE
Returns a measure of profile dissimilarity (structural equivalence).

FORMAT
D=DPOS3(M,k);

INPUTS

M	L×n stacked matrix, where L is a multiple of n
K	scalar, number of matrices in the stack L=n×K

OUTPUTS

D $n \times n$ matrix where

$$d_{ij} = \sum_{k=1}^{K} \left[\left(m_{ij(k)} - m_{ji(k)} \right)^2 + \sum_{z=1}^{n} \left\{ \left(m_{iz(k)} - m_{jz(k)} \right)^2 + \left(m_{zi(k)} - m_{zj(k)} \right)^2 \right\} \right]^{\frac{1}{2}} \qquad z \neq i, j$$

REMARKS
The main diagonals of each of the stacked matrices are set to zero. The transposes of the matrices in M are added to the stack.

REFERENCE
Burt, Ronald S. (1987) "Social Contagion and Innovation: Cohesion versus Structural Equivalence."" *American Journal of Sociology* 92: 1287-1335.

DPOS4

PURPOSE
Returns a measure of profile dissimilarity (structural equivalence).

FORMAT
D=DPOS4(M);

INPUTS
M $n \times n$ matrix

OUTPUTS
D $n \times n$ matrix where

$$d_{ij} = \left[\left(m_{ij} - m_{ji} \right)^2 + \sum_{k=1}^{n} \left(m_{ik} - m_{jk} \right)^2 \right]^{\frac{1}{2}} \qquad k \neq i, j$$

ALSO SEE
DPOS, DPOS1,DPOS2,DPOS3

EVM

PURPOSE
Returns predictions of the Expected Value Model of Social Power.

FORMAT
run evm2;

REMARKS
EVM predicts outcomes of social exchange networks comprised of two-party transactions in which each transaction provides one party with a fraction of an amount of resources and the other party with the remaining fraction.

REFERENCES
Friedkin, N.E. "An Expected Value Model of Social Exchange Outcomes." *Advances in Group Processes* 10: 163-193.

Friedkin, N.E. "An Expected Value Model of Social Power: Predictions for Selected Exchange Networks." *Social Networks* 14: 213-229.

EXPW

PURPOSE

Returns a matrix of weights from an exponential function of a distance matrix.

FORMAT

W=EXPW(A,D,v);

INPUTS

A	$n \times n$ adjacency matrix
D	$n \times n$ distance matrix where $d_{ij} \geq 0$
v	scalar where $v > 0$

OUTPUTS

W $n \times n$ matrix where

$$w_{ij} = \frac{a_{ij} v^d}{\sum_{j=1}^{n} a_{ij} v^d}$$

for all i and j

REMARKS

$$\sum_{j=1}^{n} w_{ij} = 1 \text{ and } 0 \leq w_{ij} \leq 1$$

If $v = 1$, there is no relationship between w_{ij} and d_{ij}. If $0 < v < 1$, there is a negative relationship between w_{ij} and d_{ij}. If $v > 1$, there is a positive relationship between w_{ij} and d_{ij}.

EXAMPLE

```
d;
  0.00 13.89  4.63  9.26
13.89  0.00 18.52 23.15
  4.63 18.52  0.00  4.63
  9.26 23.15  4.63  0.00

a;
  0.00  1.00  1.00  0.00
  1.00  0.00  1.00  0.00
  1.00  1.00  0.00  1.00
  0.00  0.00  1.00  0.00

wneg=expw(a,d,.01);
wneg;
  0.00  0.00  1.00  0.00
  1.00  0.00  0.00  0.00
  0.50  0.00  0.00  0.50
  0.00  0.00  1.00  0.00

wbas=expw(a,d,1);
wbas;
  0.00  0.50  0.50  0.00
  0.50  0.00  0.50  0.00
  0.33  0.33  0.00  0.33
  0.00  0.00  1.00  0.00

wpos=expw(a,d,1.99);
wpos;
  0.00  1.00  0.00  0.00
  0.04  0.00  0.96  0.00
  0.00  1.00  0.00  0.00
  0.00  0.00  1.00  0.00
```

FTOM

PURPOSE
Returns a matrix from a dyad file (ASCII or GAUSS)

FORMAT
M=FTOM(data,flag,V,n,z);

INPUTS

data	string, name of dataset	
flag	scalar	
	1	data is a delimited ASCII (.asc) file
	2	data is a GAUSS (.dat) file
V	flag=1: 4×1 vector containing the locations (in terms of rank order placement among the items of information for each observation) of the first point identifier, second point identifier, and line value. The last entry of the vector gives the number of items per observation. flag=2: 3×1 vector containing the locations (indices) of the variables for the first point identifier, the second point identifier, and the line value.	
n	size of M	
z	scalar, M is initialized to z	

OUTPUTS

M	$n\times n$ matrix

REMARKS
M is initialized to z (e.g., z = 0); hence, the dataset need only contain information on the lines in a network.

EXAMPLE

Flag=1 Example:

```
rawdata.asc  (5 observations and 5 variables):
1 2   1     1     0
1 3   0     1     1
2 4   0     1     0
3 5   1     1     0
2 3   1     0     0
data="c:\\gauss\\dat\\rawdata.asc";
 let v = 1 2 4 5;
 n=5;
 M= FTOM(data,1,v,n,0);
 M;
 0    1    1    0    0
 0    0    0    1    0
 0    0    0    0    1
 0    0    0    0    0
 0    0    0    0    0
```

Flag=2 Example:

```
rawdata1.dat
id jd rtyp1 rtyp2 rtyp3
1 2  1  1  0
1 3  0  1  1
2 4  0  1  0
3 5  1  1  0
2 3  1  0  0
1 5  0  0  1
data="c:\\gauss\\dat\\rawdat1";
let v = 1 2 5;
n=5;
M= FTOM(data,2,v,n,0);
M;
0    0    1    0    1
0    0    0    0    0
0    0    0    0    0
0    0    0    0    0
0    0    0    0    0
```

ALSO SEE

FTOM2, LTOM, MTOL

FTOM2

PURPOSE
Returns the matrix of a network from a delimited ASCII file in node list (geographic algorithm) format.

FORMAT
M=FTOM2(data);

INPUTS

data	string, name of dataset

OUTPUTS

M	$n \times n$ matrix

REMARKS
The node list (geographic algorithm) format is: number of points that send lines; then for each point that sends lines, the point identifier, the number of lines from the point, and the identifiers of the points receiving these lines. For example,

```
5        @ number of points that send lines @
1 3      @ point 1 sends 3 lines @
2 3 4    @ point 1 sends lines to points 2, 3, and 4@
2 1      @ point 2 sends 1 line @
5        @ point 2 sends the line to point 5@
etc
```

The size of the network is determined by the largest value among the numbers in the file.

ALSO SEE
FTOM, LTOM, MTOL

GEOS

PURPOSE
Returns the length and number of shortest paths (geodesics) from i to j for all i and j in a network.

FORMAT
{D,G}=GEOS(M,maxd);

INPUTS

M	$n \times n$ matrix
maxd	scalar where $1 \le \text{maxd} \le (n-1)$ is the longest path evaluated

OUTPUTS

D	$n \times n$ matrix where d_{ij} is the length of the shortest path from i to j.
G	$n \times n$ matrix where g_{ij} is the length of the shortest path from i to j.

REMARKS
$d_{ij} = 0$ indicates that a path $\le \text{maxd}$ does not exist in the network; $d_{ii} = 0$.
$g_{ij} = 0$ indicates that a path $\le \text{maxd}$ does not exist in the network; $g_{ii} = 1$.

HCLUS

PURPOSE
Returns an hierarchical cluster analysis of a memory resident matrix of distances.

FORMAT
run hclus.prg;

INPUTS

D	$n \times n$ matrix of distances

OUTPUTS
Screen output; this output is automatically stored in an ASCII file, HCLUS.OUT.

REMARKS
The clustering method is complete linkage, which defines the distance between two clusters as the maximum distance between any two members of the clusters. The program is based on an algorithm provided by Brian M. Steele.

EXAMPLE
```
D=dpos3(x,1);
run hclus.prg;
@enter the letter D when prompted @
```

ALSO SEE
CONCOR2

IFLOW

PURPOSE
Computes the probability that information has flowed from j to i through one- or two-step communication paths

FORMAT
F=IFLOW(P);

INPUTS

P $\quad$ $n \times n$ matrix, where $0 \leq p_{ij} \leq 1$ is the probability that a communication link transmits information from j to i; $p_{ij} = 0$ where there is no communication link

OUTPUTS

F $\quad$ $n \times n$ matrix, where f_{ij} is the probability of a flow of information from j to i

REMARKS

$$\mathbf{F} = \left[f_{ij} \right] = 1 - \left(1 - p_{ij}\right) \prod_{k}^{n} \left(1 - p_{ik} p_{kj}\right) \quad i \neq j \neq k$$

EXAMPLE
```
p={ 0    0   0   0,
   .50   0   0   0,
     0  25   0 .25,
   .50 .50   0   0};

f=iflow(p);f;
  1.000  0.000  0.000  0.000
  0.500  1.000  0.000  0.000
  0.234  0.344  1.000  0.250
  0.625  0.500  0.000  1.000
```

REFERENCE
Friedkin, N.E. (1982) "Information flow through strong and weak ties in intraorganizational social networks." *Social Networks* 3: 273-85.

Friedkin, N.E. (1983) "Horizons of observability and limits of informal control in organizations." *Social Forces* 62: 54-77.

LTOM

PURPOSE
Returns the matrix of a network from a delimited ASCII file or a memory resident matrix containing the first point, the second point, and line value for all nonzero lines in the network

FORMAT
M=LTOM(data,flag);

INPUTS

data	$k\times 1$ vector, for flag=0	
	string, name of ASCII file, for flag=1	
flag	scalar	
	0	dataset is memory resident vector
	1	dataset is a delimited ASCII file

OUTPUTS

M	$n \times n$ matrix

REMARKS
The size of M is the largest value among the entries for the first and second points.

EXAMPLE

Flag=1 (ascii file):
rawdata.asc:

```
1 2 1
1 5 1
2 1 1
2 3 1
3 2 1
3 5 1
5 1 1
5 3 1
```

```
data="c:\\gauss\\dat\\rawdata.asc";
ltom(data,1);
0 1 0 0 1
1 0 1 0 0
0 1 0 0 1
0 0 0 0 0
1 0 1 0 0
```

Flag=0 (memory resident vector):

```
let x=
1 2 1  1 3 1  1 4 1
2 1 1  3 1 1  4 1 1;
ltom(x,0);
0 1 1 1
1 0 0 0
1 0 0 0
1 0 0 0
```

ALSO SEE
MTOL

MCORR

PURPOSE
Computes the Pearson product-moment correlation for the entries of two matrices.

FORMAT
r=MCORR(A,B,Amv,Bmv,flag);

INPUTS

A	$n \times k$ matrix	
B	$n \times k$ matrix	
Amv	scalar, elements of A equal to Amv are set to missing	
Bmv	scalar, elements of B equal to Bmv are set to missing	
flag	scalar	
	0	sets the main diagonals of A and B to missing if both are $n \times n$
	1	otherwise

OUTPUTS

r	correlation coefficient

REMARKS
Any missing values (.) that A and B contain will not be affected by the values for Amv and Bmv.

ALSO SEE
QAP

MCT_M

PURPOSE
Computes the mean first passage matrix for a regular Markov chain.

FORMAT
M=MCT_M(P);

INPUTS
P $n \times n$ regular stochastic matrix

OUTPUTS
M $n \times n$ matrix

REMARKS

$$\mathbf{M} = \left(\mathbf{I} - \mathbf{Z} + \mathbf{E}\mathbf{Z}_{dg}\right)\mathbf{D}$$

where **D** is the diagonal matrix with elements $d_{ii} = 1/c_i$, **c** is the fixed probability vector for **P**,

$\mathbf{Z} = \left(\mathbf{I} - \mathbf{P} + \mathbf{P}^{\infty}\right)^{-1}$, $\mathbf{Z}_{dg}$ results from **Z** by setting off-diagonal entries to 0, and **E** is a $n \times n$

matrix with all entries 1 (Kemeny and Snell 1960:79).

ALSO SEE
CHAIN, MCT_N, MCT_Z

REFERENCE
Kemeny, J.G. and J.L. Snell (1960) *Finite markov chains*. Princeton, NJ: D. Van Nostrand.

MCT_N

PURPOSE

Computes the mean number of times a Markov process, starting from state i, passes through state j before reaching state k for the first time.

FORMAT

N=MCT_N(P,k);

INPUTS

P	$n \times n$ regular stochastic matrix
k	scalar

OUTPUTS

N	$n \times n$ matrix

REMARKS

$$\mathbf{N} = \left(\mathbf{I} - \mathbf{Q}\right)^{-1}$$

where **Q** is the submatrix of **P** consisting of all states but k (Kemeny and Snell, p. 112-3).

ALSO SEE

CHAIN, MCT_M, MCT_Z

REFERENCE

Kemeny, J.G. and J.L. Snell (1960) *Finite markov chains*. Princeton, NJ: D. Van Nostrand.

MCT_Z

PURPOSE
Computes the fundamental matrix for a regular Markov chain

FORMAT
Z=MCT_Z(P);

INPUTS

P $n \times n$ regular stochastic matrix

OUTPUTS

Z $n \times n$ matrix

REMARKS

$$\mathbf{Z} = \left(\mathbf{I} - \mathbf{P} + \mathbf{P}^{\infty}\right)^{-1}$$

ALSO SEE
CHAIN, MCT_M, MCT_N

REFERENCE
Kemeny, J.G. and J.L. Snell (1960) *Finite markov chains*. Princeton, NJ: D. Van Nostrand.

MINK

PURPOSE
Returns a distance matrix of Minkowski metrics from an k-dimensional space

FORMAT
D=MINK(M,r);

INPUTS

M	n×k matrix
r	scalar

OUTPUTS

D	$n \times n$ distance matrix

REMARKS
M contains k coordinates for each of n points:

$$\mathbf{D} = \left[d_{ij}\right] = \left(\sum_{a=1}^{k} \left|m_{ia} - m_{ja}\right|^{r}\right)^{\frac{1}{r}}$$

For Manhattan or city-block metrics set $r = 1$; for Euclidean distances set $r = 2$.

EXAMPLE
v= {1 5, 2 6, 3 7, 4 8};
d=MINK(v,1);d;
d=MINK(v,2);d;

0.000 2.000 4.000 6.000
2.000 0.000 2.000 4.000
4.000 2.000 0.000 2.000
6.000 4.000 2.000 0.000

0.000 1.414 2.828 4.243
1.414 0.000 1.414 2.828
2.828 1.414 0.000 1.414
4.243 2.828 1.414 0.000

ALSO SEE
VTOD

MINSERT

PURPOSE
Inserts rows of X, an $n \times k$ matrix, into an initilized matrix that has a larger number of rows than X and the same number of columns as X

FORMAT
R=MINSERT(X,flag,locXid,nrows);

INPUTS

X	$n \times k$ matrix
flag	scalar; initializes R to flag value; to initialize R to missing values set flag to -1
locXid	scalar, column position of the indices in X on the basis of which rows will be moved into R
nrows	scalar, number of rows in R

OUTPUTS

R	$(nrows) \times k$ matrix

EXAMPLE
```
x;
1  3.4
3  6.2
4  9.7
r=minsert(x,-1,1,5);
r;
   1.0   3.4
     .     .
   3.0   6.2
   4.0   9.7
     .     .
```

MISSET

PURPOSE

Sets parts of an $n \times n$ matrix to missing values.

FORMAT

M2=MISSET(M1,r,flag1,flag2);

INPUTS

M1	$n \times n$ matrix	
r	scalar 0, or $k \times 1$ vector of indices	
flag1	scalar	
	0	set main diagonal entries to missing values
	1	set upper triangular (plus main diagonal) to missing values
flag2	scalar	
	0	default (does nothing)
	1	set indice rows to missing values
	2	set indice rows and columns to missing values

OUTPUTS

M2	$n \times n$ matrix

MRAR

PURPOSE
Returns maximum likelihood estimates of a and b in the mixed regressive-autoregressive model $\mathbf{y} = a\mathbf{W}\mathbf{y} + \mathbf{X}b + \mathbf{u}$.

FORMAT
{a,b,V}=MRAR(y,X,W);

INPUTS

y	$n \times 1$ vector
X	$n \times k$ matrix of variables which may include a vector of ones as the first column
W	$n \times n$ row stochastic matrix

OUTPUTS

a	scalar, estimate of a
b	$k \times 1$ vector, estimate of b
V	$(k+2) \times (k+2)$ matrix, asymptotic variance-covariance matrix $\mathbf{V}\left(s^{2}, b, a\right)$

REMARKS
It is assumed that the errors are normally distributed with zero means and equal variances, i.e., $\mathbf{u} \sim N\left(\mathbf{0}, s^{2}\mathbf{I}\right)$

ALSO SEE

REFERENCES
Anselin, Luc (1988) *Spatial econometrics: methods and models*. Dordrecht: Kluwer Academic.

Cliff, Andrew and Keith Ord (1981) *Spatial processes, models and applications*. London: Pion.

MTOL

PURPOSE
Returns the lines and line values of a network.

FORMAT
L=MTOL(M);

INPUTS
M $n \times n$ matrix

OUTPUTS
L k×3 matrix

L[.,1:2] contain respectively the indices of the first and second points of the lines in M

L[.,3] contains the values of the lines

REMARKS
Missing values in M will not be listed.

EXAMPLE
```
m;
0.00  0.10  0.00  0.00  0.00
0.00  0.00  0.35  0.00  0.00
0.45  0.15  0.00  0.76  0.27
0.00  0.00  0.65  0.00  0.00
0.00  0.00  0.83  0.00  0.00

mtol(m);
1  2  0.10
2  3  0.35
3  1  0.45
3  2  0.15
3  4  0.76
3  5  0.27
4  3  0.65
5  3  0.83
```

ALSO SEE

LTOM, FTOM

NETDATA

PURPOSE
Returns the binary file of a network that is used as input to NETSTREE.

FORMAT
netdata [-[s|t]] netfile

INPUTS

-s	symmetrize the output network
-t	transposes the network
netfile	ASCII file with an .NET extension containing:

descriptive title (1st line)
flag1 flag2 (2nd line)
list form of network structure (3rd-nth line)

where flag1 is 0 if the values in M are binary (e.g 1,0) and 1 otherwise, flag2 is 0 if the values in M are integers and 1 otherwise (n.b. no exponential forms)

OUTPUTS

netfile	binary file with a .BIN extension

REMARKS
NETDATA was written by Professor Norman P. Hummon.

The point identifiers in the list form of the network structure are treated as strings by the program.

The binary file, .BIN, is not viewable.

NETDATA also outputs two additional files: a listing of the network (netfile.LST) and a listing of the in-degree and out-degree of the network points (netfile.DEG). You may want to erase these files after runnning NETDATA**.** All files, .NET, .BIN, .LST, .and .DEG share the same file name, e.g., netfile.*

EXAMPLE

Example #1 of a netfile.asc

```
Network 1
0 0
1 3 5
2
3 5 7
etc.
```

Example #2 of a netfile.asc

```
Network 2
1 0
node1 node3 7 node5 2
node2
node3 node5 1 node7 -3
node4 node8 3
etc.
```

Example #3 of a netfile.asc

```
Network 3
1 1
node1 node3 7.7 node5 2.0
node2
node3 node5 -1.999 node7 3.02
node4 node8 3.55
node5
```

Example of running NETDATA

```
dos netdata c:\gauss\dat\netfile.net;
```

ALSO SEE

NETSTREE

NETSTREE

PURPOSE
Returns all paths from each point in a network.

REMARKS
NETSTREE was written by Professor Norman P. Hummon.

The program requires a binary file as input (c:\gauss\dat\netfile.bin in the example). To create this binary file run NETDATA. The output of NETSTREE is directed to an ASCII file (c:\gauss\dat\pathlist.asc in the example). Beware, this file may be large.

EXAMPLE
```
dos  c:\gauss\snaps\netstree.exe
       c:\gauss\dat\netfile.bin
     >c:\gauss\dat\pathlist.asc;
```

contents of pathlist.asc:

```
Search paths for a
a b c e d f
.
(other paths from a)
.
a g e b d f

Search paths for b
b a f d e c
.
(other paths from b)
.
b d e g a f

Search paths for c
c b a f d e g
.
(other paths from c)
.
.
etc.
```

ALSO SEE
NETDATA

NID3

PURPOSE
Retrieves a network from the population of nonisomorphic digraphs with three points.

FORMAT
M=NID3(i);

INPUTS

i scalar $1 \le i \le 16$

OUTPUTS

M $n \times n$ matrix

REMARKS
NID3(i) will return a disconnected network for i={1,2,3}.

EXAMPLE
nid3(5);
nid3(16);

0 0 0
1 0 1
1 1 0

0 0 1
0 0 1
1 1 0

ALSO SEE
ALLNET, NID4S, NID5S

NID4S

PURPOSE
Retrieves a network from the population of nonisomorphic symmetric digraphs with four points.

FORMAT
M=NID4S(i);

INPUTS

i	$1 \le i \le 11$

OUTPUTS

M	$n \times n$ matrix

REMARKS
NID4S(i) will return a disconnected network for i={1-4 & 6}.

EXAMPLE
nid4s(5);

ALSO SEE
ALLNET, NID3, NID5S

NID5S

PURPOSE
Retrieves a network from the population of nonisomorphic symmetric digraphs with five points.

FORMAT
M=NID5S(i);

INPUTS
i $1 \le i \le 34$

OUTPUTS
M $n \times n$ matrix

REMARKS
NID5S(i) will return a disconnected network for i={1-8,12-14,19}.

EXAMPLE
nid5s(5);
nid5s(10);

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

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

ALSO SEE
ALLNET, NID3, NID4S

NORM

PURPOSE
Normalizes a matrix.

FORMAT
W=NORM(M);

INPUTS
M $n \times n$ matrix

OUTPUTS
W $n \times n$ row stochastic matrix with entries

$$w_{ij} = \frac{m_{ij}}{\sum_{j=1}^{n} m_{ij}}$$

REMARKS
If $\sum_j m_{ij} = 0$, then $w_{ii} = 1$

EXAMPLE
m;

0.00	4.36	0.00	0.00	3.43
5.17	0.00	2.16	0.00	5.27
0.00	0.05	0.00	7.21	3.58
0.00	0.00	8.14	0.00	0.00
8.80	2.28	5.93	0.00	0.00

norm(m);

0.00	0.56	0.00	0.00	0.44
0.41	0.00	0.17	0.00	0.42
0.00	0.00	0.00	0.66	0.33
0.00	0.00	1.00	0.00	0.00
0.52	0.13	0.35	0.00	0.00

NSEQ

PURPOSE
Returns the number of sequences of length x from i to j for all i and j in a network.

FORMAT
S=NSEQ(M,x);

INPUTS
M $n \times n$ matrix
x length of sequence

OUTPUTS
S $n \times n$ matrix where an entry is the number of sequences in M from i to j of length x.

REMARKS
In a sequence, unlike a path, a particular line may appear more than once.

EXAMPLE

m;	nseq(m,2);	nseq(m,3);
0 1 0 0 1	2 0 1 1 0	0 3 1 1 3
1 0 1 0 0	0 2 0 1 1	3 0 3 1 1
0 1 0 1 0	1 0 2 0 1	1 3 0 3 1
0 0 1 0 1	1 1 0 2 0	1 1 3 0 3
1 0 0 1 0	0 1 1 0 2	3 1 1 3 0

ALSO SEE
GEON, TWOS

REFERENCE
Pp. 39-43 in Harary, Frank, Robert Z. Norman, and Dorwin Cartwright (1965) *Structural models: An introduction to the theory of directed graphs*. New York: Wiley.

PAIRS

PURPOSE
Returns one of two indices of pair dependency

FORMAT
D=PAIRS(M,maxd,flag);

INPUTS

M	$n \times n$ matrix	
flag	scalar	
	1	Freeman
	2	Burt
maxd	maximum length of path analyzed (set $\text{maxd} = n-1$ for all paths)	

OUTPUTS

D	$n \times n$ matrix

REMARKS
Freeman's Index:

$$d_{ij(Freeman)} = \sum_{k=1}^{n} \frac{g_{ik(j)}}{g_{ik}} \quad i \neq j \neq k$$

where g_{ik} is the number of geodesics (shortest paths) joining i to k and $g_{ik(j)}$ is the number of such geodesics that contain j as an intermediary between i and k.

Burt's Index:

$$d_{ij(Burt)} = 1 - \frac{r_g}{r_{n-1}}$$

where r_k is the number of points that i can reach in k steps and g is the length of the geodesic from i to j; $d_{ii} = 1$, and $d_{ij} = 0$ if i cannot reach j.

ALSO SEE
SA1

REFERENCES
Burt, Ronald S. (1988) "Some properties of structural equivalence measures derived from sociometric choice data." *Social Networks* 10: 1-28.

Freeman, Linton C. (1980) "The gatekeeper, pair-dependency and structural centrality." *Quality and Quantity* 14: 585-592.

PATHS3

PURPOSE
Returns the number of nonredundant paths of length three connecting all pairs of points in a network

FORMAT
P3=PATHS3(M);

INPUTS
M $n \times n$ matrix

OUTPUTS
P3 $n \times n$ matrix

REMARKS
M is transformed into an adjacency matrix by converting all nonzero entries to unity. In a nonredundant path no point occurs more than once.

EXAMPLE
```
a;
0 1 1 0 0 0 1
1 0 1 0 0 0 0
1 1 0 1 1 0 0
0 0 1 0 1 1 1
0 0 1 1 0 1 1
0 0 0 1 1 0 1
1 0 0 1 1 1 0

p3=path3(a); p3;
0   0   2   4   4   4   2
0   0   0   3   3   3   3
2   0   0   3   3   5   5
4   3   3   0   2   3   4
4   3   3   2   0   3   4
4   3   5   3   3   0   2
2   3   5   4   4   2   0
```

ALSO SEE
NETSTREE

PCENLF

PURPOSE
Returns one of three measures of point centrality: degree, closeness, or betweeness.

FORMAT
{rc,sc}=PCENLF(M,flag,dmax);

INPUTS

M	$n \times n$ matrix	
flag	scalar	
	1	degree
	2	closeness
	3	betweeness
dmax	scalar, where $1 \leq \text{dmax} \leq (n-1)$ is the longest path evaluated	

OUTPUTS

rc	n$\times$1 vector of unstandardized scores
sc	n$\times$1 vector of standardized scores

REMARKS
(1) degree:

$$rc = \sum_{j=1}^{n} a_{ij}, \quad sc = \frac{rc}{n-1}$$

where $a_{ij} = 1$ if $m_{ij} \neq 0$ and $a_{ij} = 1$ otherwise.

(2) closeness:

$$cr = \frac{1}{\sum_{i=1}^{n} d(i,k)} \quad cs = \frac{n-1}{cr}$$

where $d(i,k)$ is the length of the geodesic joining i and k, and $d(i,k) \geq \text{dmax}$ is set to dmax..
(3) betweeness:

$$cr = \sum_{i<j}^{n} \sum \frac{g_{ij}(k)}{g_{ij}} \quad cs = \frac{2cr}{n^2 - 3n + 2}$$

where $g_{ij}(k)$ is the number of geodesics $(\text{length} \leq \text{dmax})$ joining i and j that contain k, and g_{ij} is the number of geodesics linking i and j $(g_{ij} = 1 \text{ if length} > \text{dmax})$. .

EXAMPLE

```
m;
0 0 0 1
1 0 0 0
1 0 0 1
1 0 0 0

degree:
{rc,sc}=pcenlf(nid5s(11),1,4);
rc~sc;
2.000 0.500
2.000 0.500
2.000 0.500
1.000 0.250
1.000 0.250

closeness:
{rc,sc}=pcenlf(nid5s(11),2,4);
rc~sc;
 7.000 0.571
 6.000 0.667
 7.000 0.571
10.000 0.400
10.000 0.400

betwenness:
{rc,sc}=pcenlf(nid5s(11),3,4);
rc~sc;
3.000 0.500
4.000 0.667
3.000 0.500
0.000 0.000
0.000 0.000
```

ALSO SEE

Freeman, Linton C. (1978) "Centrality in social networks: conceptual clarification." *Social Networks* 1:215-239.

REFERENCE

PCENPB, PCENPB2, PCENNF

PCENNF

PURPOSE
Returns Friedkin's (1991) measures of point centrality.

FORMAT
{TEC,IEC,MEC}=PCENNF(W,pfoc);

INPUTS

W	$n \times n$ regular stochastic matrix
pfoc	scalar $0 \leq \text{pfoc} \leq n$; usually set to zero.

OUTPUTS

TEC	$n \times 1$ vector, total
IEC	$n \times 1$ vector, immediate
MEC	$n \times 1$ vector, mediative

REMARKS
TEC is the total interpersonal effect of an actor on other actors in the influence network. TEC indicates the relative contribution of an actor to any consensus or collective decision that is formed among the actors.

IEC is indicative of the rapidity with which the total effects of an actor tend to emerge from the influence process. Actors with greater immediacy are less dependent on intervening actors for their total effects.

MEC indicates the extent to which an actor transmits the total effects of all other actors. If pfoc is set to the indice of a particular actor, then MEC indicates the extent to which an actor transmits the total effects of actor-pfoc.

EXAMPLE

```
w=norm(adjm(nid5s(10),21)); w;
{TEC,IEC,MEC}=pcennf(w,0); TEC~IEC~MEC;
0.333  0.333  0.000  0.000  0.333
0.250  0.250  0.250  0.250  0.000
0.000  0.500  0.500  0.000  0.000
0.000  0.500  0.000  0.500  0.000
0.500  0.000  0.000  0.000  0.500

0.231  0.133  0.667
0.308  0.250  0.875
0.154  0.069  0.347
0.154  0.069  0.347
0.154  0.056  0.264
```

ALSO SEE

PCENLF, PCENPB, PCENPB2

REFERENCE

Friedkin. N.E. (1991) "Theoretical foundations for centrality measures." *American Journal of Sociology* 96: 1478-1504.

PCENPB

PURPOSE
Returns Bonacich's (1972) measure of point centrality

FORMAT
c=PCENPB(M);

INPUTS

M	$n \times n$ matrix

OUTPUTS

c	$n \times 1$ vector of raw centrality scores

REMARKS

$$l\mathbf{c} = \mathbf{Mc}$$

where **c** is the eigenvector of **M** associated with the largest eigenvalue l.

ALSO SEE
PCENLF, PCENPB2, PCENNF

REFERENCE
Bonacich, Phillip. 1972. "Factoring and weighting approaches to status scores and clique identification." *Journal of Mathematical Sociology* 2:113-120.

PCENPB2

PURPOSE
Returns Bonacich's (1987) measure of point centrality

FORMAT
c=PCENPB2(m,a,b,flag);

INPUTS

m	$n \times n$ matrix	
a	scalar (set to any value, e.g. 1, when normalized centrality scores are being returned)	
b	scalar	
flag	scalar	
	0	for absolute value of reciprocal of the largest eigenvalue of m
	1	for raw centrality scores
	2	for normalized centrality scores

OUTPUTS

c	flag=0, scalar
	flag=1-2, $n\times 1$ vector of centrality scores

REMARKS
Raw centrality scores (flag=1): $\mathbf{c}(a,b) = a(\mathbf{I} - b\mathbf{M})^{-1}\mathbf{Mj}$, where **j** where is an $n\times 1$ vector of ones. Katz's (1953) measure of centrality is $\mathbf{t} = b\mathbf{c}(1,b)$. Normalized centrality scores (flag=2):

$a = \sqrt{\frac{n}{\mathbf{x}'\mathbf{x}}}$, where $\mathbf{x} = (\mathbf{I} - b\mathbf{M})^{-1}\mathbf{Mj}$ so that $\mathbf{c}'\mathbf{c} = n$.

ALSO SEE
PCENLF, PCENPB, PCENNF

REFERENCE
Bonacich, Phillip. 1987. "Power and centrality: a family of measures." *American Journal of Sociology* 92:1170-82.

Katz, Leo. 1953. "A new status index derived from sociometric analysis." *Psychometrika* 18:39-43.

PCLASS

PURPOSE
Classifies the points of a network into the categories of isolate, ordinary, transmitter, receiver, and carrier.

FORMAT
C=PCLASS(M);

INPUTS

M	$n \times n$ matrix

OUTPUTS

C	n×1 vector	
	0	isolate
	1	ordinary
	2	transmitter
	3	receiver
	4	carrier

REMARKS
An isolate is a point with an indegree and outdegree of 0. A transmitter is a point with a positive outdegree and an indegree of 0. A receiver is a point with an outdegree of 0 and a positive indegree. A carrier is a point with an indegree and outdegree of 1. Other points are ordinary.

EXAMPLE
```
m;
0  0  0  0  0
1  0  1  1  0
1  0  0  0  0
0  0  1  0  0
0  0  0  0  0

seqa(1,1,rows(m))~pclass(m);
1  3
2  2
3  1
4  4
5  0
```

ALSO SEE

BUNDLES, REMOVP

PSIM

PURPOSE
Returns measures of association (profile similarity) among the rows of a binary (0,1) matrix.

FORMAT
S=PSIM(M,flag1,flag2);

INPUTS

M	$n \times k$ binary (0,1) matrix	
flag1	scalar	
	0	exclude diagonal for $k=n$
	1	include diagonal
flag2	scalar	
	1	Coefficient of Jaccard (Sneath)
		$\mathbf{S} = [s_{ij}] = \frac{a}{a+b+c}$
		where $s_{ij} = 1$ if $a+b+c=0$
	2	Simple Matching Coefficient
		$\mathbf{S} = [s_{ij}] = \frac{a+d}{a+b+c+d}$
	3	Yule Coefficient
		$\mathbf{S} = [s_{ij}] = \frac{ad-bc}{ad+bc}$
	4	Product Moment Correlation Coefficient
		$\mathbf{S} = [s_{ij}] = \frac{ad-bc}{[(a+b)(a+c)(c+d)(b+d)]^{\frac{1}{2}}}$

OUTPUTS

S	$n \times n$ matrix of associations (profile similarities) among the rows of M

REMARKS

	1	0
1	a	b
0	c	d

where

a is the frequency of $\{m_{ig} = 1, m_{jg} = 1\}$,

b is the frequency of $\{m_{ig} = 1, m_{jg} = 0\}$,

c is the frequency of $\{m_{ig} = 0, m_{jg} = 1\}$, and

d is the frequency of $\{m_{ig} = 0, m_{jg} = 0\}$

for $g = 1, 2, \ldots, k$.

Input M' for a k$\times$k matrix of associations among the columns of M.

EXAMPLE

```
 m;
0  0  0  0  1  1  0  1  0  0
0  1  1  0  1  0  1  0  1  0
1  1  0  0  0  0  1  0  1  0
0  0  0  0  1  1  1  0  0  0
0  1  0  0  0  0  0  1  1  0

 s=psim(m,1,1); s;
1.000 0.143 0.000 0.500 0.200
0.143 1.000 0.500 0.333 0.333
0.000 0.500 1.000 0.167 0.400
0.500 0.333 0.167 1.000 0.000
0.200 0.333 0.400 0.000 1.000

s=psim(m,1,2); s;
1.000 0.400 0.300 0.800 0.600
0.400 1.000 0.700 0.600 0.600
0.300 0.700 1.000 0.500 0.700
0.800 0.600 0.500 1.000 0.400
0.600 0.600 0.700 0.400 1.000

s=psim(m,1,3); s;
1.000 -0.455 -1.000 0.846 0.111
-0.455 1.000 0.714 0.455 0.455
-1.000 0.714 1.000 -0.200 0.667
0.846 0.455 -0.200 1.000 -1.000
0.111 0.455 0.667 -1.000 1.000

s=psim(m,1,4); s;
 1.000 -0.218  -0.535  0.524 0.048
-0.218  1.000   0.408   0.218  0.218
-0.535  0.408   1.000 -0.089 0.356
 0.524  0.218  -0.089   1.000  -0.429
 0.048  0.218    0.356 -0.429   1.000
```

ALSO SEE

DPOS, DPOS2

REFERENCE

P.H.A. Sneath and R.R. Sokal (1973) *Numerical Taxonomy*. San Francisco, CA: W.H. Freeman

PVEC

PURPOSE

Returns relative frequency of adjacencies in A for each subpopulation with a unique value in B, i.e., $P(A|B_i)$, where A and B are $n \times n$ matrices.

FORMAT

C=PVEC(A,B,r,flag1,flag2);

INPUTS

A	$n \times n$ adjacency (0,1) matrix		
B	$n \times n$ value matrix		
r	scalar 0, or $k \times 1$ vector of indices for exclusion		
flag1	scalar		
		0	exclude main diagonal
		1	exclude upper triangular (plus main diagonal)
flag2	scalar		
		0	no exclusion
		1	exclude indice row cells
		2	exclude indice row and column cells

OUTPUTS

C	$k \times 3$ matrix,		
		c[.,1]	packr(unique(vecr(B)))
		c[.,2]	frequencies
		c[.,3]	bases of relative frequencies
		c[.,4]	relative frequencies

QAP

PURPOSE
Computes and tests the significance of the association between two square matrices based on the quadratic assignment paradigm

FORMAT
r=QAP(A,B,perm);

INPUTS

A	$n \times n$ matrix
B	$n \times n$ matrix
perm	number of permutations

OUTPUTS

r	3×1 vector
	r[1,1] raw index of association
	r[2,1] normalized index (Pearson's r)
	r[3,1] right tail probability level

EXAMPLE

```
a;                                  b;
0  1  0  1  1                       0  0  1  1  0
0  0  0  0  0                       1  0  1  0  0
1  1  0  1  1                       0  1  0  0  0
1  1  0  0  0                       0  1  1  0  0
1  1  1  0  0                       1  0  0  0  0

r=qap(a,b,1000); r';
   4  -0.167  0.963
```

REFERENCE

Hubert, L.S. and J.V. Schultz (1976) "Quadratic assignment as a general data analysis strategy." *British Journal of Mathematical and Statistical Psychology* 29: 190-241.

Hubert, L.J. and F.B. Baker (1978) "Evaluating the conformity of sociometric measurements." *Psychometrika* 43: 31-41.

ALSO SEE

QAPREG, QAPREG2

QAPREG

PURPOSE
Returns QAP Monte Carlo tests of significance on the partial correlations between a dependent variable and set of independent variables.

FORMAT
call QAPREG(Y,X,perm);

INPUTS

Y	$n \times n$ matrix (dependent variable)
X	$(n \times n) \times$ k matrix (independent variables)
perm	number of permutations

OUTPUTS

screen output

REMARKS
A column of one's is added to X if none exists. The dyads in X are assumed to be in vec (i.e. column) order. Y and X may contain missing values; diagonals are set to missing in the program.

REFERENCES
Smouse, Peter E., Jeffrey C. Long and Robert R. Sokal. 1986 "Multiple Regression and Correlation Extensions of the Mantel Test of Matrix Correspondence." *Systematic Zoology* 35: 627-632.

Krackhardt, David. 1988. "Predicting with Networks: Nonparametric Multiple Regression Analysis of Dyadic Data." *Social Networks* 10: 359-381.

ALSO SEE
QAP, QAPREG2

QAPREG2

PURPOSE
Tests the significance of regression coefficients and r-square by repetitively permuting the $n \times n$ matrix for the dependent variable calculating the regression coefficients and r-square each time.

FORMAT
call QAPREG2(Y,X,perm);

INPUTS

Y	$n \times n$ matrix (dependent variable)
X	$(n \times n) \times$ k matrix (independent variables)
perm	number of permutations

OUTPUTS
screen output

REMARKS
A column of one's is added to X if none exists. The dyads in X are assumed to be in column order. Y and X may contain missing values; main diagonals are set to missing in the program.

REFERENCES
Smouse, Peter E., Jeffrey C. Long and Robert R. Sokal. 1986 "Multiple Regression and Correlation Extensions of the Mantel Test of Matrix Correspondence." *Systematic Zoology* 35: 627-632.

Krackhardt, David. 1988. "Predicting with Networks: Nonparametric Multiple Regression Analysis of Dyadic Data." *Social Networks* 10: 359-381.

ALSO SEE
QAP, QAPREG

QPLAC

PURPOSE
Returns a 1- or 2-dimensional quadratic placement.

FORMAT
D=QPLAC(C,flag);

INPUTS

C	$n \times n$ symmetric matrix where $c_{ij} \geq 0$ and $c_{ii} = 0$	
flag	scalar	
	11	1-dimensional solution and c_{ij} is a measure of similarity (flow intensity)
	12	1-dimensional solution and c_{ij} is a measure of distance (dissimilarity)
	21	2-dimensional solution and c_{ij} is a measure of similarity (flow intensity)
	22	2-dimensional solution and c_{ij} is a measure of distance (dissimilarity)

OUTPUTS

D	n×1 vector, if flag is 11 or 12
	n×2 vector, if flag is 21 or 22

REMARKS
In the 1-dimensional solutions D contains X coordinates where X'X=1. In the 2-dimensional solutions the columns of D contain, respectively, the X and Y coordinates where X'X=1 and Y'Y=1.

EXAMPLE

```
m;
0  1  1  1  1
1  0  1  0  0
1  1  0  0  0
1  0  0  0  1
1  0  0  1  0

qplac(m,21)';
-0.000 -0.500 -0.500  0.500   0.500
-0.000 -0.628   0.628  0.326 -0.326

qplac(m,11)';
-0.000 -0.500 -0.500  0.500  0.500
```

REFERENCE

Hall, Kenneth M. (1970) "An r-dimensional quadratic placement algorithm." *Management Science* 17:219-229.

RANA

PURPOSE
Returns an adjacency matrix with n points and x randomly placed lines.

FORMAT
A=RANA(n,x);

INPUTS
n	scalar, size of adjacency matrix
x	scalar, number of lines; if x=0 a value for x will be randomly selected in the range $0 \le x \le n(n-1)$

OUTPUTS
A	$n \times n$ adjacency matrix in which $a_{ii} = 0$, $a_{ij} = 1$ if there is a line from i to j, and $a_{ij} = 0$ otherwise.

REMARKS
$$x = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}$$

Lines are sampled without replacement.

EXAMPLE
RANA(4,0);

0 0 1 1
1 0 1 1
0 0 0 0
1 0 1 0

RANA(4,3);

0 0 0 1
0 0 0 1
0 0 0 1
0 0 0 0

ALSO SEE
RANA2, RANW

RANA2

PURPOSE
Returns a symmetric adjacency matrix with n points and x randomly placed lines.

FORMAT
A=RANA2(n,x);

INPUTS

n	scalar, size of adjacency matrix
x	scalar, number of lines; if x=0 a value for x will be randomly selected in the range $0 \le x \le n(n-1)/2$

OUTPUTS

A	$n \times n$ adjacency matrix

REMARKS
Lines are sampled without replacement.

EXAMPLE

RANA2(4,3);
0 0 0 1
0 0 0 1
0 0 0 1
1 1 1 0

ALSO SEE
RANA, RANW

RANW

PURPOSE
Returns a random (markov) chain.

FORMAT
W=RANW(M);

INPUTS

M	$n \times n$ matrix

OUTPUTS

W	$n \times n$ matrix

REMARKS

$$\mathbf{W} = \left[w_{ij} \right] = \frac{a_{ij} r_{ij}}{\sum_{j=1}^{n} a_{ij} r_{ij}}$$

where $a_{ij} = 0$ if $m_{ij} = 0$, $a_{ij} = 1$ if $m_{ij} \neq 0$, and r_{ij} is a positive random number (see RNDU). If $\sum_{j=1}^{n} a_{ij} = 0$, then w_{ii} is set to 1. Hence, $0 \leq w_{ij} \leq 1$ and $\sum_{j=1}^{n} w_{ij} = 1$.

EXAMPLE
m;
0 1 1 0
1 0 0 1
1 0 0 1
0 0 0 0

ranw(m);
0.000 0.885 0.114 0.000
0.731 0.000 0.000 0.269
0.713 0.000 0.000 0.287
0.000 0.000 0.000 1.000

ALSO SEE
RANA

REACH

PURPOSE
Returns the reachability matrix of a network.

FORMAT
R=REACH(M);

INPUTS

M	$n \times n$ matrix

OUTPUTS

R	$n \times n$ matrix where $r_{ii} = 1$, $r_{ij} = 1$, if M contains a path from i to j, and $r_{ij} = 0$ otherwise.

REMARKS
In a network, i reaches j if a path exists from i to j.

EXAMPLE
```
m;
0 1 0 0 1
0 0 1 0 0
0 0 0 1 0
0 1 0 0 1
0 1 0 0 0;

reach(m);
1 1 1 1 1
0 1 1 1 1
0 1 1 1 1
0 1 1 1 1
0 1 1 1 1
```

ALSO SEE
REACH2, RSET

REACH2

PURPOSE
Returns the limited reachability matrix for paths $\leq \text{dmax}$ in length.

FORMAT
R=REACH2(M,dmax);

INPUTS

M	$n \times n$ matrix
dmax	scalar, $1 \leq \text{dmax} \leq (\text{n - 1})$

OUTPUTS

R	$n \times n$ matrix, where $r_{ij} = 1$ if there is a path less than dmax in length from i to j in the network, $r_{ii} = 1$, and $r_{ij} = 0$ otherwise.

EXAMPLE

m;	reach2(m,2);	reach2(m,3);
0 1 0 0 0	1 1 1 0 0	1 1 1 1 0
0 0 1 0 0	0 1 1 1 0	0 1 1 1 1
0 0 0 1 0	0 0 1 1 1	1 0 1 1 1
0 0 0 0 1	1 0 0 1 1	1 1 0 1 1
1 0 0 0 0	1 1 0 0 1	1 1 1 0 1

ALSO SEE
REACH, RSET

REMOVL

PURPOSE
Returns a classification, $c_i c_j$, of the lines of a network where c_i is the connectedness category of the network and c_j is the connectedness category of the network with the line removed.

FORMAT
C=REMOVL(M);

INPUTS

M	$n \times n$ matrix

OUTPUTS

C — k$\times$3 matrix where k is the number of lines in the network:

C[.,1] contains the row indices of the lines, C[.,2] contains the column indices of the lines, and C[.,3] contains two-digit codes where the first digit is the connectedness category of the network and the second digit is the connectedness category of the network with the point removed. The digits range from 0 to 3:

0	M is disconnected
1	M is strictly weak
2	M is strictly unilateral
3	M is strong

REMARKS
A line is a bridge if $c_i c_j = 10$ and M is strictly weak, or if $c_i c_j = 20$ and M is strictly unilateral (see CCAT).

EXAMPLE

m;

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

removl(m);

```
1  4 31
2  1 32
3  1 32
4  2 32
4  3 32
4  5 32
5  4 33
5  6 32
6  4 33
6  5 33
```

REFERENCE

Pp. 203-211 in Harary, Frank, Robert Z. Norman, and Dorwin Cartwright (1965) *Structural models: An introduction to the theory of directed graphs*. New York: Wiley.

REMOVP

PURPOSE

Returns a classification, $c_i c_j$, of the lines of a network where c_i is the connectedness category of the network and c_j is the connectedness category of the network with the point removed.

FORMAT

C=REMOVP(M);

INPUTS

M	$n \times n$ matrix

OUTPUTS

C	n×1 vector containing a two-digit code for each point where the first digit is the connectedness category of the network and the second digit is the connectedness category of the network with the point removed. The digits range from 0 to 3:

0	M is disconnected
1	M is strictly weak
2	M is strictly unilateral
3	M is strong

REMARKS

A point is a cut point if $c_i c_j = \{10,20,30\}$

EXAMPLE

```
m;                      seqa(1,1,5)~removp(m);
0  1  0  1  0           1 31
1  0  1  0  0           2 33
1  0  0  0  0           3 32
0  0  1  0  1           4 30
0  0  0  1  0           5 33
```

REFERENCE

Pp. 225-242 in Harary, Frank, Robert Z. Norman, and Dorwin Cartwright (1965) *Structural models: An introduction to the theory of directed graphs*. New York: Wiley.

RESP1

PURPOSE
Sorts and expands a survey data matrix by inserting rows of missing values in locations corresponding to the nonrespondents.

FORMAT
M2=RESP1(M1,r,n);

INPUTS

M1	$m \times k$ matrix
r	$m \times 1$ vector of respondent identification numbers
n	largest identification number among respondents and nonrespondents

OUTPUTS

M2	$n \times k$ matrix

REMARKS
r[i,1] is the respondent identification number associated with the data M1[i,.]

RSET

PURPOSE
Returns the points in a network that *i* can reach via paths of length dmax or less.

FORMAT
R=RSET(M,dmax)

INPUTS

M	$n \times n$ matrix
dmax	scalar

OUTPUTS

R	$n \times n$ matrix where R[i,.] contains the indices of the points that *i* can reach; r_{ij} =. if *j* is not in the reachable set of *i*.

REMARKS
RSET returns the reachable set of a point. To return the antecedent set of a point replace M with its transpose.

EXAMPLE

```
m;                        rset(m,2);              rset(m,3);
0 1 0 0 0 0               1 2 . . . .             1 2 . . . .
1 0 0 0 0 0               1 2 . . . .             1 2 . . . .
0 0 0 1 0 0               . 2 3 4 . 6             1 2 3 4 5 6
0 1 0 0 0 1               1 2 . 4 5 6             1 2 . 4 5 6
0 0 0 1 0 0               . 2 . 4 5 6             1 2 . 4 5 6
0 0 0 0 1 0               . . . 4 5 6             . 2 . 4 5 6
```

ALSO SEE
REACH, REACH2

REFERENCE
Pp. 85-109 in Harary, Frank, Robert Z. Norman, and Dorwin Cartwright (1965) *Structural models: An introduction to the theory of directed graphs*. New York: Wiley.

RTEST

PURPOSE
Returns the significance level for an approximate randomization test.

FORMAT
p=RTEST(&tstat,&rperm,data,trials);

INPUTS

&tstat	user defined procedure to compute test statistic
&rperm	user defined procedure to permute the data
data	$n \times k$ matrix
trials	scalar, number of permutations

OUTPUTS

p	scalar, probability value

REMARKS
This procedure includes an auxiliary random variable that serves to break ties among the computed test statistics while not altering their rank order.

EXAMPLE

```
data;
1.000   2.619
2.000   2.047
3.000   2.932
4.000   5.593
5.000   5.627

proc tstat(data);
    local x;
    x=corrx(data);
retp(x[1,2]);endp;

proc rperm(data);
    local x,y,ry,v,n,rdata;
    x=data[.,1];
    y=data[.,2];
    n=rows(y);
    @random {1,2,..,n}@
    v=rankindx(rndu(n,1),1);
    @permute y@
    ry=submat(y,v,0);
    rdata=x~ry;
retp(rdata);endp;

r=corrx(data);
prob=rtest(&tstat,&rperm,data,999);

"r= " r[1,2];
"prob= " prob;

r=   0.881
prob=   0.029
```

ALSO SEE

QAP

REFERENCE

Eugene S. Edgington. 1980. *Randomization Tests*. New York: Marcel Dekker.

Eric W. Noreen. 1989. *Computer Intensive Methods for Testing Hypotheses*. New York: John Wiley.

SA1

PURPOSE
Computes an index of structural access.

FORMAT
A=SA1(M,p);

INPUTS

M $n \times n$ matrix, where $m_{ij} = 1$ if $j \rightarrow i$ and $m_{ij} = 0$ otherwise

p scalar, $0 \leq p \leq 1$

OUTPUTS

A $n \times n$ matrix, where a_{ij} is *j*'s degree of access to *i*

REMARKS

$$\mathbf{A} = \left[a_{ij}\right] = 1 - \left(1 - p\right)^{x_1}\left(1 - p^2\right)^{x_2}$$

where x_1 is the number of 1-step paths from point *j* to point *i* in M and x_2 is the number of 2-step paths from point *j* to point *i* in M.

ALSO SEE
IFLOW

SINA

PURPOSE

Returns the solution for **A** in $\mathbf{Y}^{(\infty)} = \mathbf{AWY}^{(\infty)} + (\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$

FORMAT

A=SINA(W,Y,Y1);

INPUTS

W	$n \times n$ matrix
Y	n×1 vector
Y1	n×1 vector

OUTPUTS

A	$n \times n$ diagonal matrix

REMARKS

$$y_i^{(\infty)} = a_{ii} \sum_j w_{ij} y_j^{(\infty)} + (1 - a_{ii}) y_i^{(1)}$$

$$y_i^{(\infty)} - y_i^{(1)} = a_{ii} \left(\sum_j w_{ij} y_j^{(\infty)} - y_i^{(1)} \right)$$

$$\frac{y_i^{(\infty)} - y_i^{(1)}}{\sum_j w_{ij} y_j^{(\infty)} - y_i^{(1)}} = a_{ii}$$

for $\left(\sum_j w_{ij} y_j^{(\infty)} - y_i^{(1)} \right) \neq 0$; otherwise a_{ii} is set to missing.

SINSEQ

PURPOSE

Returns the result of the recursion $\mathbf{Y}^{(t+1)} = \mathbf{AWY}^{(t)} + (\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$ for $t = 1,2,\ldots,p$.

FORMAT

Y=SINSEQ(Y1,A,W,p);

INPUTS

Y1	$n \times k$ matrix
a	$n \times n$ matrix
W	$n \times n$ matrix
p	scalar, number of time periods over which the recursion is to run

OUTPUTS

Y	$n \times k$ matrix $\mathbf{Y}^{(p+1)}$

EXAMPLE

```
w;
  0.000   0.386   0.319   0.028   0.268
  0.417   0.000   0.583   0.000   0.000
  0.463   0.537   0.000   0.000   0.000
  0.681   0.000   0.000   0.000   0.319
  0.506   0.000   0.000   0.494   0.000
A=eye(5) .* .999;
Yout=sinseq(Yin,A,W,1);
Yin~Yout;
   9.000  58.765
  90.000   4.419
   1.000  52.433
  66.000  32.332
  82.000  37.193
Yout=sinseq(Yin,A,W,10); Yin~Yout;
   9.000  39.136
  90.000  39.286
   1.000  39.033
  66.000  39.228
  82.000  39.191
```

ALSO SEE

SINSEQ2, SINVT

SINSEQ2

PURPOSE

Returns the result of the recursion $\mathbf{Y}^{(t+1)} = \mathbf{A}\mathbf{W}^{(t)}\mathbf{Y}^{(t)} + (\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$ for $t = 1, 2, \ldots, p$.

FORMAT

Y=SINSEQ2(Y1,A,W,S);

INPUTS

Y1	$n \times k$ matrix
A	$n \times n$ matrix
W	$n \times n$ matrix
S	$m \times 3$ matrix S[.,1] contains the times at which a particular interpersonal influence in W will occur; more than one influence may occur at a particular time S[.,2:3] contains respectively the row and column indices of an interpersonal influence in W

OUTPUTS

Y	$n \times k$ matrix $\mathbf{Y}^{(p+1)}$ obtained from the recursion, where p is the maximum value in S[.,1]

REMARKS

$\mathbf{W}_t$ is a spanning subnet of **W** in containing all of the points and a subset of the lines of **W**. At each t, $\mathbf{W}_t$, is initialized to **0**, the values in **W** that occur at t are inserted into $\mathbf{W}_t$, and the diagonal of $\mathbf{W}_t$ is increased to maintain a row stochastic matrix.

EXAMPLE

```
W;
0.000   0.000   0.000   0.637   0.363
0.822   0.000   0.178   0.000   0.000
0.000   0.000   0.000   0.000   1.000
0.000   0.839   0.000   0.000   0.161
0.000   0.091   0.000   0.909   0.000

Y1';
64  49  55  92  59

S;
1   2   1
3   1   5
3   2   3
4   4   5
7   3   5
7   1   4
7   5   4
8   5   2
8   4   2

A;
.87  0    0    0    0
 0  .87   0    0    0
 0   0   .87   0    0
 0   0    0   .87   0
 0   0    0    0   .87

Y=sinseq2(Y1,A,W,S);
Y';
75  55  58  65  76
```

ALSO SEE

SINSEQ

SINV

PURPOSE

Computes $\mathbf{V} = (\mathbf{I} - \mathbf{AW})^{-1}(\mathbf{I} - \mathbf{A})$

FORMAT

V=SINV(A,W);

INPUTS

W $n \times n$ matrix

$\left\{ 0 \le w_{ij} \le 1, \sum_{j=1}^{n} w_{ij} = 1 \right\}$

A $n \times n$ matrix

OUTPUTS

V $n \times n$ matrix

REMARKS

$$\lim_{\mathbf{A} \to \mathbf{I}} (\mathbf{I} - \mathbf{AW})^{-1}(\mathbf{I} - \mathbf{A}) = \mathbf{W}^{\infty}$$

EXAMPLE

W;
0.000 0.000 1.000 0.000
0.000 0.000 1.000 0.000
0.648 0.270 0.000 0.081
0.000 0.000 1.000 0.000

sinv(A,W);
0.325 0.135 0.500 0.041
0.324 0.136 0.500 0.041
0.324 0.135 0.500 0.041
0.324 0.135 0.500 0.042

REFERENCE

Friedkin, Noah E. and Eugene C. Johnsen. (1990) "Social influence and opinions." *Journal of Mathematical Sociology* 15: 193-205.

SINVT

PURPOSE
Returns the total effects matrix at time t-1 and predicted outcome scores at time t for
$\mathbf{Y}^{(t)} = \mathbf{AWY}^{(t-1)} + (\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$

FORMAT
{Yt,Vt}=SINVT(Y1,A,W,t);

INPUTS

Y1	$n \times k$ matrix
A	$n \times n$ matrix
W	$n \times n$ matrix
t	scalar, time period

OUTPUTS

Yt	$n \times k$ matrix $\mathbf{Y}^{(t)} = \mathbf{V}^{(t-1)}\mathbf{Y}^{(1)}$
Vt	$n \times n$ matrix

ALSO SEE
SINSEQ2

SINY

PURPOSE

Computes $\mathbf{Y} = (\mathbf{I} - \mathbf{AW})^{-1}(\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$

FORMAT

Y=SINY(A,W,Y1);

INPUTS

A	$n \times n$ matrix
W	$n \times n$ matrix $\left\{0 \le w_{ij} \le 1, \sum_{j=1}^{n} w_{ij} = 1\right\}$
Y1	$n \times k$ matrix $\mathbf{Y}^{(1)}$

OUTPUTS

Y	$n \times k$ matrix

REMARKS

EXAMPLE

W;

0.196	0.140	0.194	0.171	0.124	0.176
0.080	0.100	0.286	0.237	0.011	0.286
0.053	0.298	0.229	0.063	0.251	0.107
0.213	0.190	0.306	0.016	0.198	0.078
0.353	0.086	0.134	0.167	0.253	0.007
0.132	0.044	0.202	0.240	0.290	0.091

A=eye(6) .* .456;
Y=siny(A,W,Y1);

Y1~Y;

90	75
94	76
66	63
10	34
33	45
46	49

ALSO SEE
SINV

SINYTOY1

PURPOSE

Computes $\mathbf{Y}^{(1)} = (\mathbf{I} - \mathbf{A})^{-1}(\mathbf{I} - \mathbf{AW})\mathbf{Y}$

FORMAT

Y1=SINYTOY1(A,W,Y);

INPUTS

A	$n \times n$ matrix
W	$n \times n$ matrix $\left\{0 \le w_{ij} \le 1, \sum_{j=1}^{n} w_{ij} = 1\right\}$
Y	$n \times k$ matrix

OUTPUTS

Y1	$n \times k$ matrix

REMARKS

$\mathbf{Y} = (\mathbf{I} - \mathbf{AW})^{-1}(\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$ implies $\mathbf{Y}^{(1)} = (\mathbf{I} - \mathbf{A})^{-1}(\mathbf{I} - \mathbf{AW})\mathbf{Y}$

EXAMPLE

```
W;
  0.290  0.550  0.034  0.094  0.032
  0.228  0.281  0.063  0.120  0.307
  0.036  0.017  0.262  0.380  0.305
  0.369  0.418  0.120  0.013  0.079
  0.178  0.212  0.149  0.231  0.229
```

```
Y=ones(5,1)*6 + rndu(5,1)/10;
A=eye(5) .* .99;
Y1=SINYTOY1(A,W,Y);
Y1~Y;
  3.877  6.060
  9.292  6.099
  0.948  6.003
  8.746  6.095
  3.988  6.043
```

ALSO SEE

SINV, SINY

STAND

PURPOSE
Standardizes a matrix.

FORMAT
S=STAND(M,flag1,flag2);

INPUTS

M	$n \times n$ matrix	
flag1	scalar	
	1	deviations from mean
	2	z-scores
	3	T-scores
flag2	scalar	
	1	standardize by row using row means and standard deviations
	2	standardize matrix using matrix mean and standard deviation

OUTPUTS

S	$n \times n$ standardized matrix

REMARKS

$$\text{flag1} = 1: \quad s_{ij} = m_{ij} - m$$

$$\text{flag1} = 2: \quad s_{ij} = \frac{m_{ij} - m}{s} \quad s_{ij} = 0 \text{ if } s = 0$$

$$\text{flag1} = 3: \quad s_{ij} = 50 + \frac{10\left(m_{ij} - m\right)}{s} \quad s_{ij} = 50 \text{ if } s = 0$$

where m is the row or matrix mean and s is the row or matrix standard deviation.

ALSO SEE
STAND2

STAND2

PURPOSE
Standarizes a matrix by ranging.

FORMAT
S=STAND2(M,flag);

INPUTS

M	$n\times k$ matrix	
flag	scalar	
	1	standardize by row using row minimum and maximum values
	2	standardize matrix using matrix minimum and maximum values

OUTPUTS

S	$n\times k$ standardized matrix

REMARKS
$s_{ij} = \dfrac{m_{ij} - m_{ij(\min)}}{m_{ij(\max)} - m_{ij(\min)}}$, where $m_{ij(\max)}$ is the maximum row (flag=1) or matrix (flag=2) value, and $m_{ij(\min)}$ is the minimum row (flag=1) or matrix (flag=2) value.

EXAMPLE

```
m;
 10  10   7   5
  0   6   7   3
  8   1   6   1
 10   6  10   6
stand2(m,1);
  1.000   1.000   0.400   0.000
  0.000   0.857   1.000   0.429
  1.000   0.000   0.714   0.000
  1.000   0.000   1.000   0.000
stand2(m,2);
  1.000   1.000   0.700   0.500
  0.000   0.600   0.700   0.300
  0.800   0.100   0.600   0.100
  1.000   0.600   1.000   0.600
```

ALSO SEE

STAND

STRGCOMP

PURPOSE
Returns the strong components of a network.

FORMAT
R=STRGCOMP(M,e,flag);

INPUTS

M	$n \times n$ matrix	
e	scalar threshold value for the lines in M; lines below e are eliminated	
flag	scalar	
	0	no output file
	1	results are stored in ASCII file strgcomp.out

OUTPUTS

R	$n \times 2$ matrix containing the point indices in the first column and the component identifiers in the second column

REMARKS
A strong component is a maximally complete subnet in which all members are mutually reachable. Setting the threshold value of e to the highest entry in M will return the strong components of M.

ALSO SEE
BLOKMOD

SYM

PURPOSE
Symmetrizes matrix a $n \times n$ matrix

FORMAT
S=SYM(M,flag);

INPUTS

M	$n \times n$ matrix		
flag	scalar		
		1	$m_{ij} = (m_{ij} + m_{ji})/2$
		2	$m_{ij} = \max(m_{ij}, m_{ji})$
		3	$m_{ij} = \min(m_{ij}, m_{ji})$

OUTPUTS

S	symmetrical matrix

EXAMPLE
```
m;
  6  2  0
 10  1  4
  9  1  7
s=sym(m,1);s;
  6.0  6.0  4.5
  6.0  1.0  2.5
  4.0  2.5  7.0
s=sym(m,2);s;
  6 10  9
 10   1  4
  9  4  7
s=sym(m,3);s;
  6  2  0
  2  1  1
  0  1  7
```

ALSO SEE
ADJM

TAU

PURPOSE
Returns Kendall's rank correlation coefficient and significance test.

FORMAT
{c,p}=TAU(x,y);

INPUTS

x	n×1 vector
y	n×1 vector

OUTPUTS

c	scalar, tau coefficient
p	scalar, probability value

REFERENCE
Siegel, Sidney (1956) *Nonparametric Statistics for the Behavioral Sciences*. New York: McGraw-Hill.

TAUP

PURPOSE
Returns Kendall's partial rank correlation coefficient.

FORMAT
c=TAUP(x,y,z);

INPUTS

x	$n \times 1$ vector
y	$n \times 1$ vector
z	$n \times 1$ vector

OUTPUTS

c	scalar, partial tau for x and y

REMARKS

$$t_{xy.z} = \frac{t_{xy} - t_{zy} t_{xz}}{\sqrt{\left(1 - t_{zy}^2\right)\left(1 - t_{zx}^2\right)}}$$

REFERENCE
Siegel, Sidney (1956) *Nonparametric Statistics for the Behavioral Sciences*. New York: McGraw-Hill.

TREE

PURPOSE
Returns a tree with n points and random outdegrees with an optional constraint on maximum outdegree.

FORMAT
M=TREE(n,maxod,flag);

INPUTS

n	scalar	
maxod	scalar, maximum outdegree $1 \leq \text{maxod}(n-1)$	
	0	no constraint
flag	scalar	
	0	zeros on main diagonal
	1	ones on main diagonal
	2	$m_{11} = 1,\ m_{ii} = 0$ otherwise

OUTPUTS

M	$n \times n$ matrix

REMARKS
maxod=1 returns a path from point 1 to point n.

EXAMPLE
tree(7,0);

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

tree(7,2);

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

ALSO SEE
RANA

TRIADS

PURPOSE
Returns a triad census.

FORMAT
TRIADS(M);

INPUTS

M	$n \times n$ value matrix in which	
	0	indicates a null tie
	1	indicates a weak tie
	2	indicates a strong tie

OUTPUTS

TRIADS.OUT file containing results.

REMARKS

This program is an adapted and modified version of a program written by Eugene C. Johnsen, Department of Mathematics, University of California, Santa Barbara, CA..

In the input matrix $M = \left[m_{ij}\right]$, only the following combinations of ties are permitted:

mij	mji	
1	1	S-Type Dyad
2	2	M-Type Dyad
2	0	A-Type Dyad
0	0	N-Type Dyad

If S-Type dyads are not present, the results are of the form of the standard triad census described by Holland and Leinhardt (1970, 1975).

The "actual values" reported on the output are the frequencies of the various types of triads; each triad type is described by four numbers {S,M,A,N} giving respectively the number of S-Type, M-Type, A-Type, and N-Type dyads that comprise the triad. In some cases, triads are further differentiated with letters {D,U,C,T}; see Holland and Leinhardt (1970, 1975) for details.

The "expected values" reported on the output are the expected frequencies of the various types of triads. The expectation assumes independence; for instance, in the example described below (see triads.doc file), the expected frequency of the 0210 triad is 309.55=(3*1330*73*99*98)/(210*209*208) and the expected frequency of the 0120C triad is 113.71=(3/2)*(1330*99*73*72)/(210*209*208).

The "standardized differences" reported on the output are computed as the (actual value - expected value)/(expected value) for each triad type.

EXAMPLE

```
m;
0 2 0 2 0 0 0 2 0 0 1 0 0 0 0 2 0 2 0 0 1
2 0 0 2 0 2 2 2 0 0 0 2 0 0 0 2 2 2 0 0 2
2 2 0 0 0 0 0 0 0 2 2 2 0 0 0 0 2 2 0 0 0
2 2 0 0 1 2 0 2 0 2 2 2 0 0 0 2 0 2 0 0 2
2 2 2 1 0 0 0 1 0 2 2 0 0 2 0 2 2 2 2 2 2
2 2 2 2 2 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 0 2 2 0 1 2 2 2 0 2 2 2 2 2 2 2 2 2
2 0 0 2 1 2 1 0 0 0 2 2 0 0 0 2 2 2 0 2 2
2 2 2 0 2 2 0 0 0 0 2 0 2 2 2 2 2 2 2 2 0
0 0 2 0 2 0 0 2 2 0 2 0 2 2 2 2 0 2 2 2 0
1 2 2 0 2 0 2 2 2 0 0 0 2 2 2 2 0 2 2 2 0
2 2 0 2 0 2 0 2 0 0 0 0 0 2 2 0 2 2 0 2 2
2 2 2 0 2 2 0 0 2 2 2 0 0 2 2 2 2 2 2 2 0
2 2 2 0 2 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 1
2 2 2 0 2 2 0 0 2 0 2 0 2 2 0 2 2 2 2 2 2
2 2 0 2 0 2 0 2 0 2 0 0 0 0 0 0 0 2 0 2 0
2 2 2 2 2 0 2 2 2 0 2 2 2 2 2 2 0 2 2 2 2
2 2 2 0 2 0 2 2 2 2 2 2 2 2 2 2 0 0 2 2 2
2 2 2 0 2 2 0 0 2 2 2 0 2 2 2 2 2 2 0 2 0
2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 0 0
1 2 0 0 0 2 2 2 0 0 0 2 0 1 0 0 2 2 0 0 0

call triads(m); @see triads.doc file for the output of this example@
```

REFERENCE

Holland, Paul W. and Samuel Leinhardt. (1970) "A method for detecting structure in sociometric data." *American Journal of Sociology* 70: 492--13.

Holland, Paul W. and Samuel Leinhardt. (1975) " Local structure in social networks." In D.R. Heise (Ed.), *Sociological Methodology*. San Francisco: Jossey-Bass.

TWOS

PURPOSE

Returns the frequency of one of five types of two-step sequences from i to j.

FORMAT

F=TWOS(M,flag);

INPUTS

M	$n \times n$ matrix		
flag	scalar		
		1	disjoint sequences
		2	$i \leftarrow k \rightarrow j$
		3	$i \rightarrow k \leftarrow j$
		4	$i \rightarrow k \rightarrow j$
		5	$i \leftarrow k \leftarrow j$

OUTPUTS

F	$n \times n$ matrix where f_{ij} sequences of a particular type from i to j.

REMARKS

The number of disjoint two-step sequences is the number of distinct "middle" points on the two-step sequences that join i and j.

EXAMPLE

	twos(m,1);	twos(m,2);	twos(m,3);	twos(m,4);	twos(m,5);
m;	disjoints	$i \leftarrow k \rightarrow j$	$i \rightarrow k \leftarrow j$	$i \rightarrow k \rightarrow j$	$i \leftarrow k \leftarrow j$
0 1 1 1	0 2 2 2	0 0 0 0	0 1 1 1	0 1 1 1	0 0 0 0
0 0 0 1	2 0 2 2	0 0 1 1	1 0 0 0	0 0 1 0	1 0 0 1
0 1 0 0	2 2 0 2	0 1 0 1	1 0 0 0	0 0 0 1	1 1 0 0
0 0 1 0	2 2 2 0	0 1 1 0	1 0 0 0	0 1 0 0	1 0 1 0

ALSO SEE

NSEQ, GEON

VTOD

PURPOSE
Returns a distance matrix from a vector

FORMAT
D=VTOD(v,flag);

INPUTS

v	$n\times 1$ vector	
flag	scalar	
	1	$\mathbf{D}=\left[d_{ij}\right]=v_i v_j$
	2	$\mathbf{D}=\left[d_{ij}\right]=\left(v_i-v_j\right)^2$
	3	$\mathbf{D}=\left[d_{ij}\right]=\left\|v_i-v_j\right\|$

OUTPUTS

D	$n\times n$ distance matrix

EXAMPLE
```
v= {1,2,3,4};
d=vtod(v,1);d;
d=vtod(v,2);d;
d=vtod(v,3);d;

1   2   3   4
2   3   6   8
3   6   9  12
4   8  12  16

0   1   4   9
1   0   1   4
4   1   0   1
9   4   1   0

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

ALSO SEE

BETA PROCS

SINMODA

PURPOSE
Returns **A** and **W** in $\mathbf{Y}^{(\infty)} = \mathbf{AWY}^{(\infty)} + (\mathbf{I} - \mathbf{A})\mathbf{Y}^{(1)}$

FORMAT
{ A,W,V,notes } = SINMODA(Y1,Y,C);

INPUTS
Y1	$\mathbf{Y}^{(1)}$	n×k matrix of initial opinions
Y	$\mathbf{Y}^{(\infty)}$	n×k matrix of equilibrium opinions
C	$\mathbf{C} = [c_{ij}]$	$n \times n$ matrix of relative interpersonal weights $\left(0 \le c_{ij} \le 1, c_{ii} = 0, \sum_j c_{ij} = 1\right)$ or scalar 0 for a baseline martix

OUTPUTS
A	$\mathbf{A} = [a_{ii}]$	$n \times n$ matrix $\left(0 \le a_{ii} \le 1, a_{ij} = 0, i \ne j\right)$
W	$\mathbf{W} = \mathbf{AC} + \mathbf{I} - \mathbf{A}$	n×n matrix of relative direct effects
V	$\mathbf{V}$	$n \times n$ matrix of total effects
Y	$\hat{\mathbf{Y}}^{(\infty)}$	n×k matrix of predicted equilibrium opinions
eff		scalar, number of time periods to reach equilibrium
notes		informational codes indicating why the system is outside the scope conditions of the model:

notes[1,1] = 1 there is an initial consensus
Y = $\mathbf{Y}^{(1)}$ and **A=I**

notes[2,1] = 1 **Y** is outside the convex hull of $\mathbf{Y}^{(1)}$
Y = A = W = V =.

notes[3,1] = 1 other
Y = A = W = V =.

REMARKS
Set $\mathbf{C} = 0$ (i.e., scalar zero) for a baseline $\mathbf{C} = \frac{1}{n-1}(\mathbf{1} - \mathbf{I})$