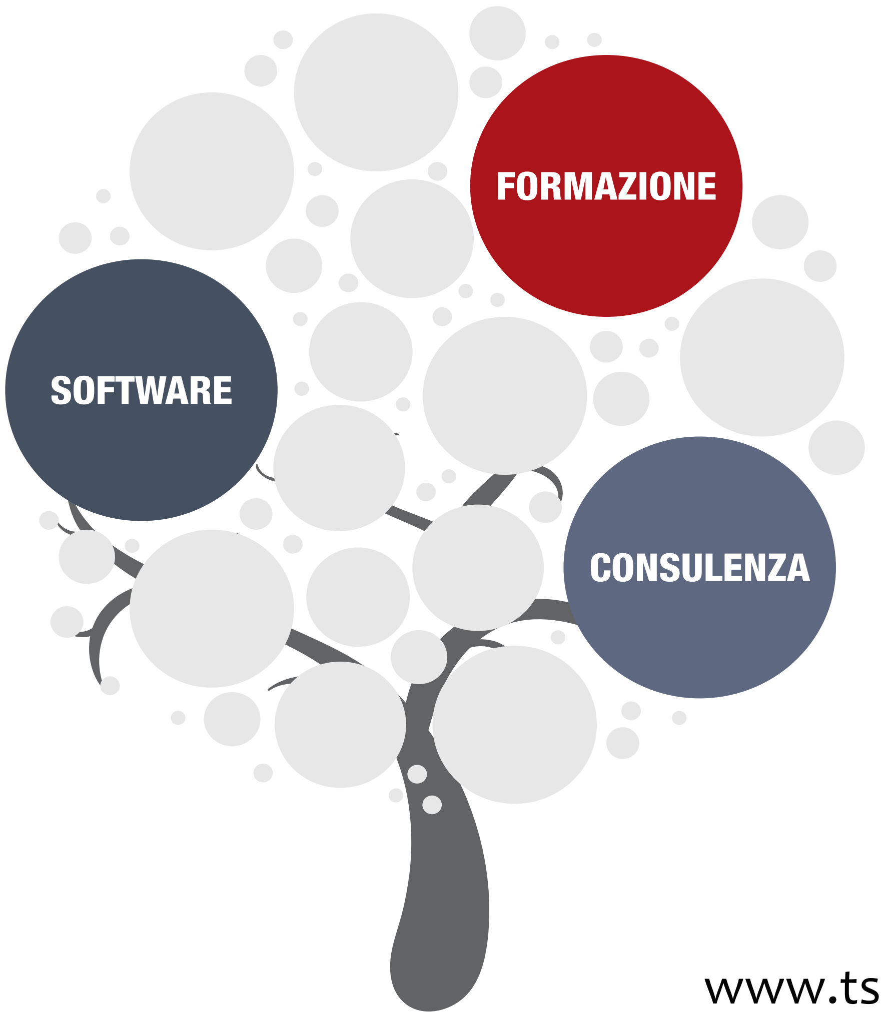
Social Network Analysis Using Stata

Thomas Grund
University College Dublin

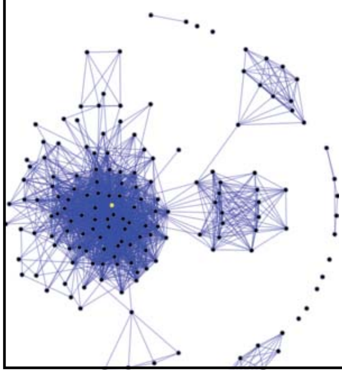
XII Italian Stata Users Group Meeting • Florence, 12 November 2015



www.tstat.it

SOCIAL NETWORK ANALYSIS USING STATA

Thomas Grund
University College Dublin
thomas.u.grund@gmail.com



November 2015
Italian Stata User Group

http://nwcommands.org

/TUTORIALS AND SLIDES

BOOK

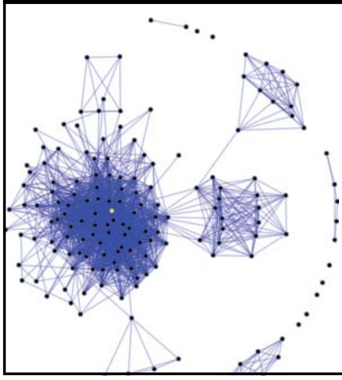
Grund, T. and Hedström, P. (in preparation) Social Network Analysis Using Stata. StataPress.

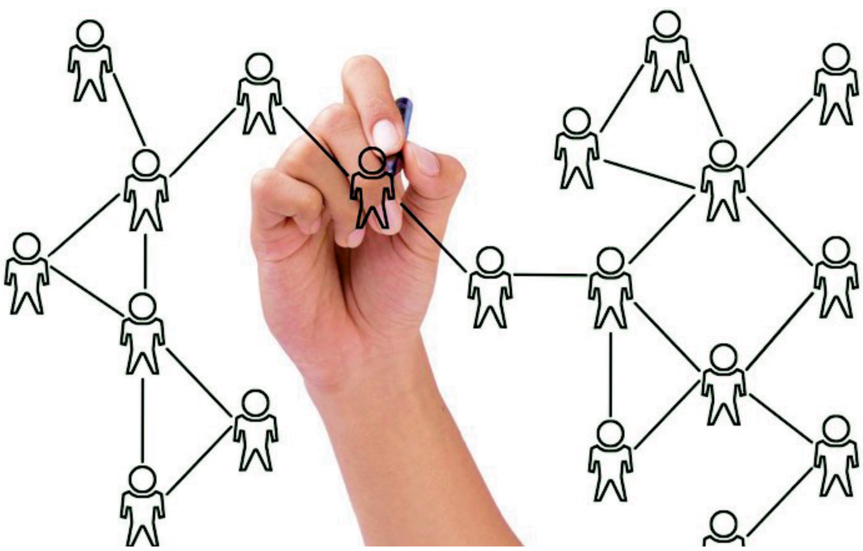
GoogleGroup: nwcommands

Twitter: nwcommands

Search “nwcommands” to find a channel with video tutorials.

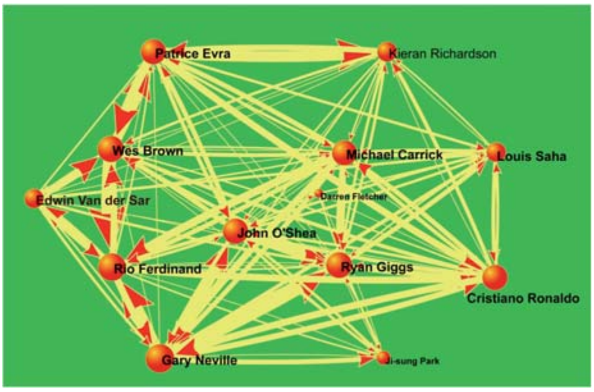
SOCIAL NETWORKS





MANCHESTER UTD – TOTTENHAM

9/9/2006, Old Trafford



SOCIAL NETWORKS

- **Social**
  - Friendship, kinship, romantic relationships
- **Government**
  - Political alliances, government agencies
- **Markets**
  - Trade: flow of goods, supply chains, auctions
  - Labor markets: vacancy chains, getting jobs
- **Organizations and teams**
  - Interlocking directorates
  - Within-team communication, email exchange

DEFINITION

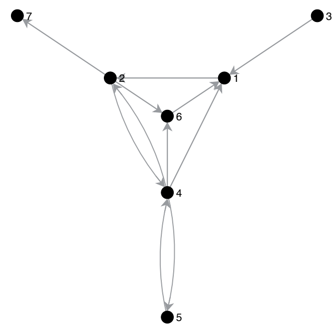
- Mathematically, a (binary) network is defined as $G = (V, E)$ where $V = \{1,2,..,n\}$ is a set of "vertices" (or "nodes") and $E \subseteq \{\langle i,j \rangle | \, i,j \in V\}$ is a set of "edges" (or "ties", "arcs"). Edges are simply pairs of vertices, e.g. $E \subseteq \{(1,2),(2,5) \ldots\}$.
- We write $y_{ij} = 1$ if actors i and j are related to each other (i.e., if $\langle i,j \rangle \in E$), and $y_{ij} = 0$ otherwise.
- In digraphs (or directed networks) it is possible that $y_{ij} \neq y_{ji}$.

ADJACENCY MATRIX

- We write $y_{ij} = 1$ if actors i and j are related to each other (i.e., if $\langle i,j \rangle \in E$), and $y_{ij} = 0$ otherwise
- The matrix $\boldsymbol{y}$ is called the adjacency matrix and is a convenient representation of a network.

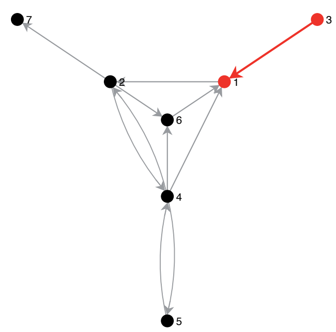
$$\boldsymbol{y} = \begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{nj} & \cdots & y_{nb} \end{bmatrix}$$

ADJACENCY MATRIX



	1	2	3	4	5	6	7
1	0	1	0	0	0	0	0
2	0	0	0	1	0	1	1
3	1	0	0	0	0	0	0
4	1	1	0	0	1	1	0
5	0	0	0	1	0	0	0
6	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0

ADJACENCY MATRIX



	1	2	3	4	5	6	7
1	0	1	0	0	0	0	0
2	0	0	0	1	0	1	1
3	1	0	0	0	0	0	0
4	1	1	0	0	1	1	0
5	0	0	0	1	0	0	0
6	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0

NETWORK ANALYSIS

- Simple description/characterization of networks
- Calculation of node-level characteristics (e.g. centrality)
- Components, blocks, cliques, equivalences…
- Visualization of networks
- Statistical modeling of networks, network dynamics
- ….

Purpose-built

UCINET

Pajek

Gephi makes graphs handy

Excel/R extensions

NODEXL Network Graphs The Social Media Research Foundation

R

C++/Python libraries

SNAP

NetworKit

NetworkX

NWCOMANDS

NWCOMMANDS

- Software package for Stata. Almost 100 new Stata commands for handling, manipulating, plotting and analyzing networks.
- Ideal for existing Stata users. Corresponds to the R packages “network”, “sna”, “igraph”, “networkDynamic”.
- Designed for small to medium-sized networks (< 10000).
- Almost all commands have menus. Can be used like Ucinet or Pajek. Ideal for beginners and teaching.
- Not just specialized commands, but whole infrastructure for handling/dealing with networks in Stata.
- Writing own network commands that build on the nwcommands is very easy.

LINES OF CODE

Type	Files	LoC
.ado	94	14548
.dlg	57	5707
.sthlp	97	9954

Downloads 4833 (since Jan 2015)

INSTALLATION

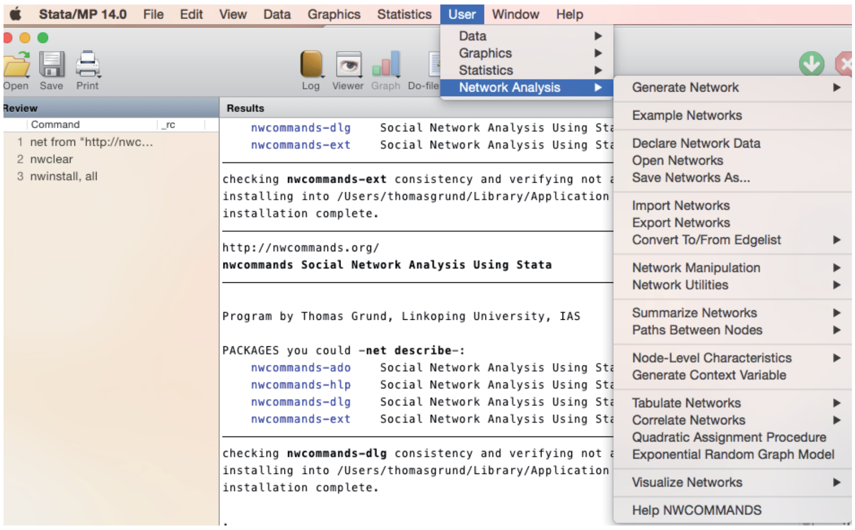
```
. findit nwcommands
```

=> (manually install the package "nwcommands-ado")

Or

```
. net from http://nwcommands.org
. net install "nwcommands-ado"
```

```
. nwinstall, all
```



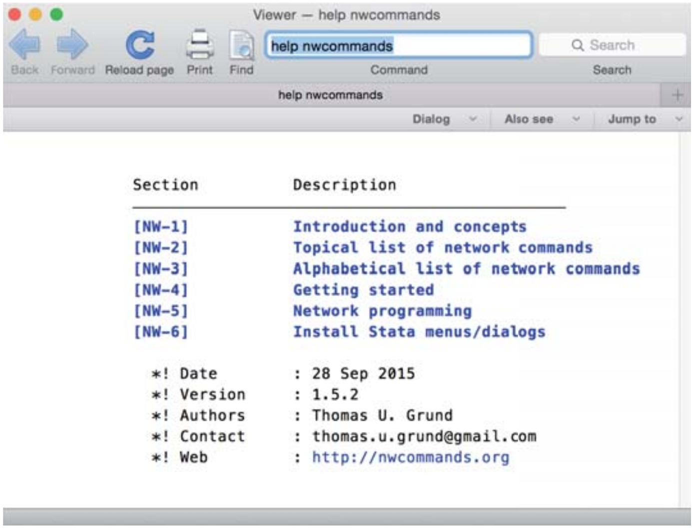
INTUITION

- Software introduces ***netname*** and ***netlist***.
- Networks are dealt with like normal variables.
- Many normal Stata commands have their network counterpart that accept a *netname*, e.g. nwdrop, nwkeep, nwclear, nwtabulate, nwcorrelate, nwcollapse, nwexpand, nwreplace, nwrecode, nwunab and more.
- Stata intuition just works.

NETWORK NAMES AND LISTS

Example	Description
mynet	Just one network
mynet1 mynet2	Two networks
mynet*	All networks starting with mynet
*net	All networks ending with net
my*t	All networks starting with my and ending with t
my~t	One network starting with my and ending with t
my?t	All networks starting with my and ending with t and one character in between
mynet1-mynet6	mynet1, mynet2, ..., mynet6
_all	All networks in memory

OVERVIEW




```
. help nwcommands
```

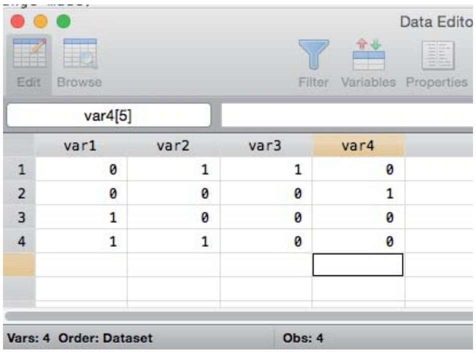
SETTING NETWORKS

- "Setting" a network creates a network quasi-object that has a ***netname***.
- After that you can refer to the network simply by its ***netname***, just like when refer to a variable with its ***varname***.

Syntax:

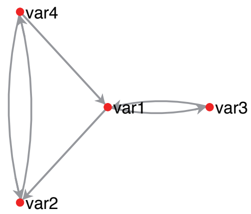
nwset *varlist* [, **edgelist** **directed** **undirected** **name(***newnetname***)** **labs(***string***)** **labsfromvar(***varname***)** **vars(***string***)** **keeporiginal** **xvars**]

nwset, **mat(***matamatrix***)** [**directed** **undirected** **name(***newnetname***)** **labs(***string***)** **labsfromvar(***varname***)** **vars(***string***)** **xvars**]

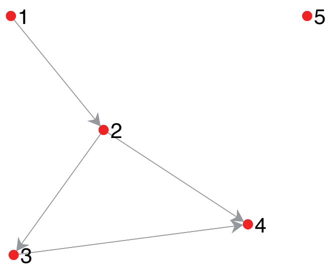


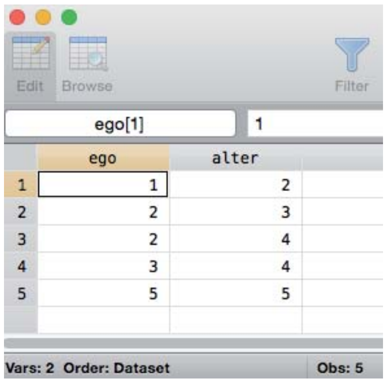
	var1	var2	var3	var4
1	0	1	1	0
2	0	0	0	1
3	1	0	0	0
4	1	1	0	0

Vars: 4 Order: Dataset Obs: 4




```
. nwset _all

. nwplot, lab
```





	ego	alter
1	1	2
2	2	3
3	2	4
4	3	4
5	5	5

Vars: 2 Order: Dataset Obs: 5

```
. nwset ego alter, edgelist

. nwplot, lab
```

LIST ALL NETWORKS

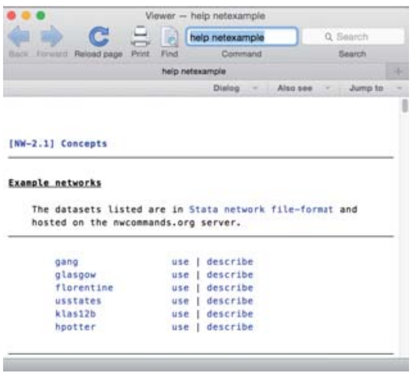
```
. nwds
network     network_1

. nwset
(2 networks)
------------------
      network
      network_1
```

These are the names of the networks in memory. You can refer to these networks by their name.

Check out the return vector. Both commands populate it as well.

LOAD NETWORK FROM THE INTERNET

```
. webnwuse florentine

Loading successful
(4 networks)
------------------
      network
      network_1
      flobusiness
      flomarriage
```




```
. help netexample
```

IMPORT NETWORK

- A wide array of popular network file-formats are supported, e.g. Pajek, Ucinet, by **`nwimport`**.
- Files can be imported directly from the internet as well.
- Similarly, networks can be exported to other formats with **`nwexport`**.

```
. nwimport http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/zachary.dat, type(ucinet)
______________________________
Importing successful
(6 networks)
____________________
      network
      network_1
      flobusiness
      flomarriage
      ZACHE
      ZACHC
```

SAVE/USE NETWORKS

- You can save network data (networks plus all normal Stata variables in your dataset) in almost exactly the same way as normal data.
- Instead of **save**, the relevant command is **nwsave**.
- Instead of **use**, the relevant command is **nwuse**.

DROP/KEEP NETWORKS

- Dropping and keeping networks works almost exactly like dropping and keeping variables.

```
. nwdrop flo*

. nwkeep ZACHE ZACHC
```

DROP/KEEP NODES

You can also drop/keep nodes of a specific network.

```
. nwdrop flomarriage if _nodevar == "strozzi"

. nwdrop flomarriage if _n == 1
```


. nwclear

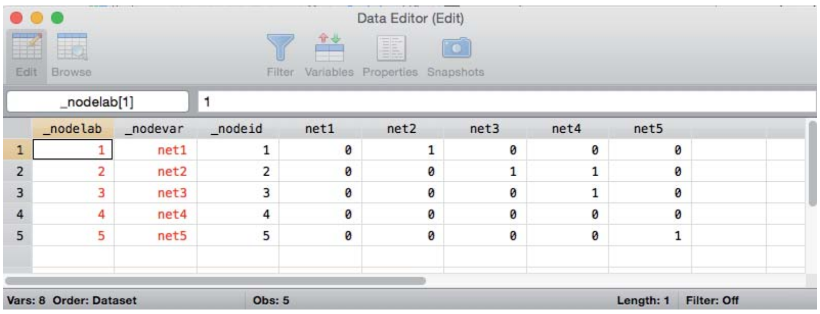

EXAMINE NETWORK

SUMMARIZE

```
. nwsummarize network_1

    Network name:   network_1
    Network id:   1
    Directed: true
    Nodes: 5
    Arcs: 4
    Minimum value:   0
    Maximum value:   1
    Density:   .2
```

OBTAIN TIE VALUES

```
. nwload network_1

. edit
```



	_nodelab	_nodevar	_nodeid	net1	net2	net3	net4	net5
1	1	net1	1	0	1	0	0	0
2	2	net2	2	0	0	1	1	0
3	3	net3	3	0	0	0	1	0
4	4	net4	4	0	0	0	0	0
5	5	net5	5	0	0	0	0	1

TABULATE NETWORK

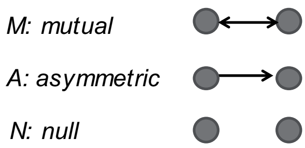
```
. webnwuse florentine, nwclear

Loading successful
(2 networks)
____________________

      flobusiness
      flomarriage

. nwtabulate flomarriage

   Network:  flomarriage       Directed: false

flomarriage |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        100       83.33       83.33
          1 |         20       16.67      100.00
------------+-----------------------------------
      Total |        120      100.00
```

TABULATE TWO NETWORKS

```
. nwtabulate flomarriage flobusiness

   Network 1:  flomarriage    Directed: false
   Network 2:  flobusiness    Directed: false

flomarriag |      flobusiness
         e |         0          1 |     Total
-----------+----------------------+----------
         0 |        93          7 |       100
         1 |        12          8 |        20
-----------+----------------------+----------
     Total |       105         15 |       120
```

DYAD CENSUS

```
. webnwuse glasgow

Loading successful
(3 networks)
____________________
      glasgow1
      glasgow2
      glasgow3
```




```
. nwdyads glasgow1
```

Dyad census: glasgow1

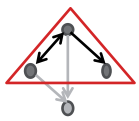
Mutual	Asym	Null
39	35	1151

Reciprocity: .527027027027027

```
. nwtriads glasgow1
```

Triad census: glasgow1

003	012	021D	021U
16243	1470	5	18
021C	**030T**	**030C**	**102**
21	5	0	1724
120D	**120U**	**120C**	**111D**
6	5	2	42
111U	**201**	**210**	**300**
30	15	9	5



Transitivity: .3870967741935484

CHANGE NETWORK

TABULATE NETWORK

```
. webnwuse gang, nwclear

. nwtabulate gang_valued

   Network:  gang_valued        Directed: false
```

gang_valued	Freq.	Percent	Cum.
0	1,116	77.99	77.99
1	182	12.72	90.71
2	92	6.43	97.13
3	25	1.75	98.88
4	16	1.12	100.00
Total	1,431	100.00	

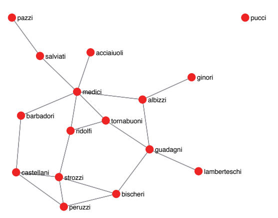
RECODE TIE VALUES

```
. nwrecode gang_valued (2/4 = 99)

(gang_valued: 266 changes made)

. nwtabulate gang_valued

   Network:  gang_valued       Directed: false
```

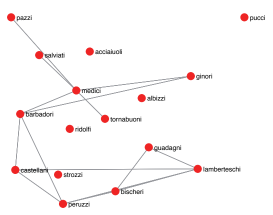
gang_valued	Freq.	Percent	Cum.
0	1,116	77.99	77.99
1	182	12.72	90.71
99	133	9.29	100.00
Total	1,431	100.00	

FLORENTINE FAMILIES

```
. webnwuse florentine, nwclear

Loading successful
(2 networks)
------------------
      flobusiness
      flomarriage
```



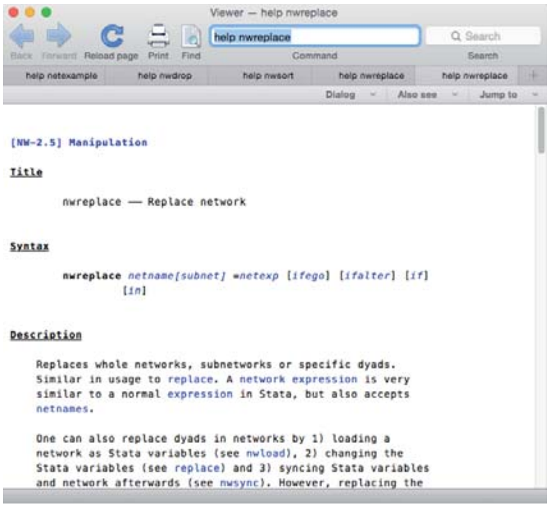
Marriage ties



Business ties

REPLACE TIE VALUES

```
. nwreplace flomarriage = 2 if flobusiness == 1 & flomarriage == 1

. nwtabulate flomarriage

   Network:  flomarriage       Directed: false

flomarriage |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        100       83.33       83.33
          1 |         12       10.00       93.33
          2 |          8        6.67      100.00
------------+-----------------------------------
      Total |        120      100.00
```

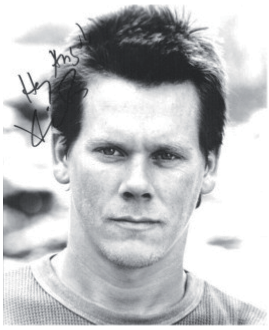



```
. help nwreplace
```

DISTANCE AND PATH

Kevin Bacon

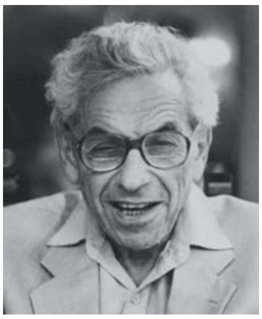
?



http://oracleofbacon.org/

Paul Erdős

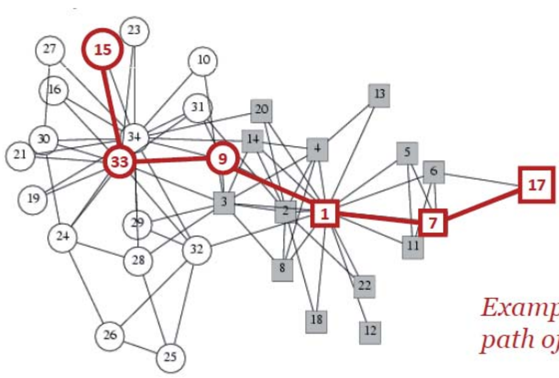
?



http://academic.research.micros
oft.com/VisualExplorer

DISTANCE

Length of a shortest connecting path defines the (geodesic) distance between two nodes.

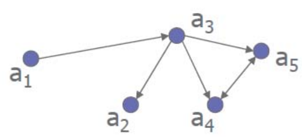


Example of a shortest path of length 5

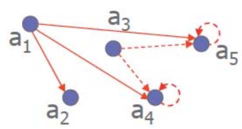
DISTANCE

How can we calculate the distance?

- Matrix y indicates which row actor is directly connected to which column actor.
- The squared matrix y^2 indicates which row actor can reach which column actor in two steps.
- The matrix y^l indicates who reaches whom in l steps.



$$y^2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} \end{pmatrix}$$



DISTANCE

When we take the average of the shortest paths between all nodes (if all are connected) we get the "average shortest path length" ℓ of the network.

Intuition: If we were to select two nodes at random, how many steps would it take 'on average' to connect them?

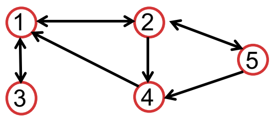
For a random graph one can show that:

$$\ell \approx \frac{\ln(n)}{\ln(k)}$$

n = number of nodes

k = average degree of nodes

DISTANCE



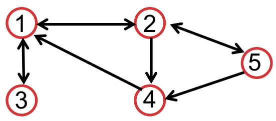
$$distances = \begin{bmatrix} 0 & 1 & 1 & 2 & 2 \\ 1 & 0 & 2 & 1 & 1 \\ 1 & 2 & 0 & 3 & 3 \\ 1 & 2 & 2 & 0 & 3 \\ 2 & 1 & 3 & 1 & 0 \end{bmatrix}$$

$$avgerage\ shortest\ path\ length = \ 1.8$$

DISTANCE DISTRIBUTION

- Networks can have the same “average shortest path length”, but still be vastly different from each other.
- Better, look at the “distribution of shortest paths” instead of the average.
  - Calculate how often each distance occurs.

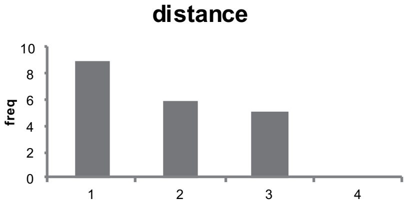
$$\begin{bmatrix} 0 & 1 & 1 & 2 & 2 \\ 1 & 0 & 2 & 1 & 1 \\ 1 & 2 & 0 & 3 & 3 \\ 1 & 2 & 3 & 0 & 3 \\ 2 & 1 & 3 & 1 & 0 \end{bmatrix}$$



DISTANCE DISTRIBUTION

- Networks can have the same “average shortest path length”, but still be vastly different from each other.
- Better, look at the “distribution of shortest paths” instead of the average.
  - Calculate how often each distance occurs.

$$\begin{bmatrix} 0 & 1 & 1 & 2 & 2 \\ 1 & 0 & 2 & 1 & 1 \\ 1 & 2 & 0 & 3 & 3 \\ 1 & 2 & 3 & 0 & 3 \\ 2 & 1 & 3 & 1 & 0 \end{bmatrix}$$



DISTANCE

```
. webnwuse florentine, nwclear

. nwgeodesic flomarriage
------------------------------------------
 Network name: flomarriage
 Network of shortest paths: geodesic
------------------------------------------
   Nodes: 16
   Symmetrized : 1
   ---------------------------------------
   Paths (largest component) : 105
   Diameter (largest component): 5
   Average shortest path (largest component): 2.485714285714286
```

DISTANCE

```
. nwset
(3 networks)
--------------------
      flobusiness
      flomarriage
      geodesic
```

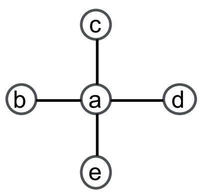
```
. nwtabulate geodesic

  Network:  geodesic      Directed: false

   geodesic |      Freq.     Percent        Cum.
------------+-----------------------------------
         -1 |         15       12.50       12.50
          1 |         20       16.67       29.17
          2 |         35       29.17       58.33
          3 |         32       26.67       85.00
          4 |         15       12.50       97.50
          5 |          3        2.50      100.00
------------+-----------------------------------
      Total |        120      100.00
```

CENTRALITY

CENTRALITY

Well connected actors are in a structurally advantageous position.

- Getting jobs
- Better informed
- Higher status
- …

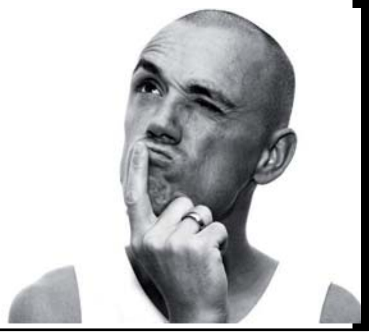


CENTRALITY

Well connected actors are in a structurally advantageous position.

- Getting jobs
- Better informed
- Higher status
- …

What is "well-connected?"



DEGREE CENTRALITY

Degree centrality

- We already know this. Simply the number of incoming/outgoing ties => indegree centrality, outdegree centrality
- How many ties does an individual have?

$$C_{odegree}(i) = \sum_{j=1}^{N} y_{ij} \qquad C_{idegree}(i) = \sum_{j=1}^{N} y_{ji}$$

DEGREE CENTRALITY

Degree centrality

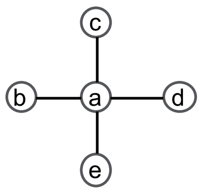
$$C_{degree}(i) = \sum_{j=1}^{N} y_{ij}$$

$C_{degree}(a) = 4$

$C_{degree}(b) = 1$

$C_{degree}(c) = 1$

…



CLOSENESS CENTRALITY

Closeness centrality

- How close is an individual (on average) from all other individuals?

Farness

- How many steps (on average) does it take an individual to reach all other individuals?

$$Farness(i) = \frac{1}{N-1}\sum_{j=1}^{N} l_{ij}$$

$j \neq i$

l_{ij} = shortest path between i and j

FARNESS

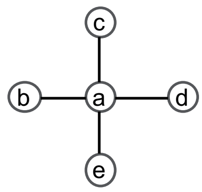
Farness

$$Farness(i) = \frac{1}{N-1}\sum_{j=1}^{N} l_{ij}$$

$$Farness(a) = \frac{1}{4}(1+1+1+1) = 1$$

$$Farness(b) = \frac{1}{4}(1+2+2+2) = \frac{7}{4}$$

...



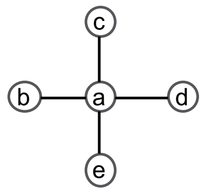
CLOSENESS CENTRALITY

$$C_{closeness}(i) = \frac{1}{Farness(i)}$$

$$C_{closeness}(a) = 1/\left[\frac{1}{4}(1+1+1+1)\right] = 1$$

$$C_{closeness}(b) = 1/\left[\frac{1}{4}(1+2+2+2)\right] = \frac{4}{7}$$

...



BETWEENNESS CENTRALITY

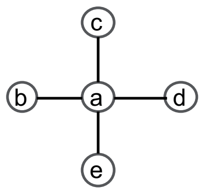
Betweeness centrality

- How many shortest paths go through an individual?

$C_{betweenness}(a) = 6$

$C_{betweenness}(b) = 0$

…

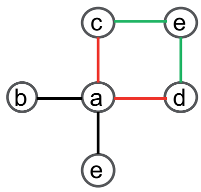


BETWEENNESS CENTRALITY

Betweeness centrality

- How many shortest paths go through an individual?

What about multiple shortest paths? E.g. there are two shortest paths from c to d (one via a and another one via e)



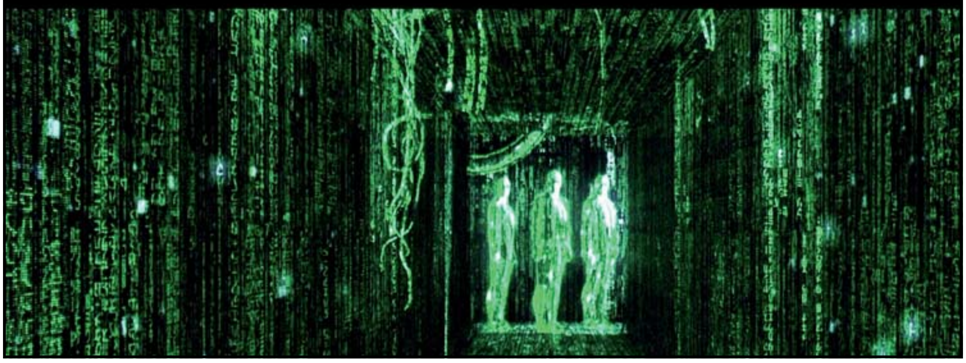
Give each shortest path a weight inverse to how many shortest paths there are between two nodes.

```
. nwbetween flomarriage

  Network name: flomarriage

    Betweenness centrality

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
    _between |         16        19.5    24.60111          0         95

. list _nodelab _between

       +-----------------------+
       |   _nodelab   _between |
       |-----------------------|
    1. | acciaiuoli          0 |
    2. |    albizzi   38.66667 |
    3. |  barbadori         17 |
    4. |   bischeri         19 |
```

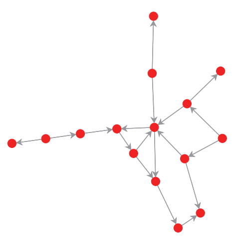
CENTRALITY

nwdegree
nwbetween
nwevcent
nwcloseness
nwkatz

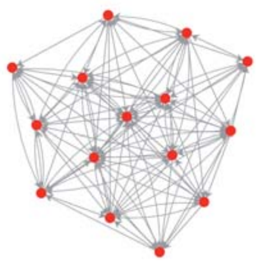
SIMULATION



RANDOM NETWORK



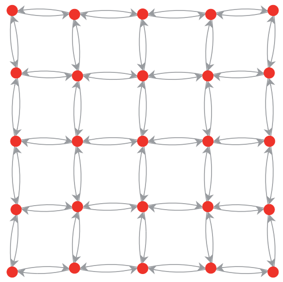
`nwrandom 15, prob(.1)`



`nwrandom 15, prob(.5)`

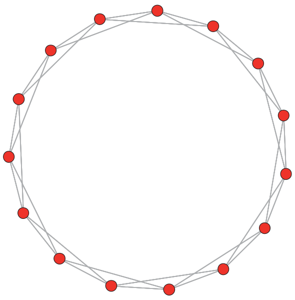
Each tie has the same probability to exist, regardless of any other ties.

LATTICE



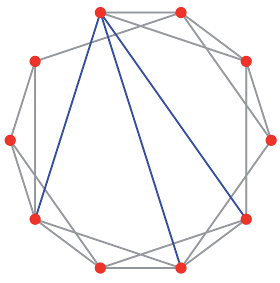
nwlattice 5 5

RING LATTICE



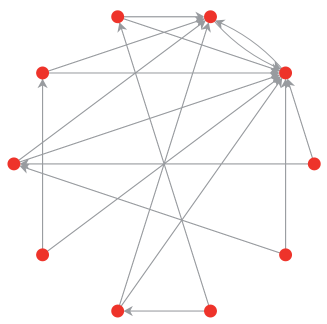
nwring 15, k(2) undirected

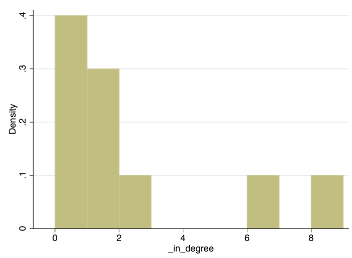
SMALL WORLD NETWORK



nwsmall 10, k(2) shortcuts(3) undirected

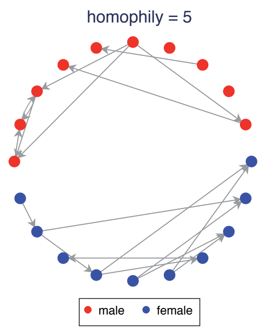
PREFERENTIAL ATTACHMENT NETWORK

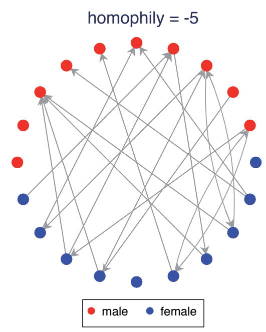





```
nwpref 10, prob(.5)
```

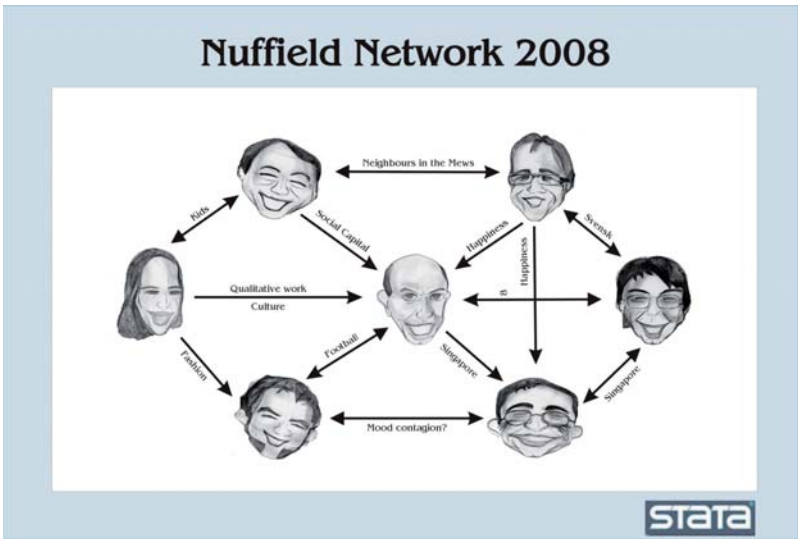
HOMOPHILY NETWORK

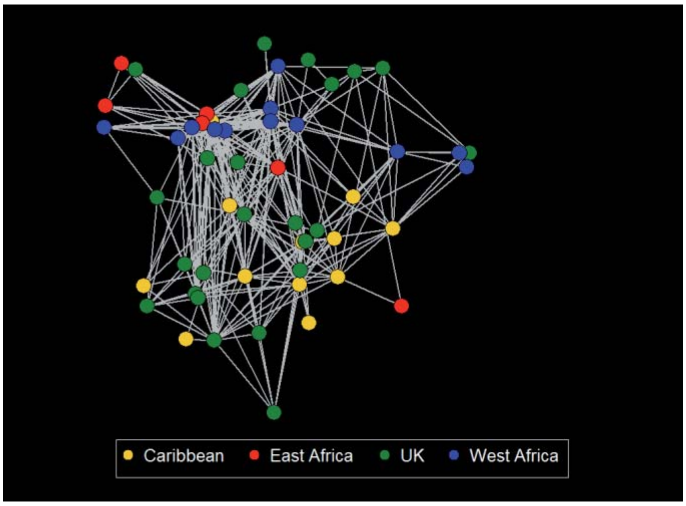




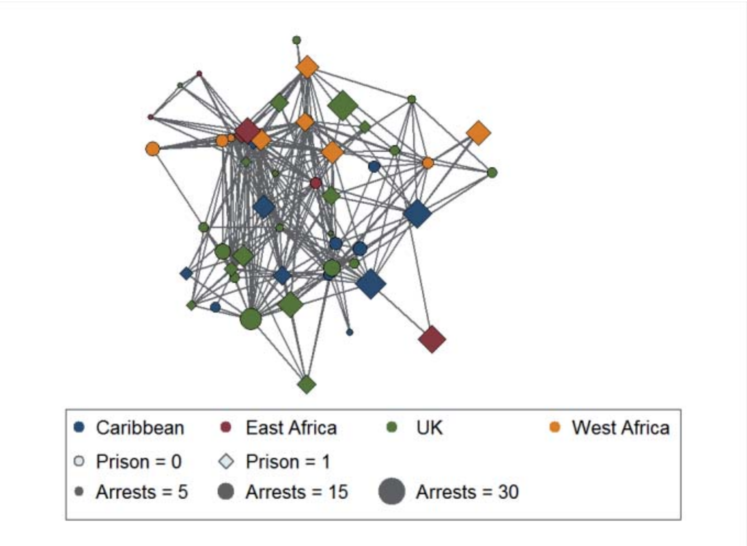

```
nwhomophily gender, density(0.05) homophily(5)
```

VISUALIZATION

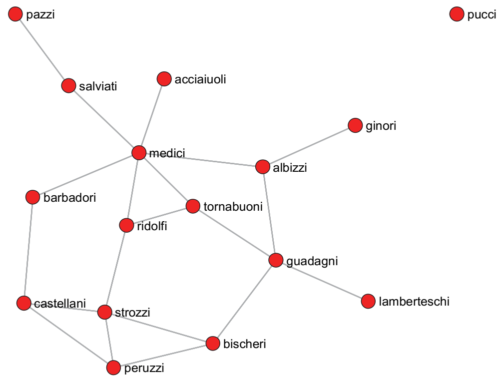




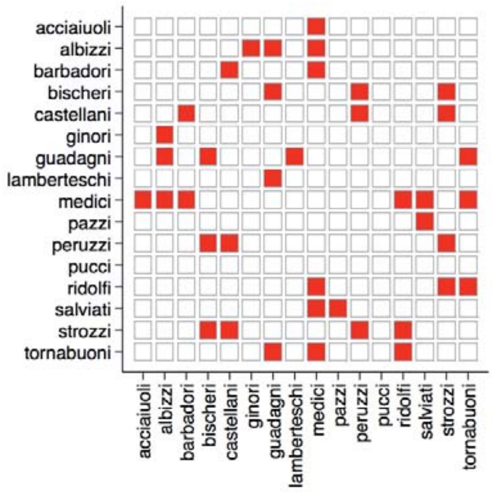

```
. webnwuse gang
. nwplot gang, color(Birthplace)
```



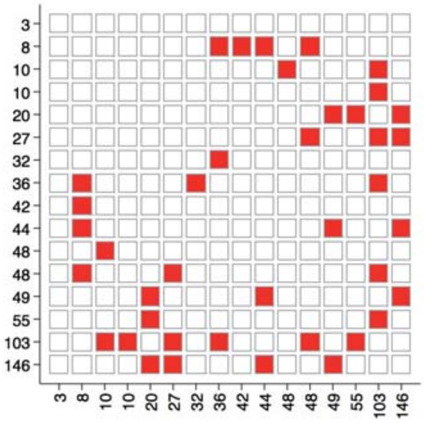

```
nwplot gang, color(Birthplace) symbol(Prison) size(Arrests)
```



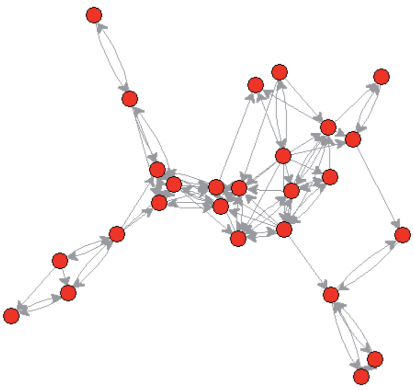

```
. webnwuse florentine
. nwplot flomarriage, lab
```



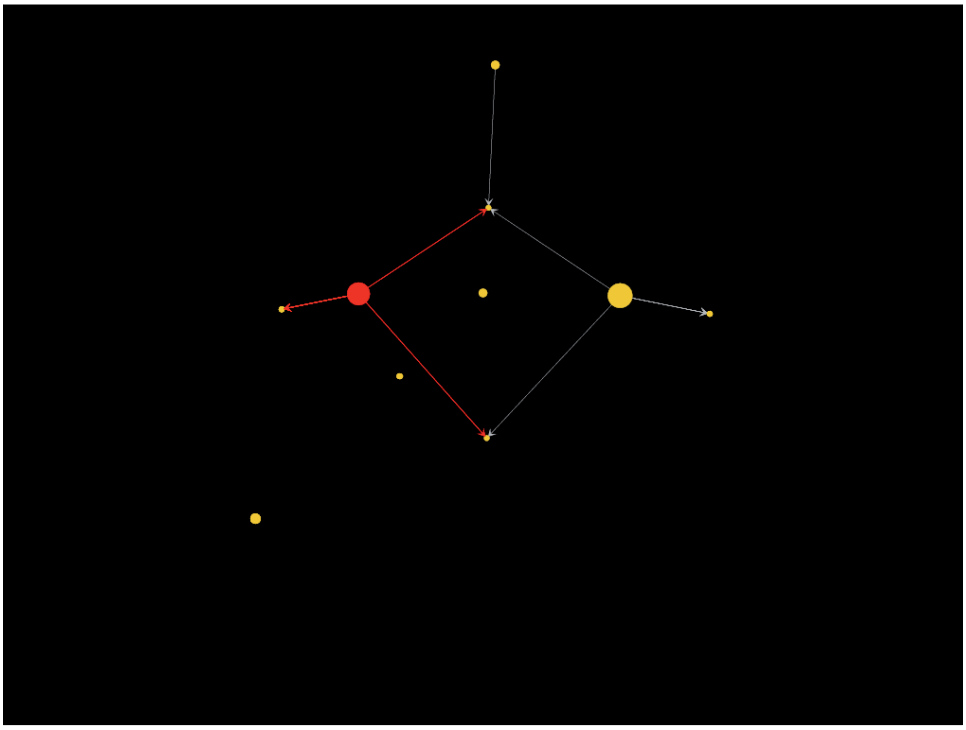

```
. nwplotmatrix flomarriage, lab
```

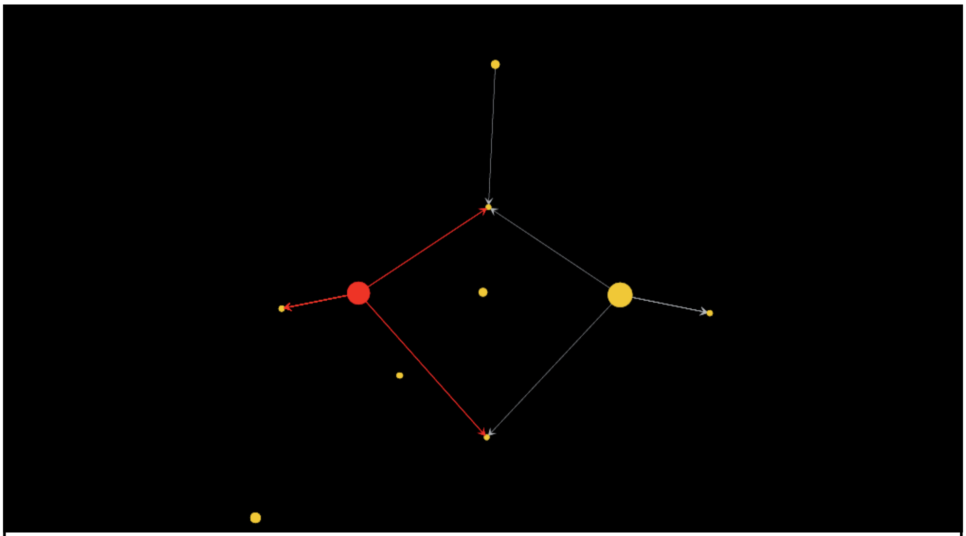



```
. nwplotmatrix flomarriage, sortby(wealth) label(wealth)
```



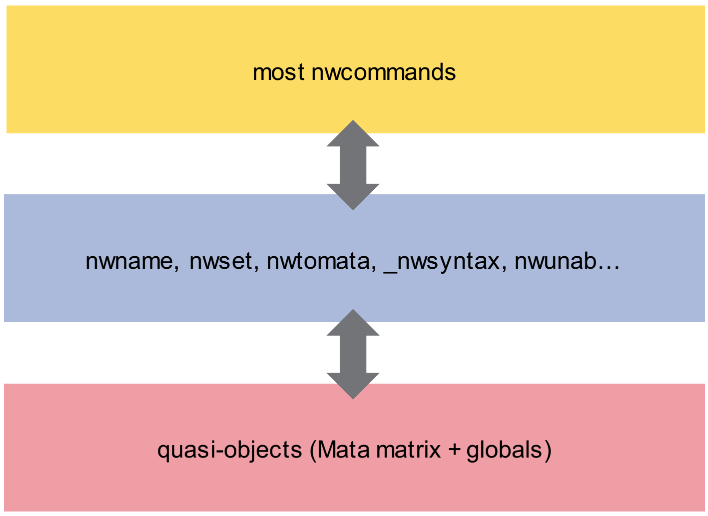
```
. webnwuse klas12
. nwmovie klas12_wave1-klas12_wave4
```





```
. nwmovie _all, colors(col_t*) sizes(siz_t*) edgecolors(edge_t*)
```

UNDER THE HOOD



THREE STEPS IN PROGRAMS

1. Parse network
2. Obtain adjacency matrix and meta-information
3. Perform some calculation with the adjacency matrix

EXAMPLE: OUTDEGREE

```
capture program drop myoutdegree
program myoutdegree
    syntax [anything]
    _nwsyntax `anything'

    nwtomata `netname', mat(net)

    mata: outdegree = rowsum(net)
    getmata outdegree

    mata: mata drop net outdegree
end
```

EXAMPLE: OUTDEGREE

```
capture program drop myoutdegree
program myoutdegree
    syntax [anything]
    _nwsyntax `anything'

    nwtomata `netname', mat(net)

    mata: outdegree = rowsum(net)
    getmata outdegree

    mata: mata drop net outdegree
end
```

Parse networks. Populate local “netname”.

EXAMPLE: OUTDEGREE

```
capture program drop myoutdegree
program myoutdegree
    syntax [anything]
    _nwsyntax `anything'

    nwtomata `netname', mat(net)

    mata: outdegree = rowsum(net)
    getmata outdegree

    mata: mata drop net outdegree
end
```

Obtain adjacency matrix “net”

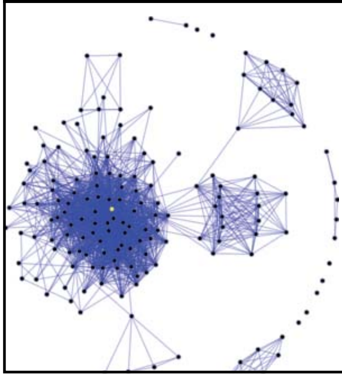
EXAMPLE: OUTDEGREE

```
capture program drop myoutdegree
program myoutdegree
    syntax [anything]
    _nwsyntax `anything'

    nwtomata `netname', mat(net)

    mata: outdegree = rowsum(net)
    getmata outdegree

    mata: mata drop net outdegree
end
```

Functionality

SOCIAL NETWORK ANALYSIS USING STATA

Thomas Grund
University College Dublin
thomas.u.grund@gmail.com



November 2015
Italian Stata User Group

http://nwcommands.org

http://grund.co.uk