XI CONVEGNO ITALIANO DEGLI UTENTI DI STATA

Milano, 13-14 Novembre 2014

`datanet`: a Stata routine for organizing a dataset for network analysis purposes

Giovanni Cerulli and Antonio Zinilli
CNR, Ceris
National Research Council of Italy
Institute for Economic Research on Firms and Growth
g.cerulli@ceris.cnr.it; a.zinilli@ceris.cnr.it

Abstract. This paper presents and applies a new user-written Stata program, `datanet`, which facilitates the dataset organization for network analysis' purposes. Given a fixed number of units (or nodes) belonging to the same group (there will be a variable denoting group membership), possibly connected one each other or possibly not, this routine creates a new dataset containing all their possible couplings to then be easily exploited using Stata network analysis' commands. So far, to our knowledge, no routine has been developed in Stata which executes this type of procedure.

1. Introduction

This paper presents a user-written Stata routine - `datanet` - motivated by the need of organizing a dataset for network analysis purposes. Given a fixed number of units (or nodes) belonging to a same group, possibly connected one each other or possibly not, this routine creates all their possible couplings. For each series of nodes within a network, there exists a finite number of possible relationships. Each node can act as a source or destination in relation to all other nodes. Each line corresponds to a relationship and data will be organized into the following columns:

- the identifier of the individual who initiated the relationship;
- the identifier of the individual who serves as the target of the relationship;
- the weight of the relationship (optional).

To launch the command, one needs to have at least two variables in the dataset, the first we call here *id*, and the second we call here *x*. The variable *id* indexes units (such as people or organizations) forming the nodes o(r vertices) of the network, while the variable *x* denotes the group to which the single unit belong, coded through a specific group identifier.

2. An illustrative example

In order to understand the nature of the problem, suppose to have a dataset containing a variable *id* identifying units, and a variable *x* representing group membership as follows:

Table 1. *Input dataset*.

	id	x
1.	A	1
2.	B	1
3.	C	2
4.	D	3
5.	E	1
6.	F	4
7.	G	2
8.	H	1

For instance, the variable *id* may contain names of people, organizations, and so forth. In this specific example, *id* contains eight units coded as A, B, C, D, E, F, G, and H, while *x* refers to four different groups indexed by 1, 2, 3, 4. In this form, the dataset is not ready for being read by

network analyses software yet, including network analyses commands for Stata. The correct dataset is a rearranged version of the previous one, taking the following form:

Table 2. *Output dataset*.

	IN	OUT	x
1.	A	B	1
2.	A	E	1
3.	A	H	1
4.	B	E	1
5.	B	H	1
6.	C	G	2
7.	D		3
8.	E	H	1
9.	F		4

where, within the same group, all possible couples have been created. For instance, since both units A and unit B belong to group 1, we need to create a new dataset where matched units are in the same row but different columns, with the group identifier appearing in the same row. Our aim is thus that of producing such form of the starting dataset implementing a Stata routine doing this.

3. Syntax of `datanet`

The syntax of the `datanet` command, whose ADO-file is reported in Appendix A, is as follows:

`datanet` *`unit_identifier group_identifier`*

where:

`unit_identifier`: is the variable identifying the units

`group_identifier`: is the variable identifying the group

As output, `datanet` produces two variables, "*IN*" and "*OUT*", which indicate the name of the unit from which the link originates (*IN*) and that of the unit receiving the link (*OUT*).

4. Application

As example, we descriptively analyze the collaborations among Italian researchers over a specific funding scheme, the Project of National Research (PRIN) in the Chemical area (years 2010-2011).

We have a dataset containing a variable "*researcher*" identifying units, with a total number of involved researchers equal to 179; and a variable, "*code*", representing group membership. In network studies, the three most important centrality measures used in applications are: *degree*, *closeness*, and *betweenness* (Freeman, 1977), some of which measures differ in their applications to undirected and directed ties.

The orginal dataset in a form similar to that in Figure 1, so we have to turn it into a dataset taking the form as in Figure 2. To this end, we use the `datanet` command:

```
. datanet researcher code
```

Subsequently, we can use the Stata user-written commands `netsis` and `netsummarize` provided by Miura (2012) to compute the three centrality measures presented above:

Degree centrality. A researcher with high degree centrality maintains numerous contacts with other researchers. A researcher has high centrality when he is able to influence over others, at least beyond a certain extent. The degree centrality is:

$$C_D(p_k) = \sum_{i=1}^{n} a(p_i + p_k)$$

where n is the number of nodes and $a(p_1+p_2)=1$ if and only if node i and k are connected, and $a(p_1+p_2)=0$ otherwise. To calculate this measure in Stata, we can use the user-written command `netsis`, taking as arguments IN and OUT, i.e. the output provided by `datanet`:

```
. duplicates drop IN  // this eliminates name duplicates
. netsis IN OUT , measure (adjacency) name (A, replace) 0
. netsummarize A/(rows(A)-1), generate(degree) statistic(rowsum)
```

Closeness centrality. Closeness is based on the length of the average shortest path between a vertex and all vertices in the graph. A researcher that is close to many others can quickly interact and communicate with them without going through many intermediaries. It is the inverse of the sum of geodesic distances from researcher i to the g-1 other researchers. The closeness centrality is:

$$C_C(p_k) = \left[\sum_{j=1}^{g} d(p_i, p_j) \right]^{-1}$$

. where $d(p_i, p_j)$ is the geodesic distance (shortest paths) linking p_i and p_j. Using Stata, we have:

```
. netsis IN OUT, measure(distance) name(B,replace)
. netsummarize (rows(B)-1):/rowsum(B), ///
generate(closeness) statistic(rowsum)
```

Betweenness centrality. It considers the number of times a particular node lies "between" the various other nodes in the network. It is the portion of the number of shortest paths that pass through the given node divided by the numbers of shortest path between any pair of nodes (Borgatti, 1995). The closeness centrality is formulated as follows:

$$C_B(p_k) = \sum_{i<j} g_{ij}(p_k) / g_{ij}$$

where g_{ij} is the geodesic distance (shortest paths) linking p_i and p_j and $g_{ij}(p_k)$ is the geodesic distance linking p_i and p_j that contains p_k. Using Stata, we have:

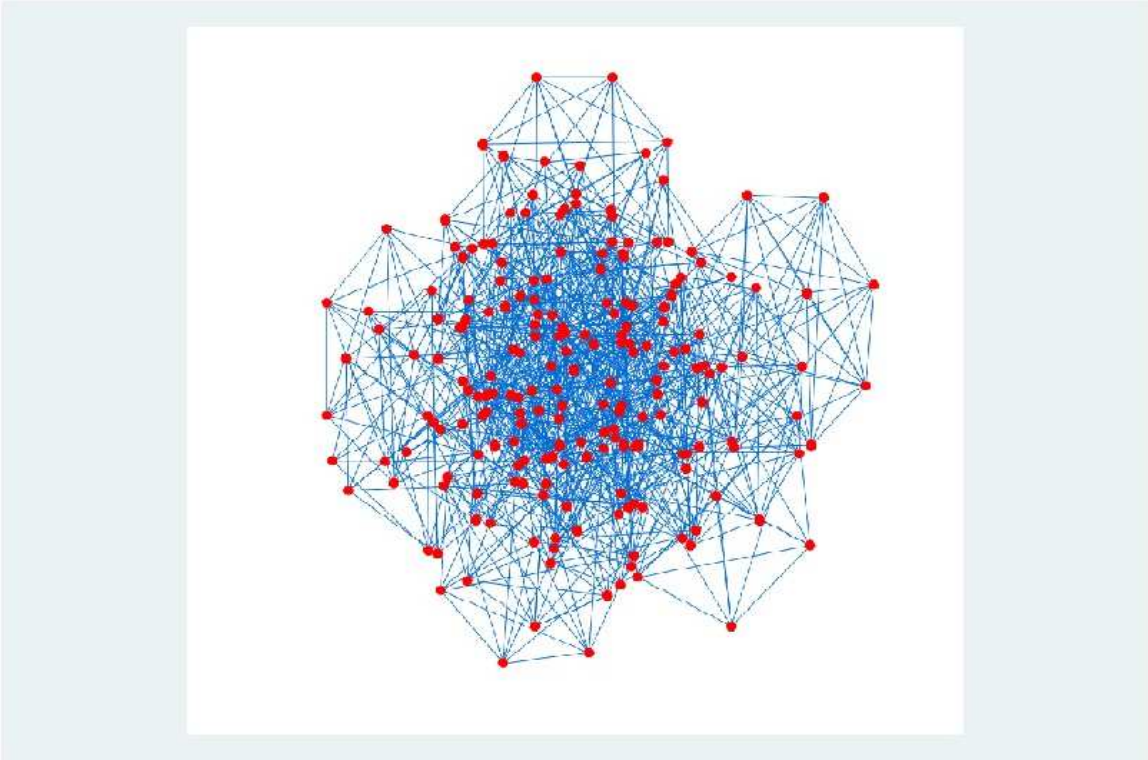
```
. netsis IN OUT , measure(betweenness) name(C,replace)
. netsummarize C/((rows(C)-1)*(rows(C)-2)), ///
generate(betweenness) statistic(rowsum)
```

We can summarize the Degree, Closeness and Betweennes generated variables as follows:

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
Degree       |       179    .0367945    .0075901   .0197044   .0541872
Closeness    |       179    15.26751    15.36465   3.060606        101
Betweennes   |       179    .0004081    .0003381          0   .0014778
-------------+--------------------------------------------------------
```

Finally, we can also draw the graphical representation (Graph 1) of the network considered by the user-written command `netplot`, still taking as arguments the output variables of `datanet`:

```
. netplot IN OUT, type(mds)
```



Source: Authors elaboration based on MIUR data.

The option `type(mds)` specifies the type of layout. In this case, the option `mds` allows for the positions of the vertices to be calculated using multidimensional scaling. This is the default in this Stata routine.

References

Borgatti S. (1995), Centrality and AIDS. Connections, 18 (1) 112-114.

Freeman L. (1997), Centrality in social networks conceptual clarification. Social Networks, 1 (3): 215-239.

Miura, H. (2012), Stata graph library for network analysis, The Stata Journal, Volume 12 Number 1: pp. 94-129.

Appendix A. The Stata ADO-file for `datanet`

```
program define datanet
*! June 19, 2014
version 13
args id x
tempvar idn id1 id2 y id1s id2s id_new1 id_new2 id_new3 id_in id_out
sort `id'
encode `id', gen (`idn')
drop `id'
rename `idn' `id'
order `id' `x'
global n=_N
expand $n
cap drop `id2'
bys id: gen `id2'=_n
label values `id2' `idn'
rename `id' `id1'
order `id1' `id2'
global N=_N
gen `y'=.
forvalues i=1($n)$N{
    local k=1
    forvalues j=1/$n {
       local m=`j'+`i'-1
         if `x'[`m']==`x'[`k']{
                                      replace `y'=`x' in `m'
                                      }
         else{
                              replace `y'=0 in `m'
                    }
      local k = `k'+$n

       }
}
drop if `y'==0
drop if `id1' == `id2'
drop `y'
order `id1' `id2' `x'
tostring `id1' `id2' , gen(`id1s' `id2s')
gen `id_new1'=`id1s'+`id2s'
gen `id_new2'=`id2s'+`id1s'
qui count
global n_new=r(N)
cap drop `id_new3'
gen `id_new3'=.
forvalues i=1/$n_new{
forvalues j=1/$n_new{
if (`id_new1'[`i']==`id_new2'[`j']){
replace `id_new3'=1 in `i'
replace `id_new3'=0 in `j'
}
}
}
drop if `id_new3'==1
decode `id1' , generate(_id_1)
decode `id2' , generate(_id_2)
rename _id_1 IN
rename _id_2 OUT
drop `x'
end
```