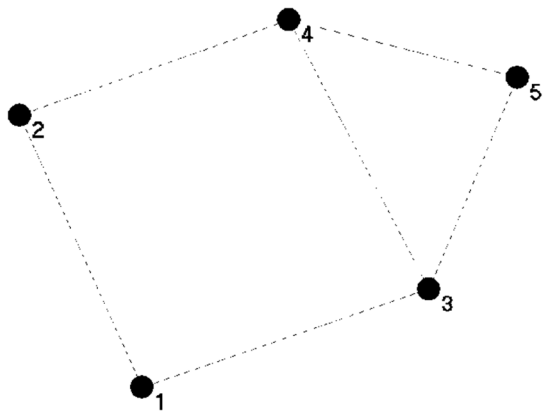
# Edgelists, and merging edgelists in STATA

## What is an edgelist?

An edgelist is the encoding of a graph (network) into a list of edges. The edgelist code every dyad of the graph by using the identifier of the nodes to indicate an edge. For example the edgelist coding of the graph in figure 1 is:

```
1 2
1 3
2 4
3 4
3 5
4 5
```

*Figure 1.*




If the network is asymmetric (nodes are connected by arcs) one would consider the first node in the edgelist as the sender, and the second as a receiver.

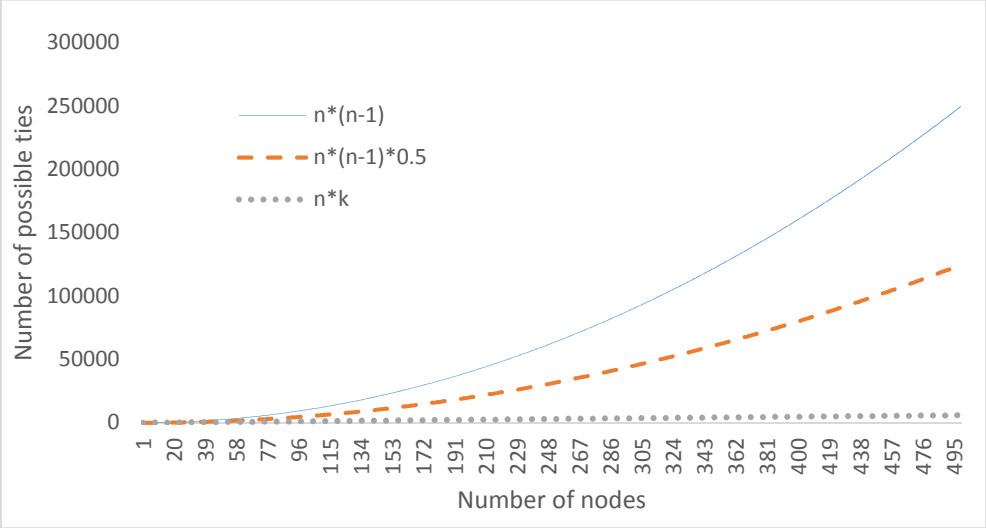
## Why the edgelist is useful for working with big data?

Storing network structures in edgelist format has the desirable property that they require less disk space, or memory for storage. Why? The reason is that in a matrix format one has to encode all possible edges, so it means that non-existent connections are also appear.

In small datasets it does not make a big difference, but as the network grows edgelists provide a considerable advantage because most of the empirical networks are sparse networks. Let me explicate what sparse networks are (or sometimes referred to as sparse matrices). The number of possible ties in a graph with *n* ties is *n***n* (or *n**(*n*-1), if self-ties are ignored). However in most empirical networks the growth rate of possible ties by network size is limited by the number of

maximal ties that a node can sustain. For example we cannot befriend with everybody in a city, because nobody has the time and energy to sustain the required intimacy and closeness with thousands of people. So in this network the number of maximal ties is *n*k*, where *k* refers to the maximal number of friends one can have at a time. Figure 2 shows that in these networks the number of possible ties grows much slower with size then in networks where no maximal tie number is defined, or where it is set according to a proportion of the network. These structures are called sparse networks or sparse matrices, because most of the cells are 0. In figure 2 one can see that the gap between the lines of n*n-1 and n*k is growing exponentially, which means that in sparse networks the number of 0 cells grows fast. Because of their large proportion 0 cells are redundant information, and edge lists, because they are ignore them, save a lot of space in memory and on the hard disk. It is important to understand, that 0 cells are meaningful empirical information, but from an information theoretical point of view they are redundant because of their large proportion and euality.

*Figure 2. Growth of possible ties in networks (n=1-500, k=12)*




**How edgelists are coded in Pajek?**

Pajek (spider in Slovenian) is a software for large network analysis, and one of the most popular network analysis tools. It can read and store networks in various formats. The edgelist format can include more information than the exemplary simple list in the previous example. The pajek edgelist, which is a text file has the file extension .net. It can store the following information (see figure 3):

- At the minimum it has a line *vertices n, where n is the number of nodes, and either a *edges line or a *arcs with some symmetric edges and arcs listed under them.

- Under the *vertices n header one may write the node identifiers ($v_n$), labels, node shapes, and coordinates in 3 dimension.
- Under the *arcs and *edges headers one may put edge weights ($x_n$) next to the edge codes.
- Both headers can appear in the same dataset.

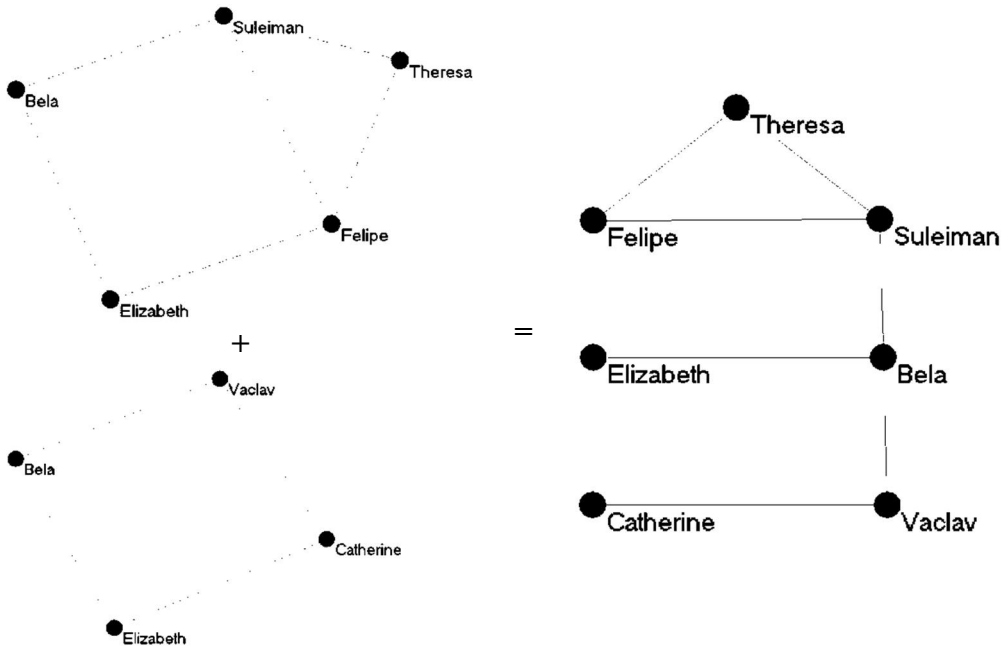
*Figure 3. Pajek edgelist format (n=number of nodes)*

*vertices n
$v_1$ "label$_1$" box $X_1$ $Y_1$ $Z_1$
$v_2$ "label$_2$" ellipse $X_2$ $Y_2$ $Z_2$
… … … … … …
$v_n$ "label$_n$" box $X_n$ $Y_n$ $Z_n$
*arcs
$v_1$ $v_2$ $x_1$
$v_3$ $v_1$ $x_3$
...
*edges
$v_2$ $v_2$ $x_2$
$v_4$ $v_1$ $x_4$
...

**Merging edgelists**

An example of network merging is in figure 4. There must be a set of nodes that has the same label in both datasets. We may need to perform a merge, if we have networks from different contexts that share members. Graphs which has unique nodes are called labeled graphs. It simply means, that each node has some sort of identity. Merging labeled graphs in Pajek is not possible, because Pajek doesn't consider labels of nodes as an important variable. Node identifiers, the number that we use to write edgelists, are not labels, they must always have to be consecutive integers within a network dataset, so they cannot identify unambiguously nodes across datasets.

Figure 4. Merging networks




One can use the net_merge ado file to merge networks. See its help file for details about the input parameters.