SOCIAL NETWORK ANALYSIS USING STATA

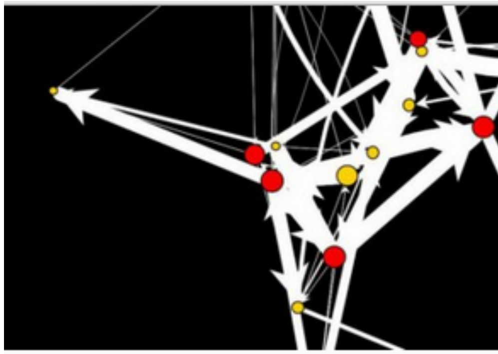
11 September 2015
UK Stata Group, London

Thomas Grund
thomas.u.grund@gmail.com

www.grund.co.uk
www.nwcommands.org

http://nwcommands.org

http://nwcommands.org



NETWORK ANALYSIS USING STATA

nwcommands.org

- ABOUT
- NEWS
- INSTALLATION
- GETTING STARTED
- GLOSSARY
- TUTORIALS AND SLIDES

About

Here you find the the beta-version of the nwcommands – a collection of programs for social network analysis in Stata.

A more thorough description will follow.

Browse through the tutorials and the alphabetical list of the nwcommands to get a first idea about how you can do social network analysis in Stata.

Installation instructions are here.

If you have a question, you can ask it in the forum for the nwcommands. Alternatively, you can send an email to thomas.u.grund@gmail.com. You can also join the email list for the nwcommands here: https://groups.google.com/forum/#!forum/nwcommands/join. Once you are signed up you will receive information about updates, new releases and so on.

If you find any bugs in the software, please contact us by sending an email

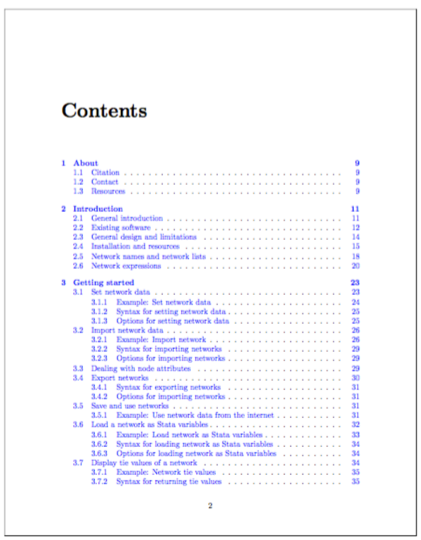
GoogleGroup: nwcommands

Twitter: nwcommands

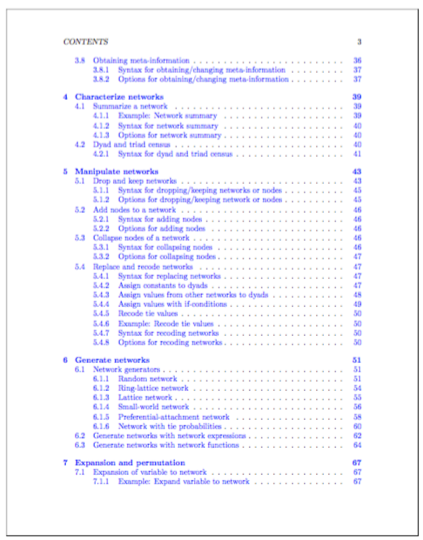
Search “nwcommands” to find a channel with video tutorials.

BOOK

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



Contents



WORKSHOPS

14 November, Florence, Italian Stata Group

11/12 and 18/19 December, Cologne, University of Cologne

12-15 April 2016, Rome, TStat S.r.l.

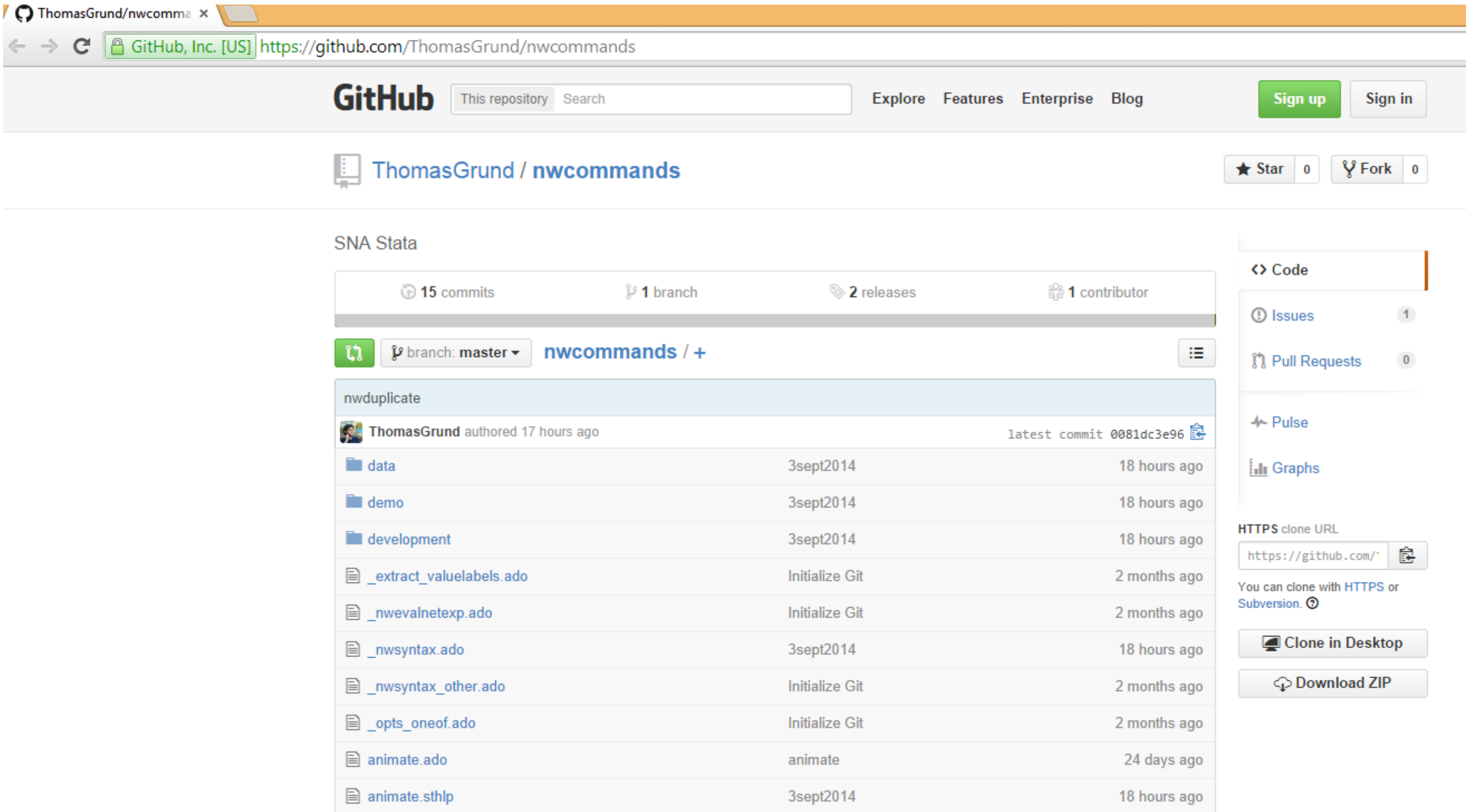
August 2016, Stockholm, Metrika

NWCOMMANDS

- Software package for Stata. Almost 100 new Stata commands for 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.
- Commands for centrality, paths, equivalences, MR-QAP, ERGM (wrapper)...
- 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.

GITHUB

HTTPS://GITHUB.COM/THOMASGRUND/NWCOMMANDS




STATA®

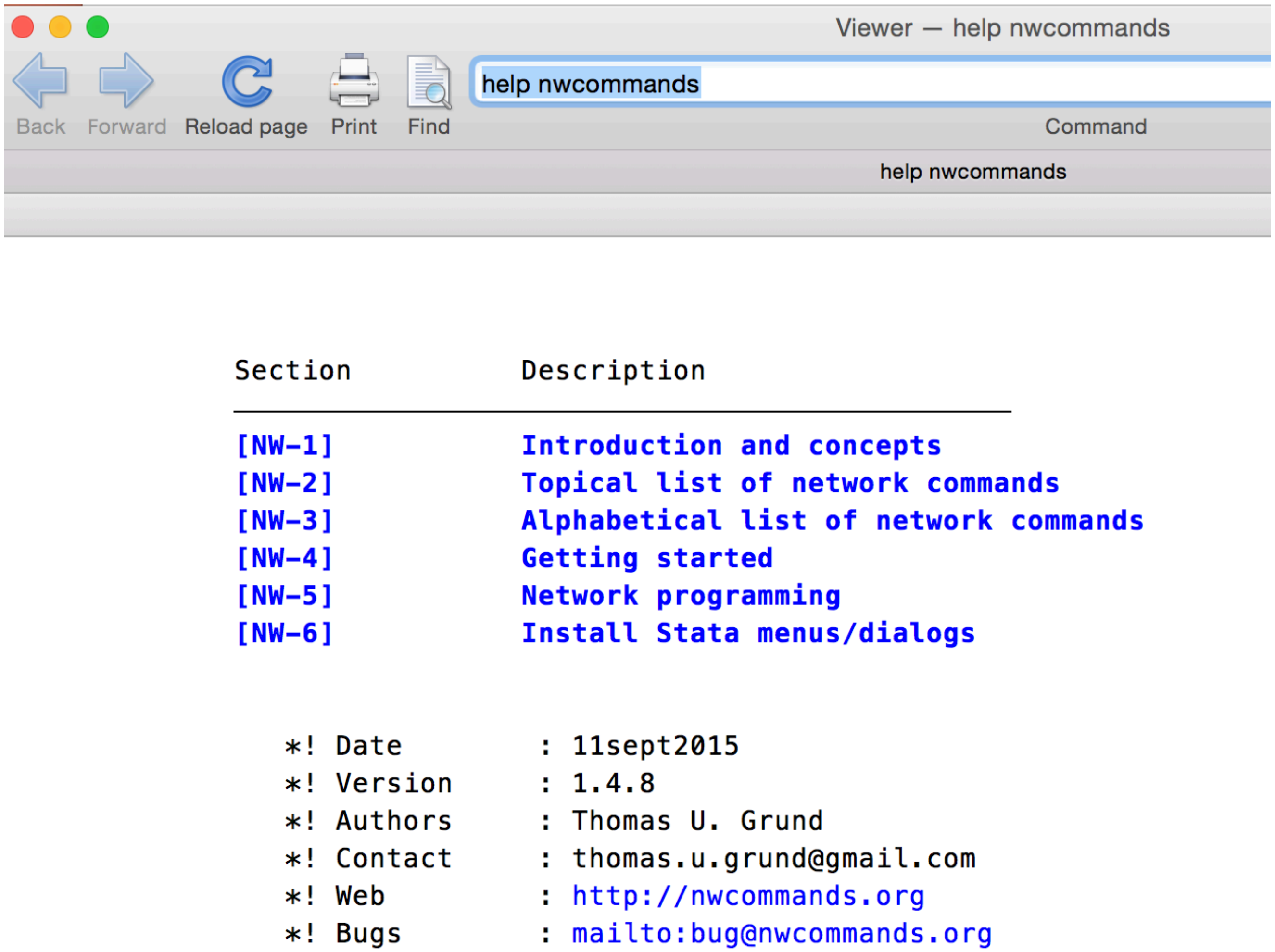
INSTALLATION

```
. findit nwcommands
```

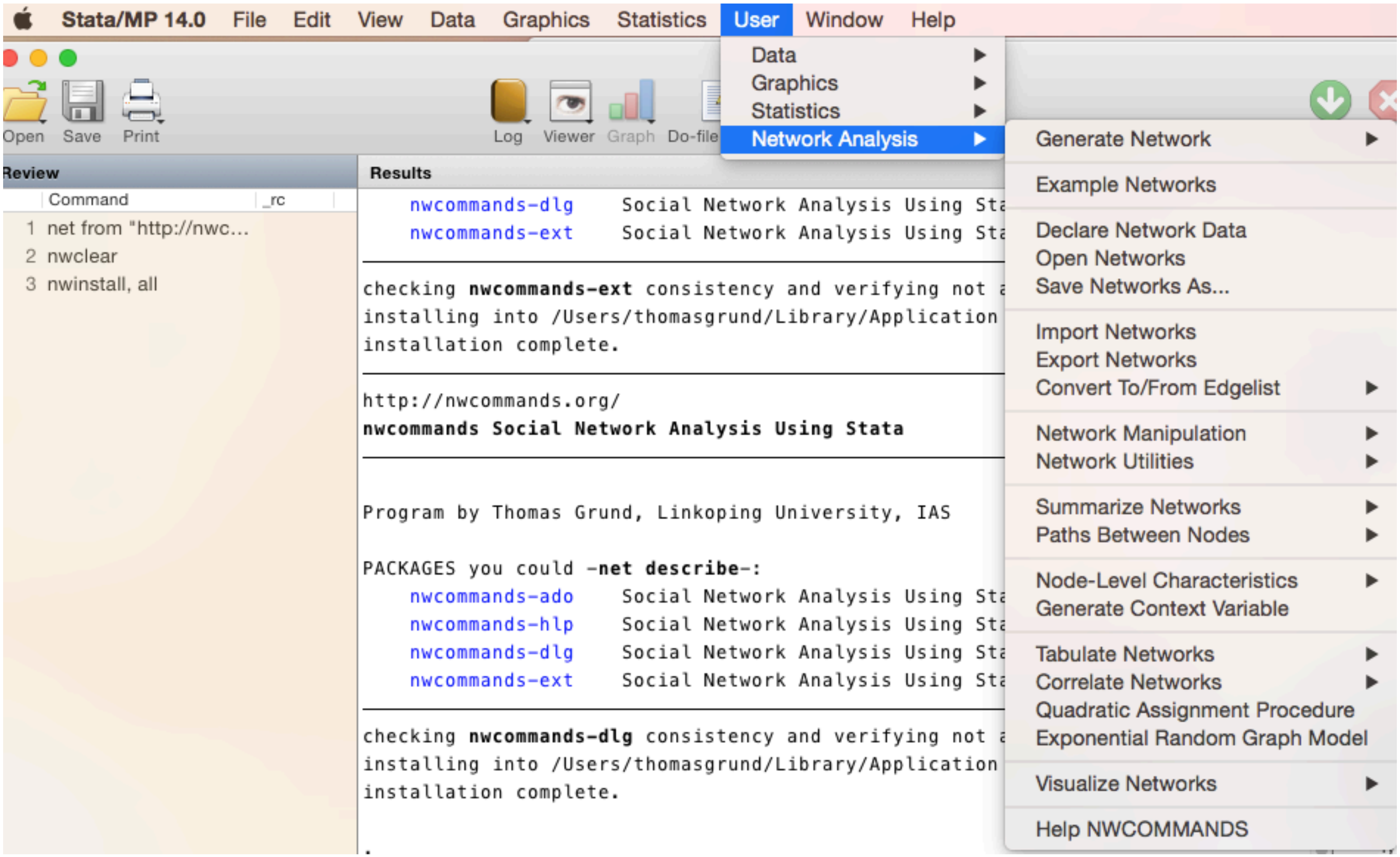
=> (manually install the package ”nwcommands-ado”)

Or

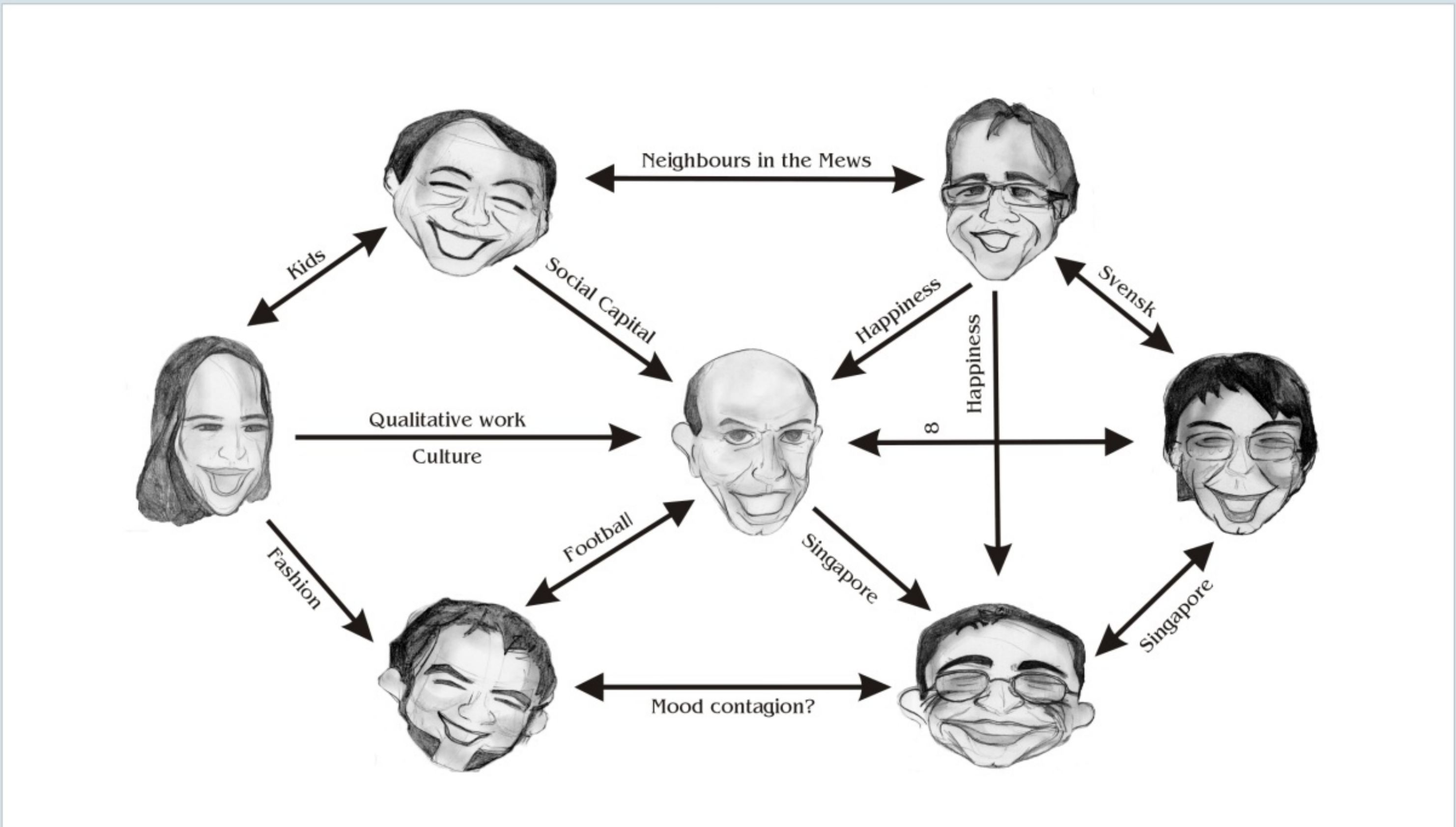
```
. net from http://nwcommands.org
. net install ”nwcommands-ado”

. nwinstall, all
```




```
. help nwcommands
```

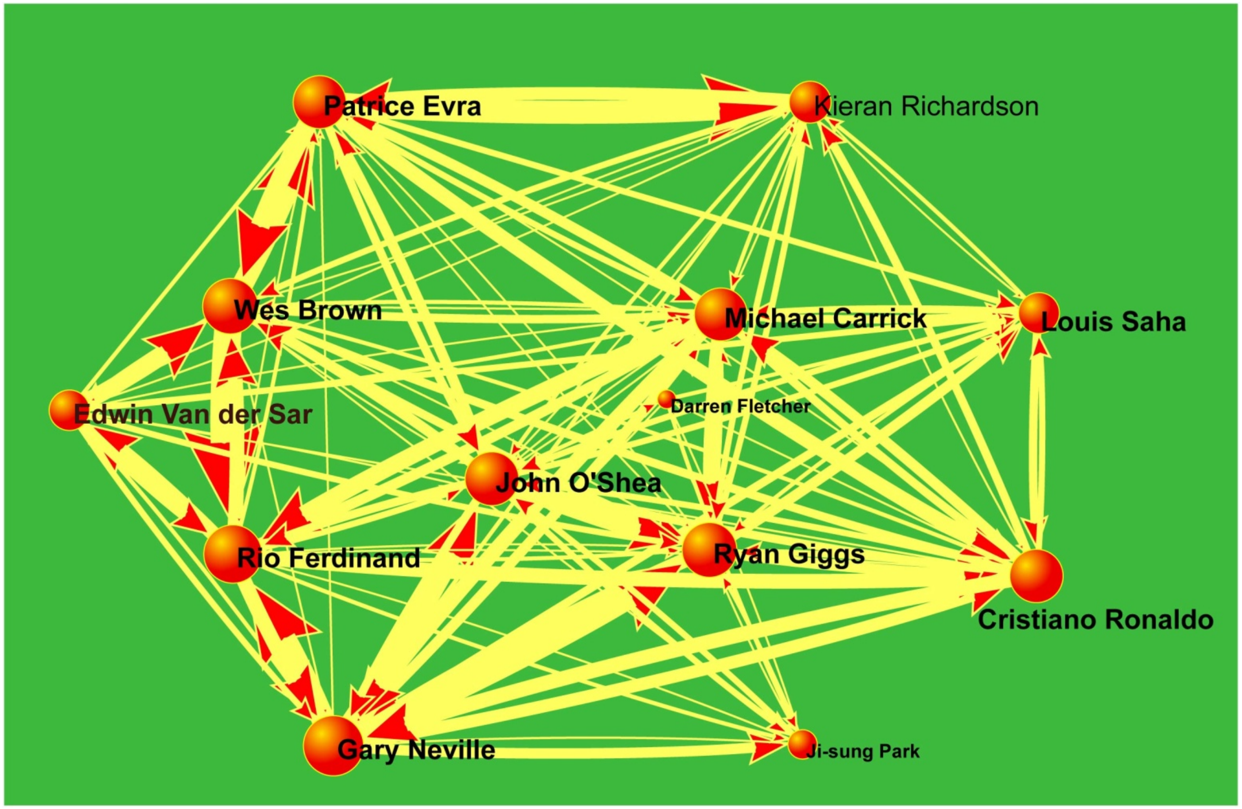

Stata/MP 14.0
File
Edit
View
Data
Graphics
Statistics
User
Window
Help
Open
Save
Print
Log
Viewer
Graph
Do-file
Review
Command
_rc
1 net from "http://nwc...
2 nwclear
3 nwinstall, all
Results
nwcommands-dlg Social Network Analysis Using Sta
nwcommands-ext Social Network Analysis Using Sta
checking nwcommands-ext consistency and verifying not a
installing into /Users/thomasgrund/Library/Application
installation complete.
http://nwcommands.org/
nwcommands Social Network Analysis Using Stata
Program by Thomas Grund, Linkoping University, IAS
PACKAGES you could -net describe-:
nwcommands-ado Social Network Analysis Using Sta
nwcommands-hlp Social Network Analysis Using Sta
nwcommands-dlg Social Network Analysis Using Sta
nwcommands-ext Social Network Analysis Using Sta
checking nwcommands-dlg consistency and verifying not a
installing into /Users/thomasgrund/Library/Application
installation complete.
Data
Graphics
Statistics
Network Analysis
Generate Network
Example Networks
Declare Network Data
Open Networks
Save Networks As...
Import Networks
Export Networks
Convert To/From Edgelist
Network Manipulation
Network Utilities
Summarize Networks
Paths Between Nodes
Node-Level Characteristics
Generate Context Variable
Tabulate Networks
Correlate Networks
Quadratic Assignment Procedure
Exponential Random Graph Model
Visualize Networks
Help NWCOMMANDS

Nuffield Network 2008



MANCHESTER UTD – TOTTENHAM

9/9/2006, Old Trafford

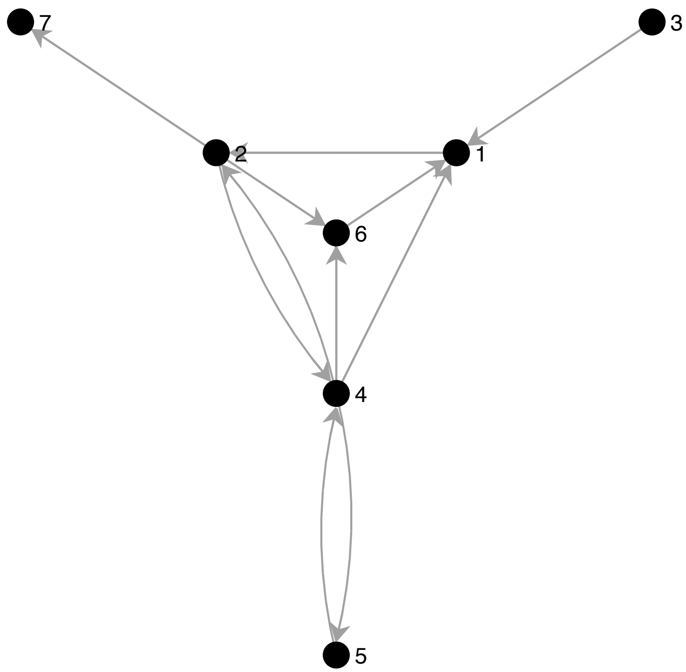


SOCIAL NETWORKS

- **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

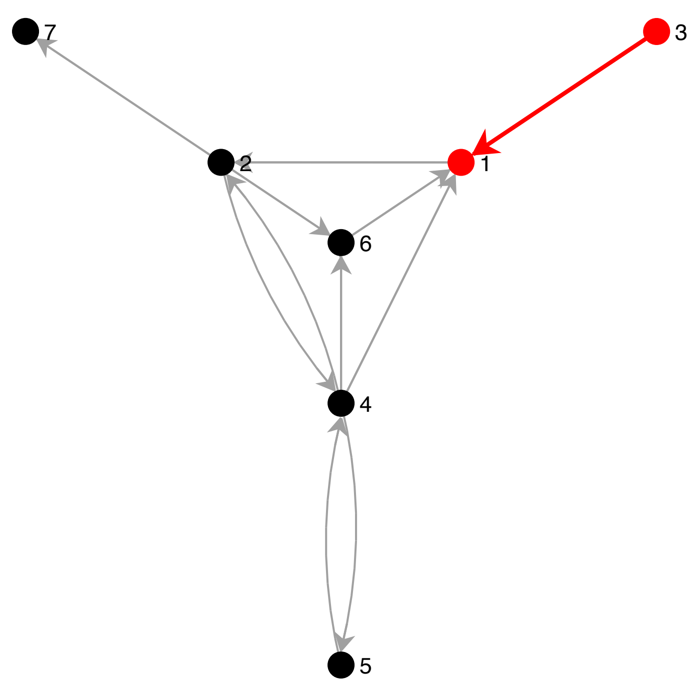
NETWORK DATA

ADJACENCY MATRIX



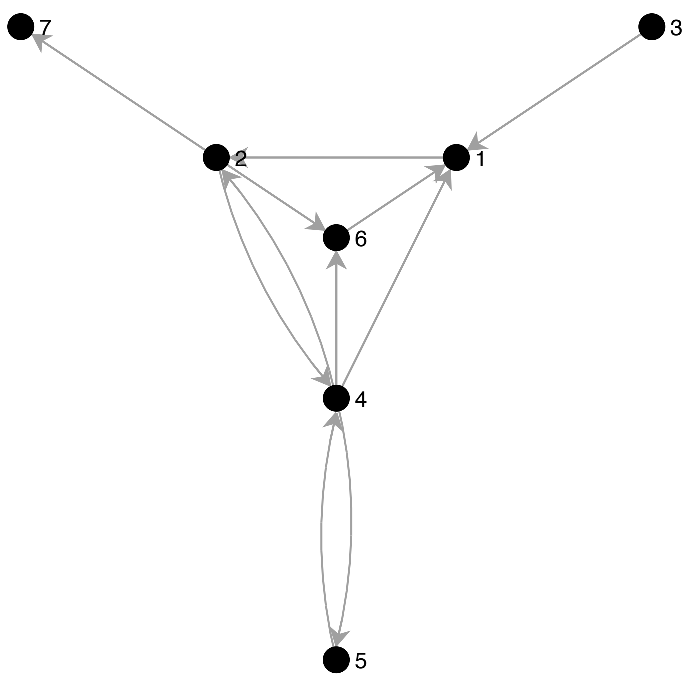
	1	2	3	4	5	6	7
1	0	1	0	0	0	0	0
2	0	0	0	1	0	1	1
3	1	0	0	0	0	0	0
4	1	1	0	0	1	1	0
5	0	0	0	1	0	0	0
6	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0

ADJACENCY MATRIX



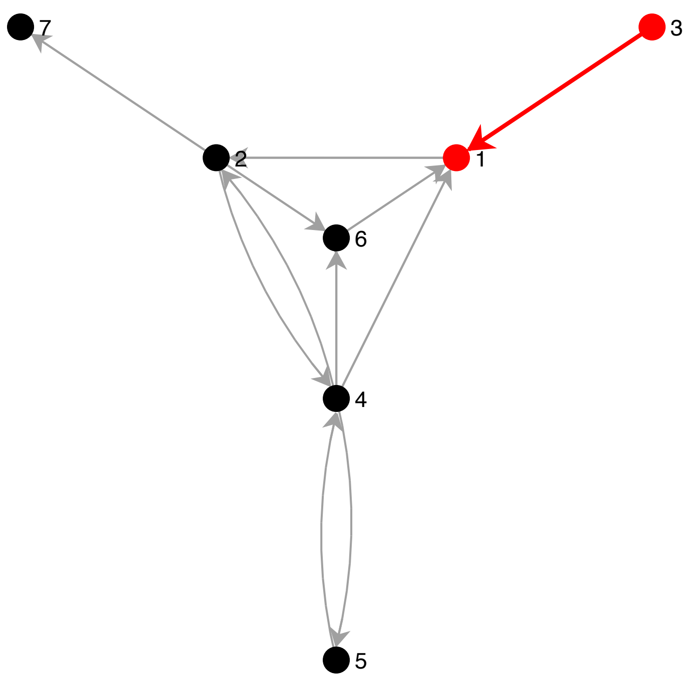
	1	2	3	4	5	6	7
1	0	1	0	0	0	0	0
2	0	0	0	1	0	1	1
3	1	0	0	0	0	0	0
4	1	1	0	0	1	1	0
5	0	0	0	1	0	0	0
6	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0

ADJACENCY LIST



	ego	alter
1	1	2
2	2	4
3	2	6
4	2	7
5	3	1
6	4	1
7	4	2
8	4	5
9	4	6
10	5	4
11	6	1

ADJACENCY LIST



	ego	alter
1	1	2
2	2	4
3	2	6
4	2	7
5	3	1
6	4	1
7	4	2
8	4	5
9	4	6
10	5	4
11	6	1

OVERVIEW

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

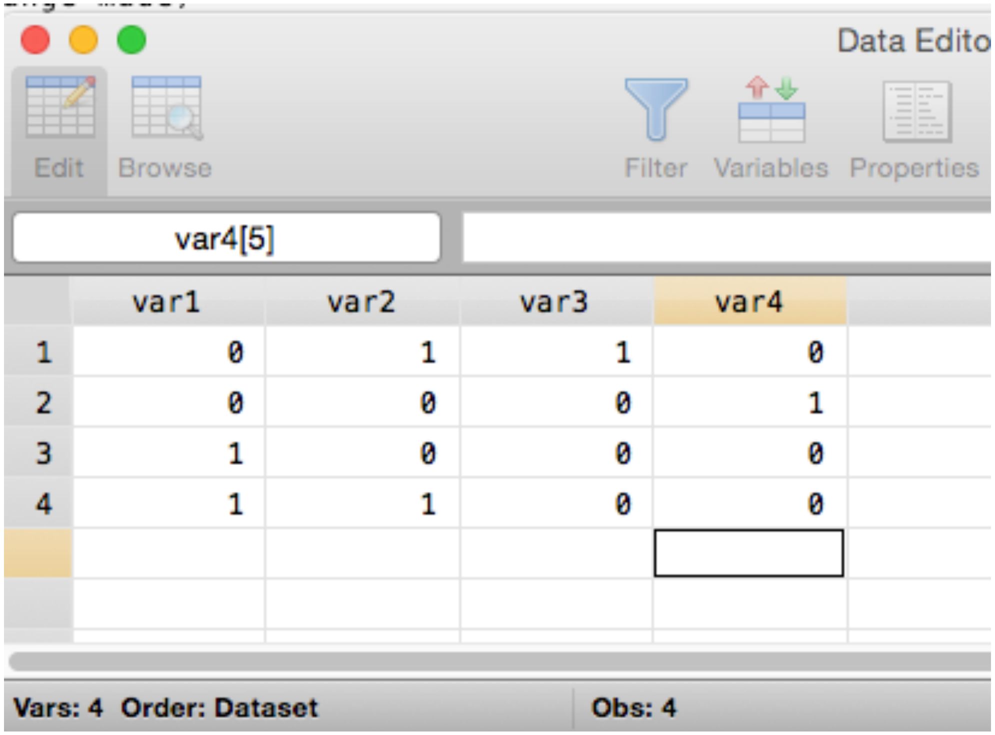
SETTING NETWORKS

- "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***.

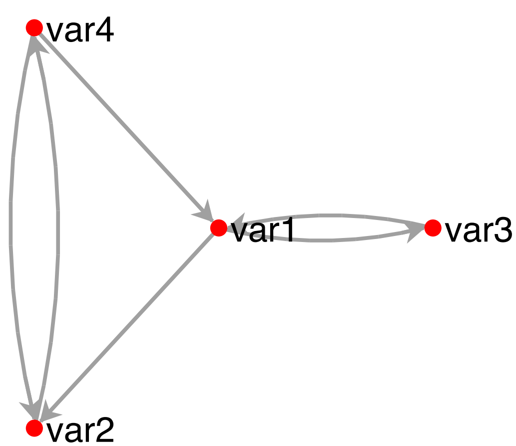
Syntax:

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

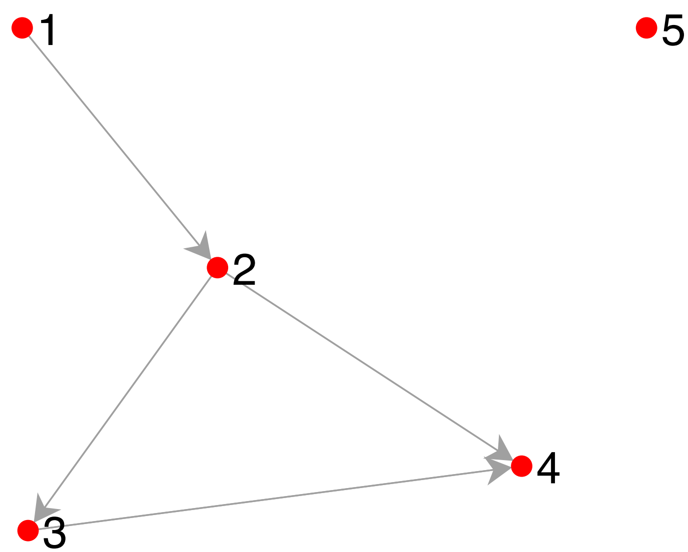
nwset, mat(*matamatrix***)** [**directed undirected name(***newnetname***) labs(***string***)** **labsfromvar(***varname***) vars(***string***) xvars**]



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




```
. nwset _all

. nwplot, lab
```



ego[1] 1

	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.

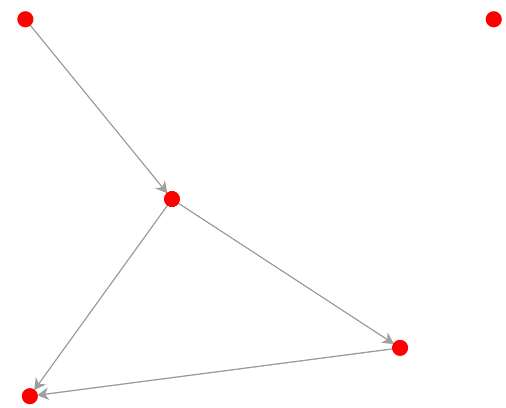
```
. nwset, detail
(2 networks)

--------------------------------------------------
 1) Stored Network
--------------------------------------------------
   Network name: network
   Directed: true
   Nodes: 4
   Network id: 1
   Variables: var1 var2 var3 var4
   Labels: var1 var2 var3 var4
   Edgelabels:

--------------------------------------------------
 2) Current Network
--------------------------------------------------
   Network name: network_1
   Directed: true
   Nodes: 5
   Network id: 2
   Variables: net1 net2 net3 net4 net5
   Labels: 1 2 3 4 5
   Edgelabels:
```

CURRENT NETWORK

- Many nwcommands ask for a ***netname***.
- When a command allows for a ***netname*** to be optional, you do not have to provide a network name and can just leave it blank.
- In this case, the command automatically applies to the *current network*.

```
. nwds
network     network_1

. nwplot
```

```
. nwplot network_1
```



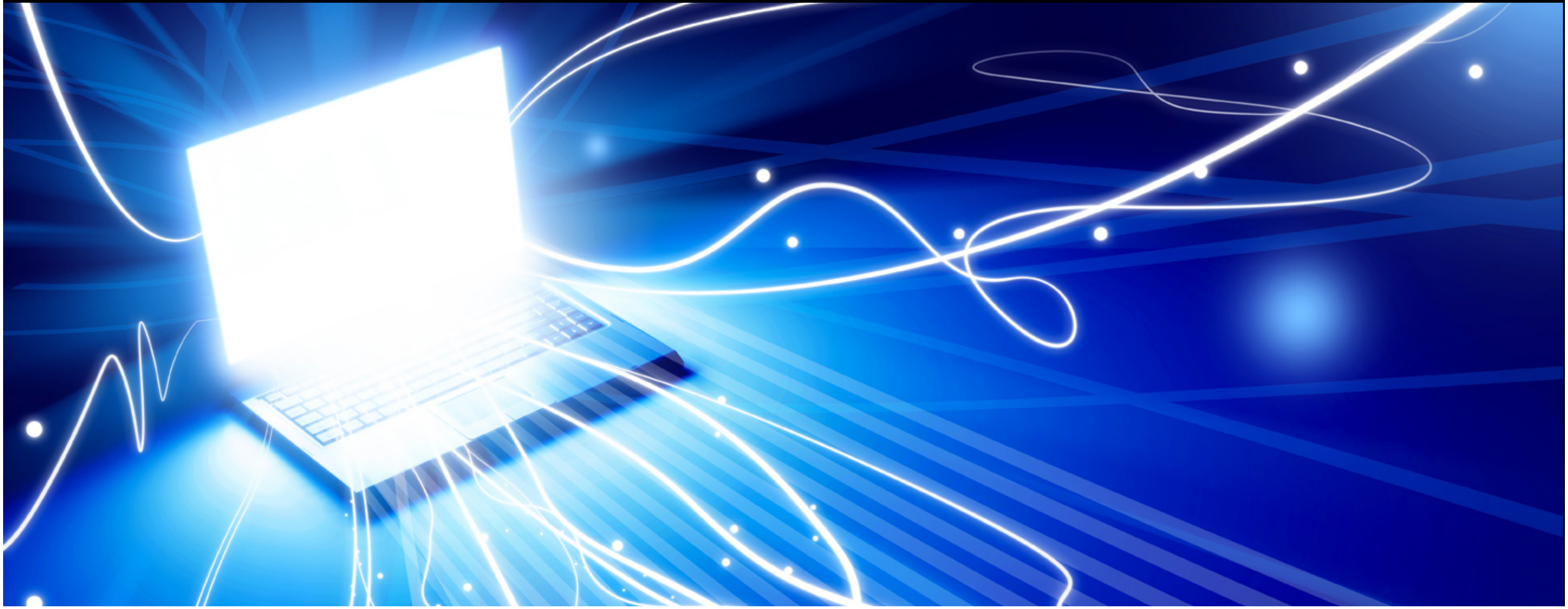
```
. nwset, detail
(2 networks)

-------------------------------------------
 1) Stored Network
-------------------------------------------
   Network name: network
   Directed: true
   Nodes: 4
   Network id: 1
   Variables: var1 var2 var3 var4
   Labels: var1 var2 var3 var4
   Edgelabels:

-------------------------------------------
 2) Current Network
-------------------------------------------
   Network name: network_1
   Directed: true
   Nodes: 5
   Network id: 2
   Variables: net1 net2 net3 net4 net5
   Labels: 1 2 3 4 5
   Edgelabels:
```

Simply the last network that you "set" or generated

OVERVIEW

```
nwset
nwds
nwcurrent
```

DATA MANAGEMENT

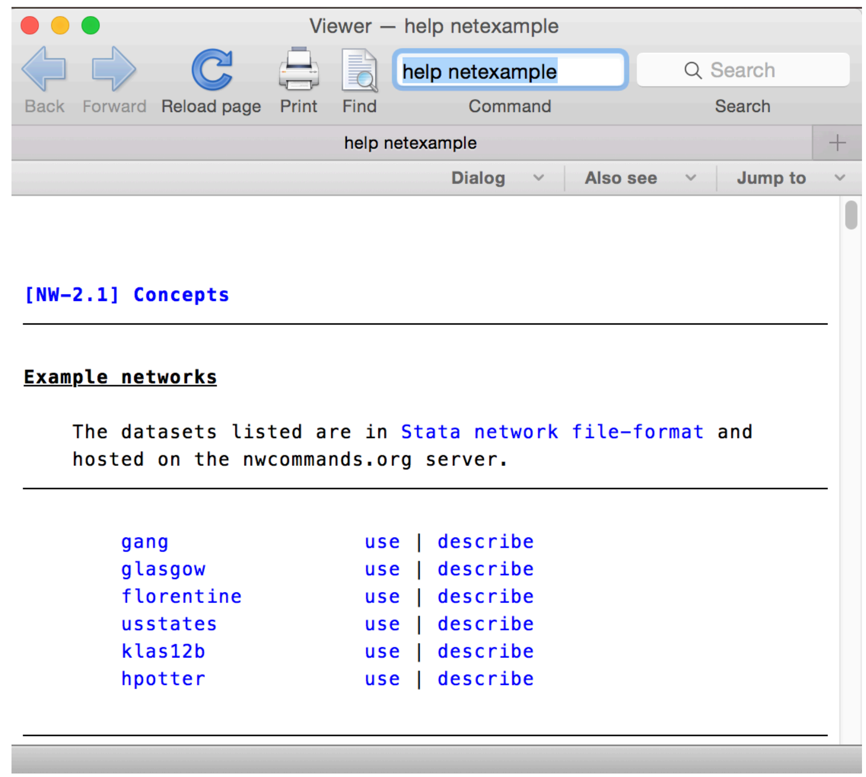


LOAD NETWORK FROM THE INTERNET

```
. 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

NODE ATTRIBUTES

NODE ATTRIBUTES

```
. webnwuse florentine, nwclear
```

	wealth	priorates	seat	_nodelab	_nodevar	_nodeid
1	10	53	1	acciaiuoli	acciaiuoli	1
2	36	65	1	albizzi	albizzi	2
3	55	0	0	barbadori	barbadori	3
4	44	12	1	bischeri	bischeri	4
5	20	22	1	castellani	castellani	5
6	32	0	0	ginori	ginori	6
7	8	21	1	guadagni	guadagni	7
8	42	0	0	lamberteschi	lamberteschi	8

- Every node of a network has a ***nodeid***, which is matched with the observation number in a normal dataset.
- In this case, the node with ***nodeid*** == 1 is the “acciaiuoli” family and they have a wealth of 10.

NODE ATTRIBUTES

```
. webnwuse florentine, nwclear
```

	wealth	priorates	seat	_nodelab	_nodevar	_nodeid
1	10	53	1	acciaiuoli	acciaiuoli	1
2	36	65	1	albizzi	albizzi	2
3	55	0	0	barbadori	barbadori	3
4	44	12	1	bischeri	bischeri	4
5	20	22	1	castellani	castellani	5
6	32	0	0	ginori	ginori	6
7	8	21	1	guadagni	guadagni	7
8	42	0	0	lamberteschi	lamberteschi	8

- Every node of a network has a ***nodeid***, which is matched with the observation number in a normal dataset.
- In this case, the node with ***nodeid*** == 1 is the “acciaiuoli” family and they have a wealth of 10.

DROP/KEEP NODES

- When you drop/keep nodes, by default, attributes are not included in the change. But with the option **`attributes()`** you can include attribute variables in the drop/keep.

```
. nwdrop flomarriage if _nodelab == "albizzi", attributes(wealth priorates seat)
```

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
```

```
. nwsummarize glasgow1, detail
------------------------------------------------------------------------
   Network name:  glasgow1
   Network id:  1
   Directed: true
   Nodes: 50
   Arcs: 113
   Minimum value:  0
   Maximum value:  1
   Density:  .0461224489795918
   Reciprocity:  .52702702702702 7
   Transitivity:  .3870967741935484
   Betweenness centralization:  .0821793002915452
   Indegree centralization:: .119533527696793
   Outdegree centralization:: .0570595585172845
```

OBTAIN TIE VALUES

```
. nwsummarize network_1, matonly
        1    2    3    4    5

  1     0    1    0    0    0
  2     0    0    1    1    0
  3     0    0    0    1    0
  4     0    0    0    0    0
  5     0    0    0    0    1
```

OBTAIN TIE VALUES

```
. nwvalue network_1[2,3]
  1
```

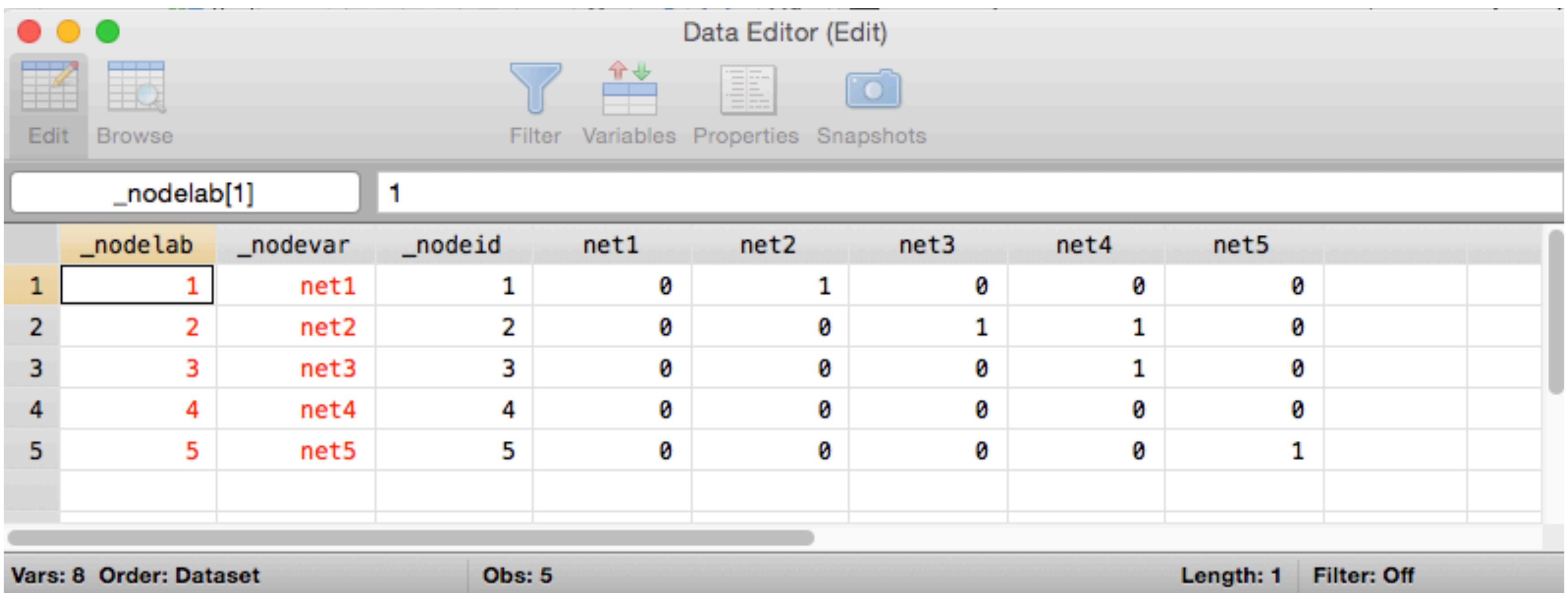
```
. nwvalue network_1[(1::3),(1::3)]
```

$$\begin{array}{c|ccc|}
 & 1 & 2 & 3 \\
\hline
1 & 0 & 1 & 0 \\
2 & 0 & 0 & 1 \\
3 & 0 & 0 & 0 \\
\hline
\end{array}$$

Example	Description
mynet	The whole network.
mynet[2,3]	Specific tie value; toe that node 3 received from node 2.
mynet[(2::4),3]	All ties that node 3 receives from nodes 2 to 4.
mynet[(2::4,(3::4)]	All ties that nodes 3 to 4 receive from nodes 2 to 4.
mynet[|(2,3)\\(4,4)|]	All ties that nodes 2 to 4 send to nodes 3 to 4.

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
```

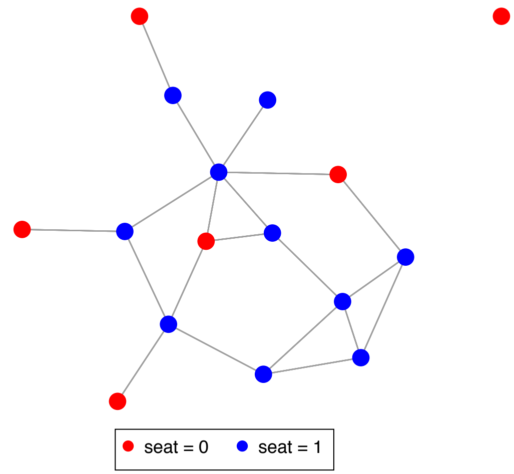
flomarriage	flobusiness 0	1	Total
0	93	7	100
1	12	8	20
Total	105	15	120

TABULATE NETWORK AND ATTRIBUTE

```
. nwtabulate flomarriage seat
(0 observations deleted)

   Network:   flomarriage       Directed: false
   Attribute:   seat

       The network is undirected.
       The table shows two entries for each edge.
```

from_seat	to_seat 0	1	Total
0	0	8	8
1	8	24	32
Total	8	32	40

```
   E-I Index: -.2    p-value: .22
```



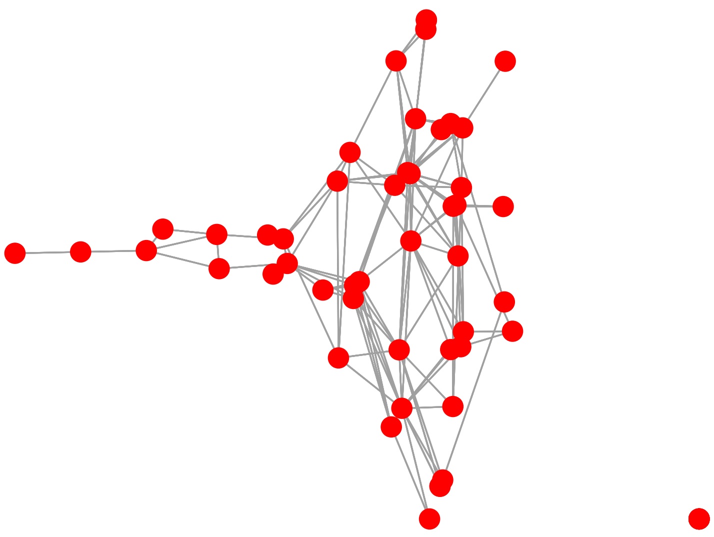
CHANGE NETWORK

GANG NETWORK

```
. webnwuse gang, nwclear
```

Loading successful
(2 networks)

```
gang_valued
gang
```



TABULATE NETWORK

```
. 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	

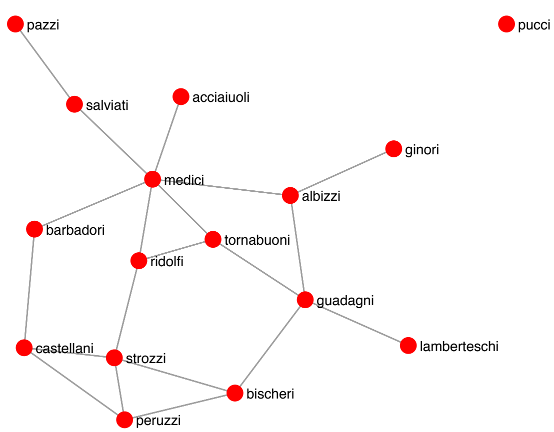
RECODE TIE VALUES

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

(gang_valued: 266 changes made)

. nwtabulate gang_valued

   Network:  gang_valued      Directed: false
```

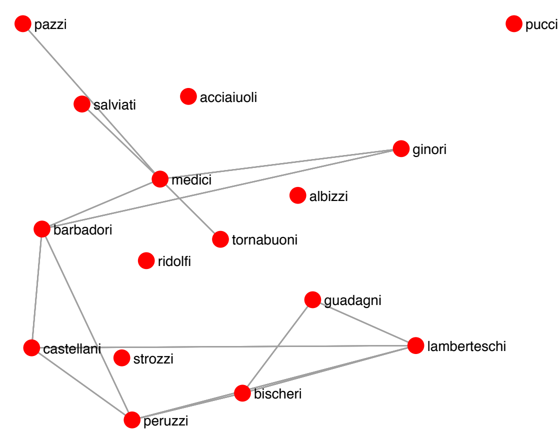
gang_valued	Freq.	Percent	Cum.
0	1,116	77.99	77.99
1	182	12.72	90.71
99	133	9.29	100.00
Total	1,431	100.00	

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	

Viewer — help nwreplace

Back Forward Reload page Print Find

help nwreplace

Command Search

help netexample | help nwdrop | help nwsort | help nwreplace | help nwreplace

Dialog Also see Jump to

[NW-2.5] Manipulation

Title

nwreplace — Replace network

Syntax

nwreplace *netname[subnet]* =*netexp* [*ifego*] [*ifalter*] [*if*] [*in*]

Description

Replaces whole networks, subnetworks or specific dyads. Similar in usage to replace. A network expression is very similar to a normal expression in Stata, but also accepts netnames.

One can also replace dyads in networks by 1) loading a network as Stata variables (see nwload), 2) changing the Stata variables (see replace) and 3) syncing Stata variables and network afterwards (see nwsync). However, replacing the

```
. help nwreplace
```

GENERATE NETWORKS

```
. nwgen both = (flobusiness & flomarriage)

. nwtabulate both

   Network:  both      Directed: false
```

both	Freq.	Percent	Cum.
0	112	93.33	93.33
1	8	6.67	100.00
Total	120	100.00	

Viewer — help nwgen

Back Forward Reload page Print Find

help nwgen

Command

help nwgen

[NW-2.6] Analysis

Title

nwgen —— Network extensions to generate

Syntax

nwgen *newvar* = *netfcn1*(*arguments*) [, *options*]

nwgen *newnetname* = *netfcn2*(*arguments*) [, *options*]

nwgen *newnetname* = *netexp* [if] [, *options*]

where the *options* are also *fcn* dependent.

Description

These are network extensions to generate. The command is very similar to egen and allows producing either variables or networks. There are basically three ways to use this commands: 1) produce Stata variables with some function *netfcn1*, 2) produce networks with some function *netfnc2*, 3) produce networks with an expression *netexp*. A network expression is very similar to normal expressions in Stata.

```
. help nwgen
```

DYADS AND TRIADS

DYAD

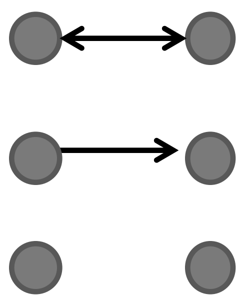
A dyad is a pair of actors (i, j) in the network, plus the configuration of the tie variables (y_{ij}, y_{ji}) between them.

- In a directed, binary network, there are $n(n-1)$ tie variables located in $n(n-1)/2$ dyads.
- Dyads can be of three types:

M: mutual

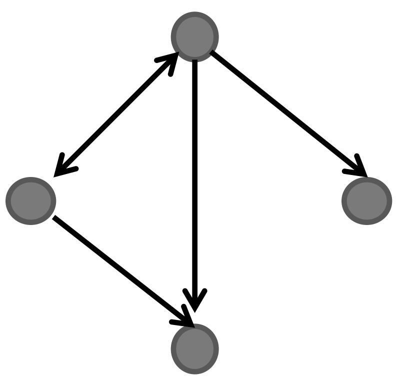
A: asymmetric

N: null

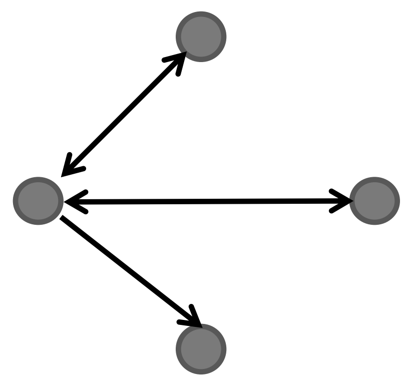


DYAD CENSUS

We can describe a network by counting the number of **mutual, asymmetric** and **null** dyads. It is like taking a "fingerprint" of a network.



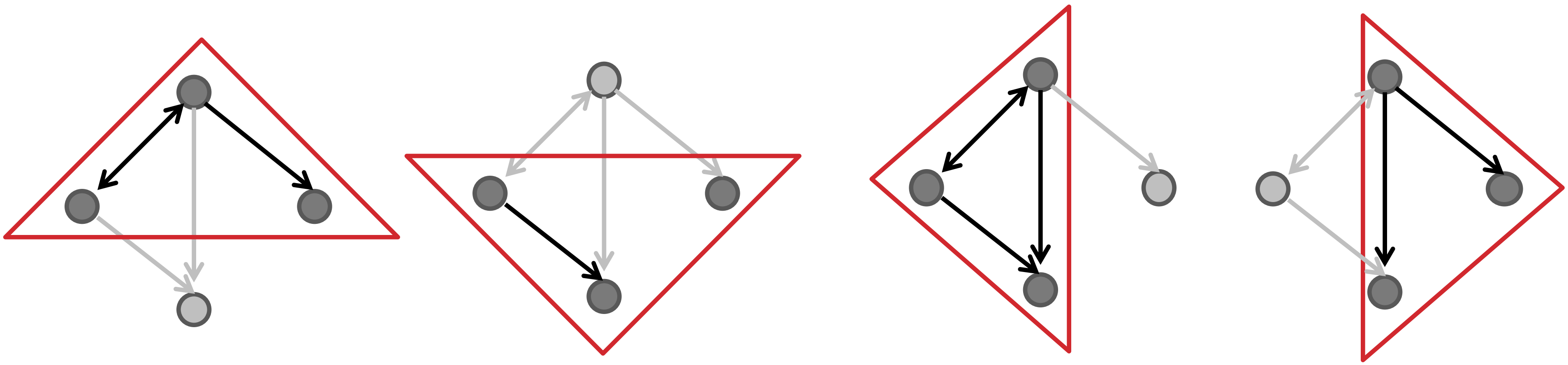
MAN = 132

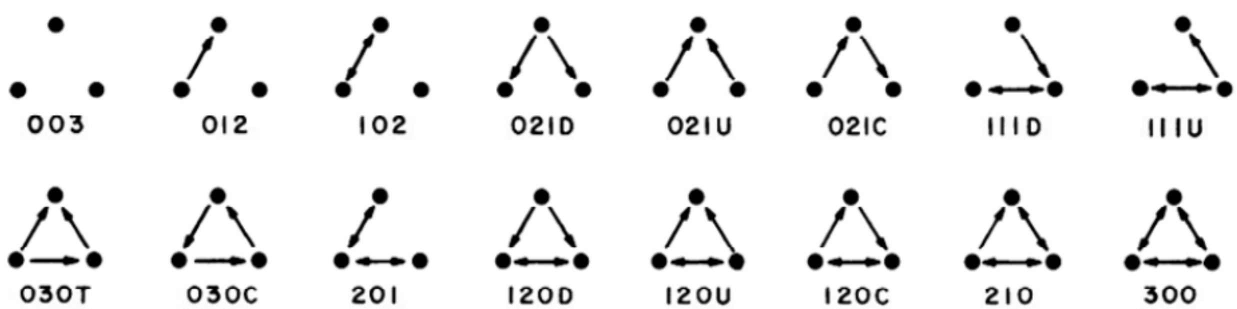


MAN = 213

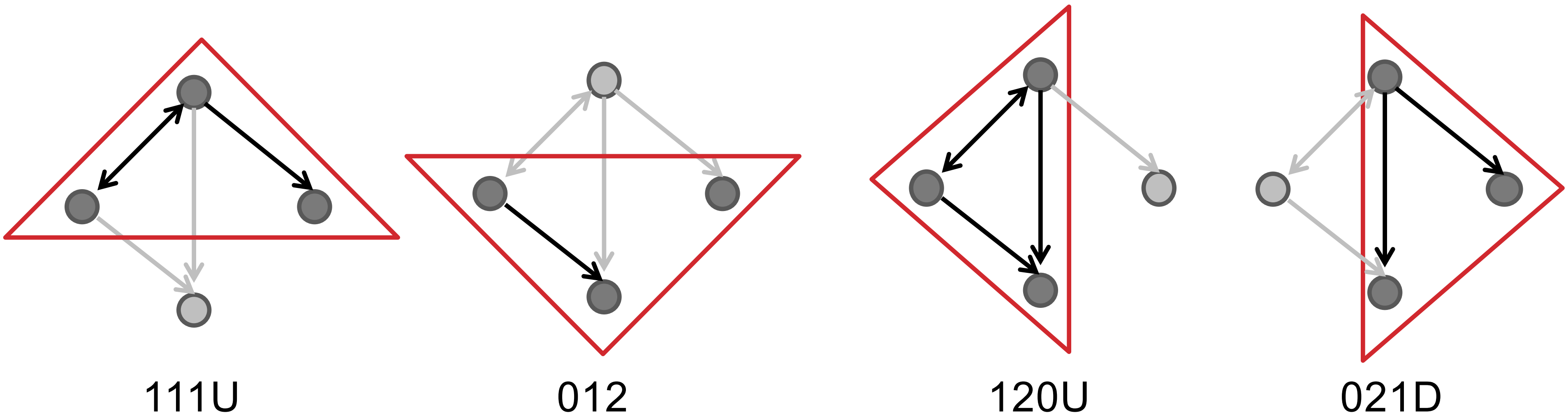
```
. nwuse glasgow

Loading successful
(3 networks)
______________________________
        glasgow1
        glasgow2
        glasgow3

. nwdyads glasgow1

     Dyad census:  glasgow1

     Mutual |        Asym |          Null
------------+-------------+--------------
         39 |          35 |          1151

     Reciprocity: .527027027027027
```

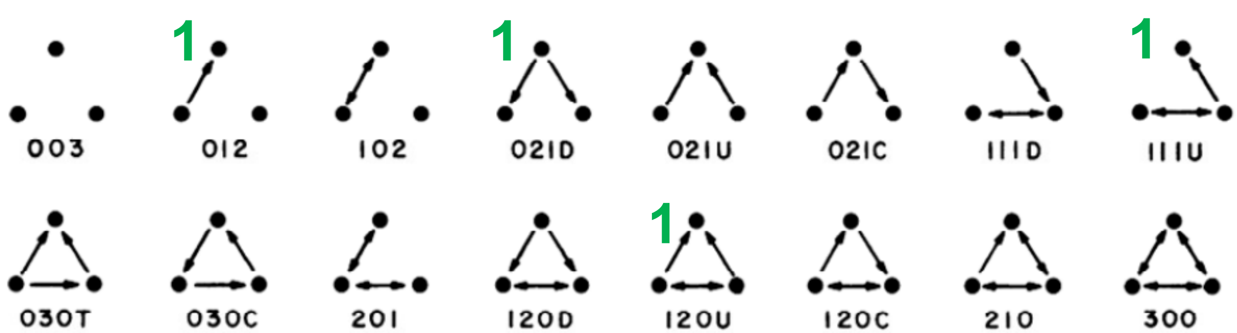
TRIAD CENSUS





TRIAD CENSUS





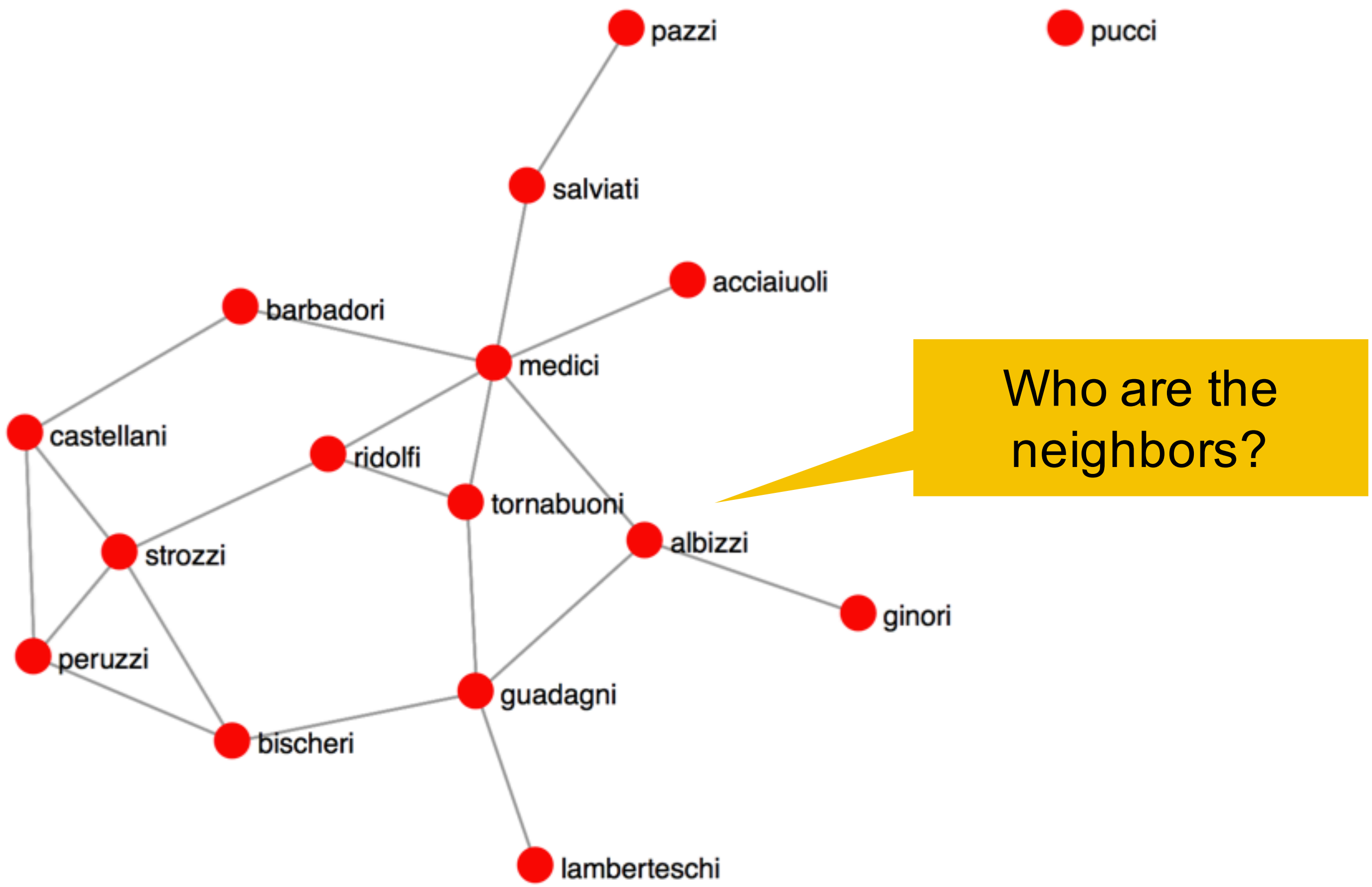
```
. nwtriads glasgow1

    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
```

NEIGHBORS AND CONTEXT

FLORENTINE FAMILIES



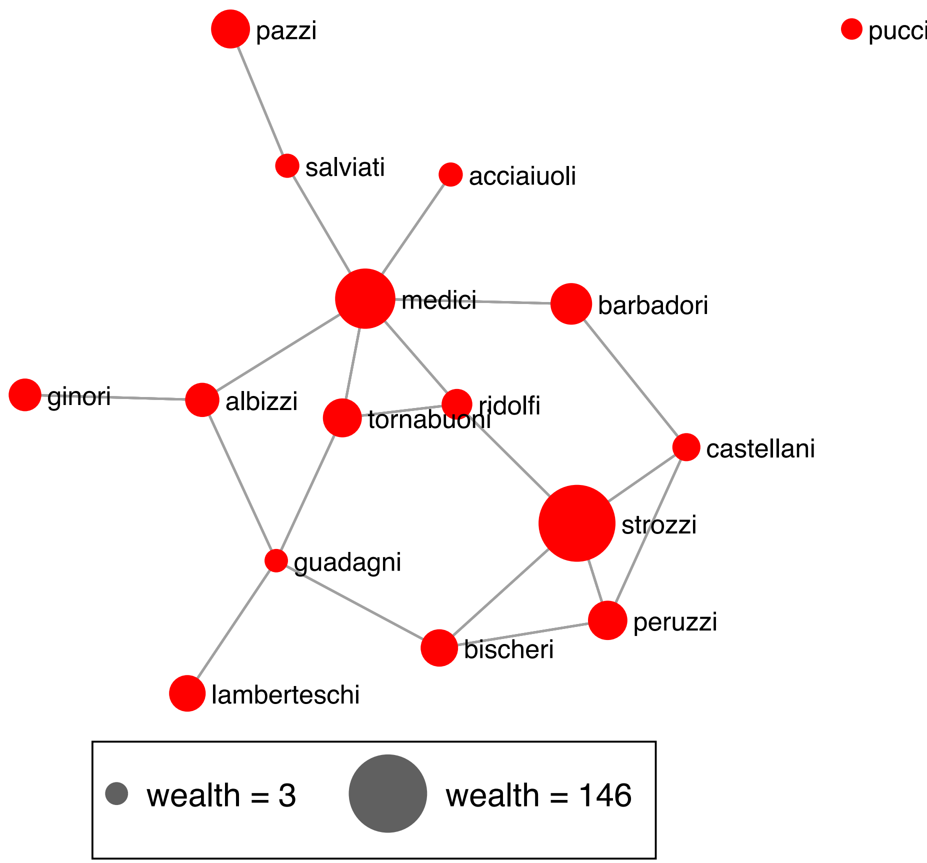
NEIGHBORS

```
. webnwuse florentine, nwclear

. nwneighbor flomarriage, ego(albizzi)

--------------------------------------------------
  Network: flomarriage
--------------------------------------------------
    Ego          : albizzi
    Neighbors    : ginori , guadagni , medici
--------------------------------------------------
```

NEIGHBORS

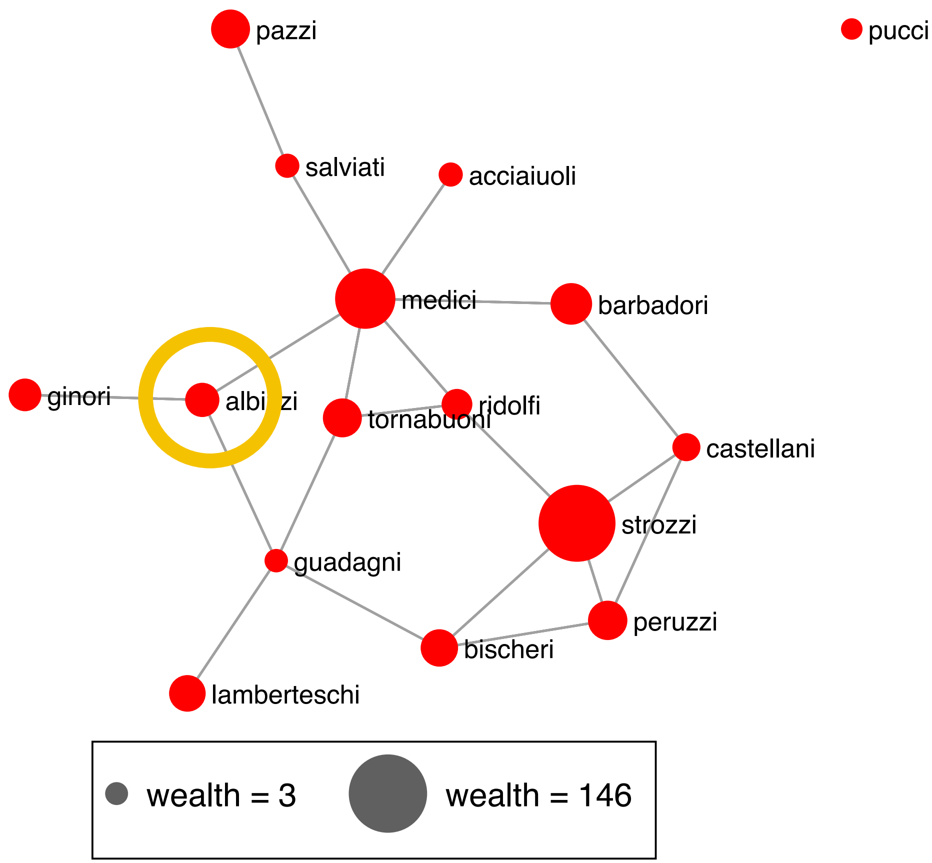
```
. return list

scalars:
                  r(ego) =  2
          r(oneneighbor) =  6

macros:
     r(neighbors_list2) : " ginori guadagni medici"
     r(neighbors_list1) : " 6 7 9"

matrices:
            r(neighbors) :  3 x 1
```

CONTEXT



pazzi
pucci
salviati
acciaiuoli
medici
barbadori
ginori
albizzi
tornabuoni
ridolfi
castellani
strozzi
guadagni
peruzzi
bischeri
lamberteschi
wealth = 3
wealth = 146

CONTEXT



What is the average wealth of the “albizzi’s” network neighbors?

CONTEXT

```
. nwcontext flomarriage, attribute(wealth) stat(mean) generate(wmean)

. nwcontext flomarriage, attribute(wealth) stat(max) generate(wmax)

. nwcontext flomarriage, attribute(wealth) stat(min) generate(wmin)

. list _nodelab w*
```

	_nodelab	wealth	wmean	wmax	wmin
1.	acciaiuoli	10	103	103	103
2.	albizzi	36	47.66667	103	8
3.	barbadori	55	61.5	103	20
4.	bischeri	44	67.66666	146	8
5.	castellani	20	83.33334	146	49

CONTEXT

statistic	**Description**
`mean`	Mean of *varname* over network neighbors; defaul.
`max`	Maximum of *varname* over network neighbors.
`min`	Minimum of *varname* over network neighbors.
`sum`	Sum of *varname* over network neighbors.
`sd`	Standard deviation of *varname* over network neighbors.
`meanego`	Mean of *varname* over network neighbors and *ego*.
`maxego`	Maximum of *varname* over network neighbors and *ego*.
`minego`	Minimum of *varname* over network neighbors and *ego*.
`sumego`	Sum of *varname* over network neighbors and *ego*.
`sdego`	Standard deviation of *varname* over network neighbors and *ego*.

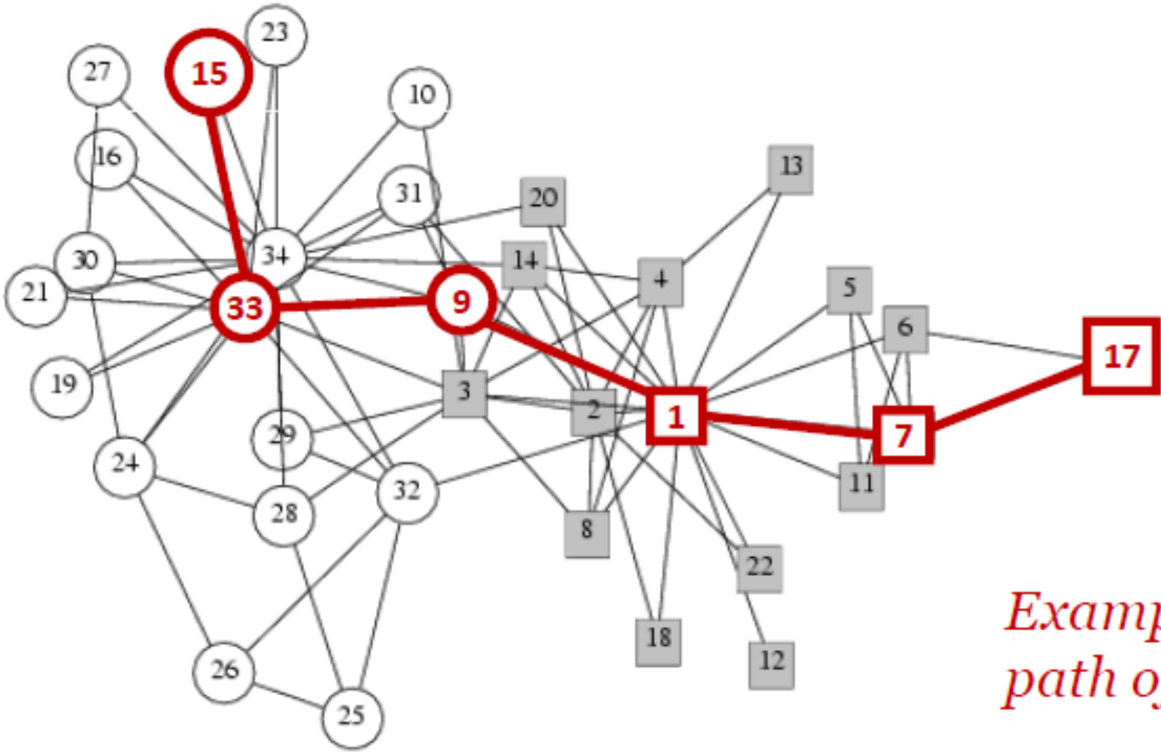
CONTEXT

context	**Description**
`outgoing`	Network neighbors of node *ego* are all nodes *alter* who receive a tie from *ego*; **`default.`**
`incoming`	Network neighbors of node *ego* are all nodes *alter* who send a tie to *ego*.
`both`	Network neighbors of node *ego* are all nodes *alter* who either receive or send a tie to/from *ego*.

DISTANCE

DISTANCE

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

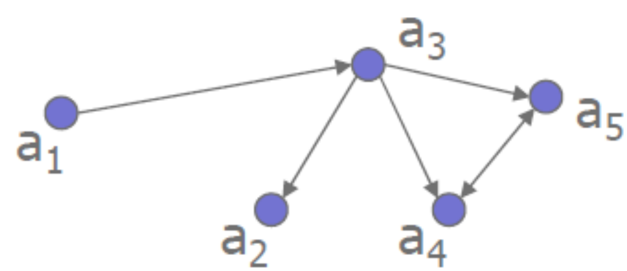


Example of a shortest path of length 5

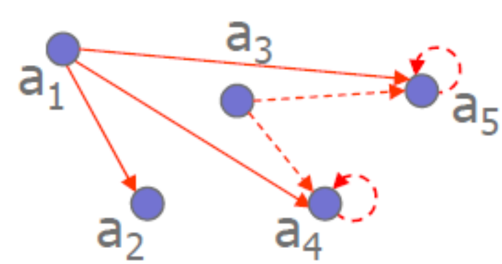
DISTANCE

How can we calculate the distance?

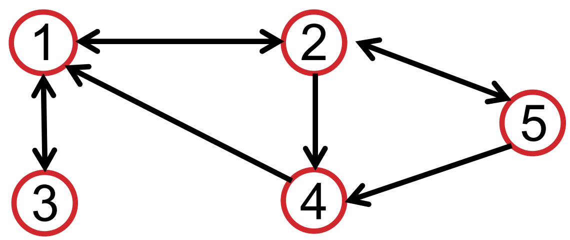
- Matrix $\boldsymbol{y}$ indicates which row actor is directly connected to which column actor.
- The squared matrix $\boldsymbol{y}^2$ indicates which row actor can reach which column actor in two steps.
- The matrix $\boldsymbol{y}^l$ indicates who reaches whom in l steps.



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



DISTANCE



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

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

DISTANCE

```
. 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
```

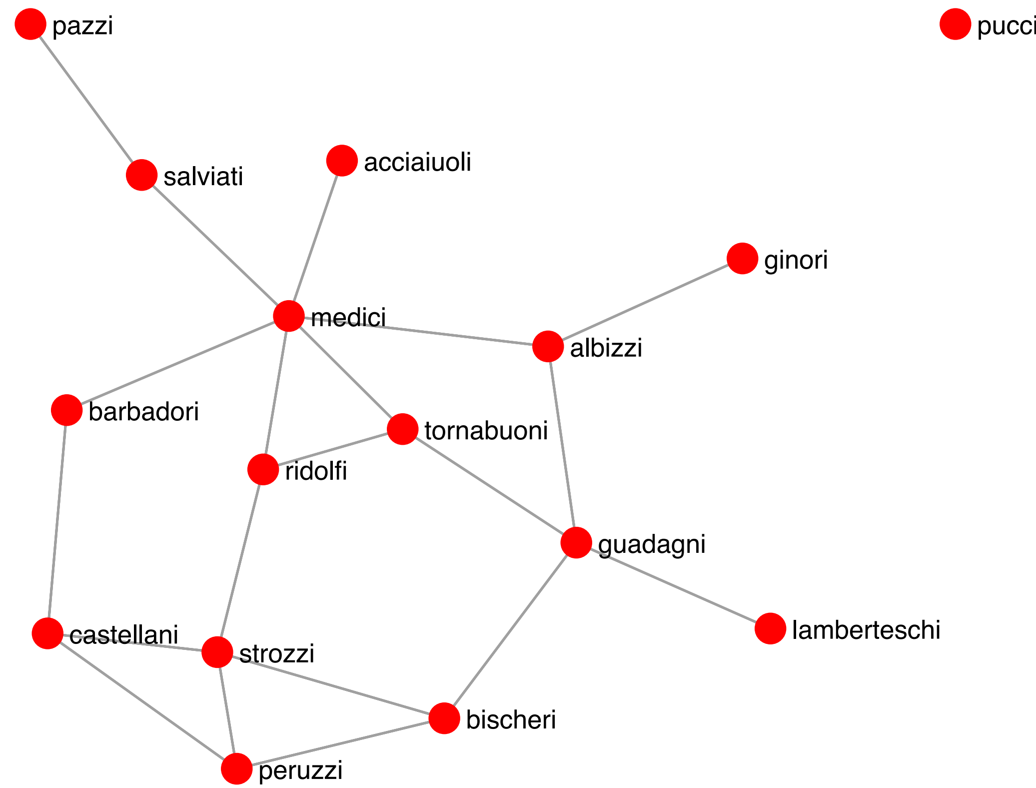
```
. nwtabulate geodesic

   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	

PATHS

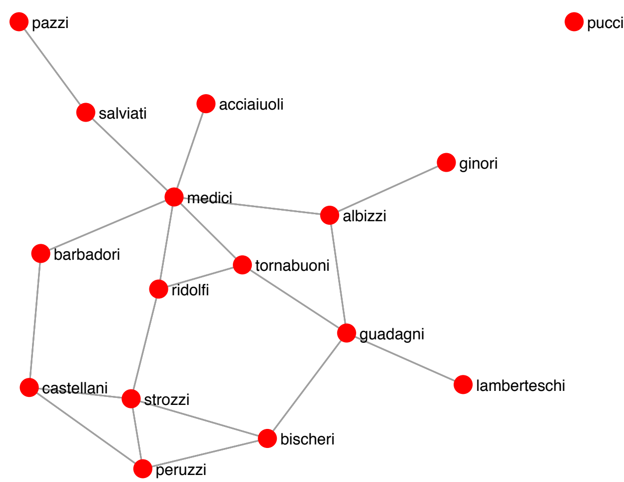
```
. webnwuse florentine, nwclear
```

How can one get from the “peruzzi” to the “medici”?



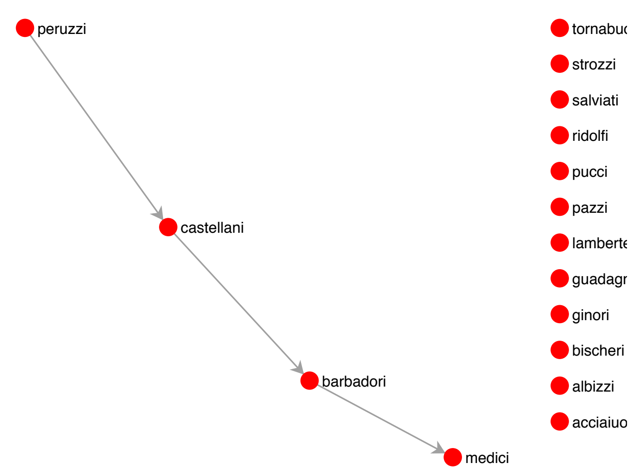
PATHS

```
. nwpath flomarriage, ego(peruzzi) alter(medici)

------------------------------------------------
  Network: flomarriage
------------------------------------------------
    Ego                    : 11 (peruzzi)
    Alter                  : 9 (medici)
    Shortest path length : 3
    Selected length        : 3
------------------------------------------------

  Path 1:  peruzzi => castellani => barbadori => medici
  Path 2:  peruzzi => strozzi => ridolfi => medici
```



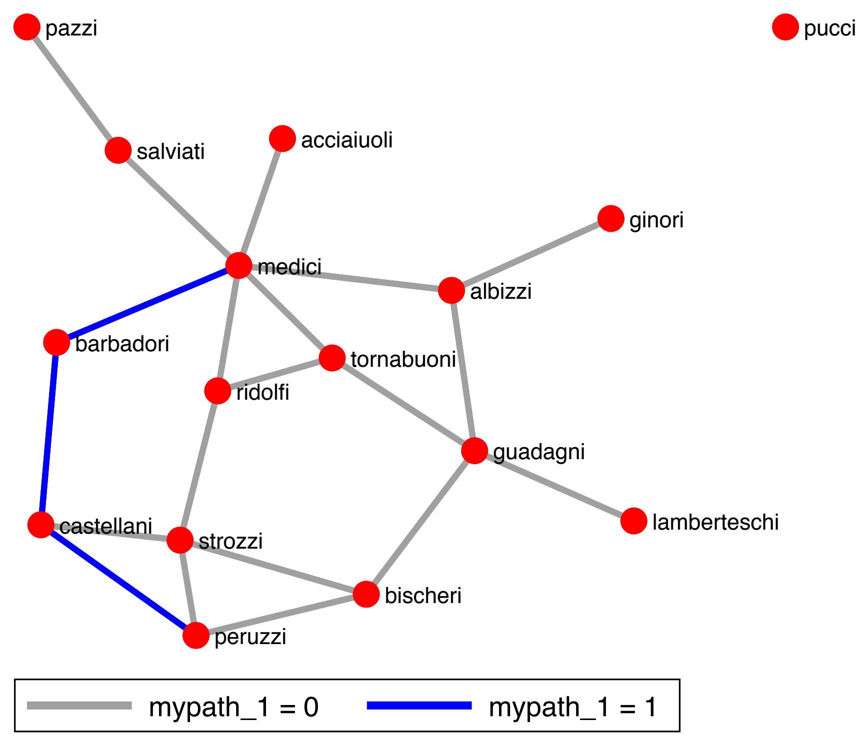
PATHS

```
. nwpath flomarriage, ego(peruzzi) alter(medici) generate(mypath)

  Network: flomarriage

     Ego                    : 11 (peruzzi)
     Alter                  : 9 (medici)
     Shortest path length : 3
     Selected length        : 3

  Path 1:   peruzzi => castellani => barbadori => medici
  Path 2:   peruzzi => strozzi => ridolfi => medici

. nwset
(4 networks)

       flobusiness
       flomarriage
       mypath_1
       mypath_2
```



PATHS

```
. nwplot flomarriage, lab edgecolor(mypath_1) edgefactor(3)
```



PATHS OF SPECIFIC LENGTH

```
. nwpath flomarriage, ego(peruzzi) alter(medici) length(4)

  Network: flomarriage

    Ego                   : 11 (peruzzi)
    Alter                 : 9 (medici)
    Shortest path length  : 3
    Selected length       : 4

  Path 1:  peruzzi => bischeri => guadagni => albizzi => medici
  Path 2:  peruzzi => bischeri => guadagni => tornabuoni => medici
  Path 3:  peruzzi => bischeri => strozzi => ridolfi => medici
  Path 4:  peruzzi => castellani => strozzi => ridolfi => medici
  Path 5:  peruzzi => strozzi => castellani => barbadori => medici
  Path 6:  peruzzi => strozzi => ridolfi => tornabuoni => medici
```

CENTRALITY

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?

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

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

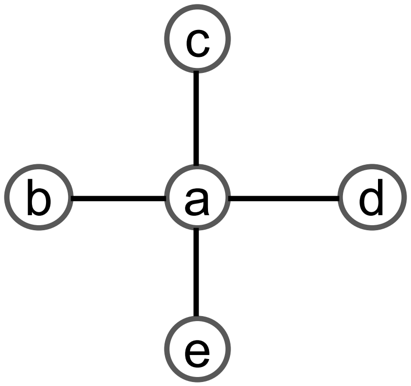
Farness

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

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

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

…



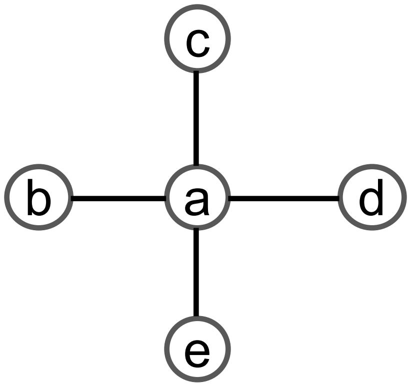
CLOSENESS CENTRALITY

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

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

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

…



BETWEENNESS CENTRALITY

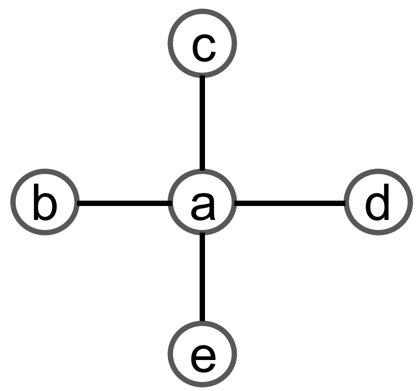
Betweeness centrality

- How many shortest paths go through an individual?

$C_{betweenness}(a) = 6$

$C_{betweenness}(b) = 0$

…

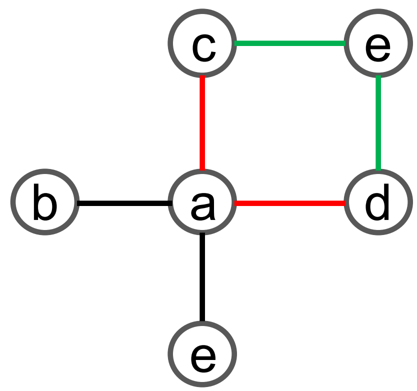


BETWEENNESS CENTRALITY

Betweeness centrality

- How many shortest paths go through an individual?

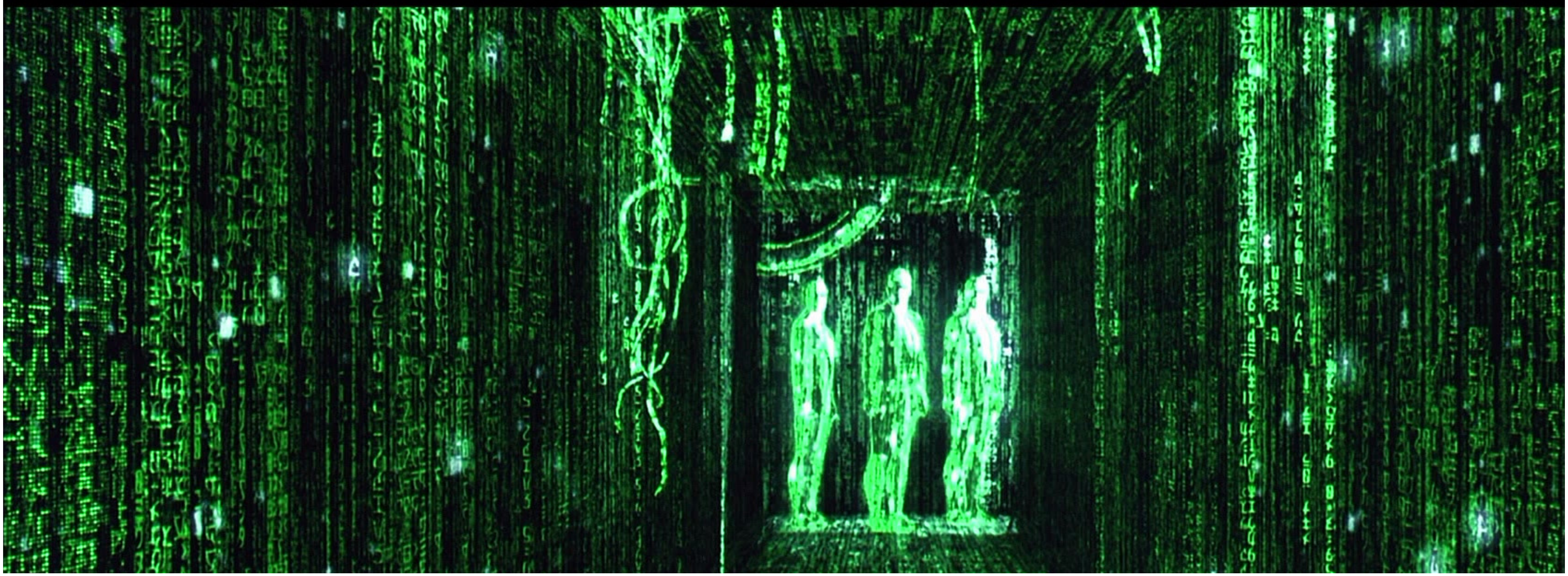
What about multiple shortest paths? E.g. there are two shortest paths from c to d (one via a and another one via e)



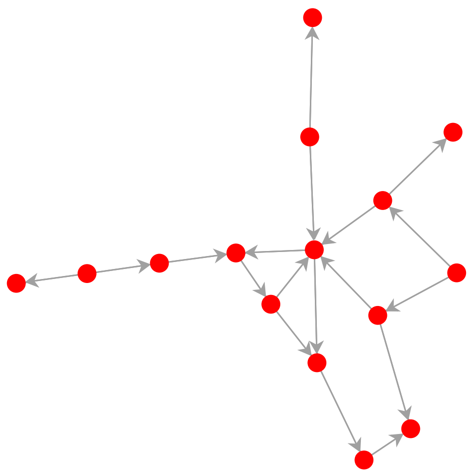
Give each shortest path a weight inverse to how many shortest paths there are between two nodes.

```
. 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 |
```

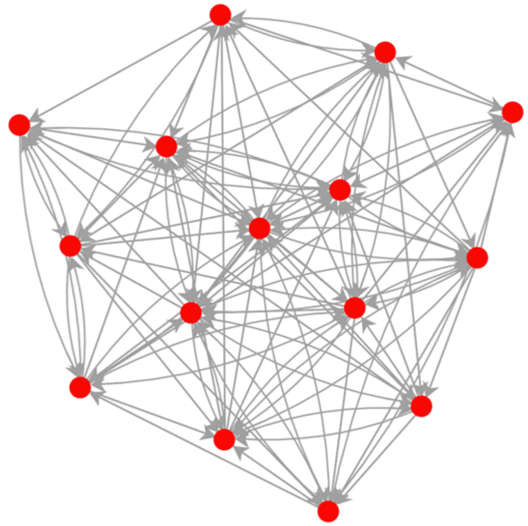
SIMULATION



RANDOM NETWORK



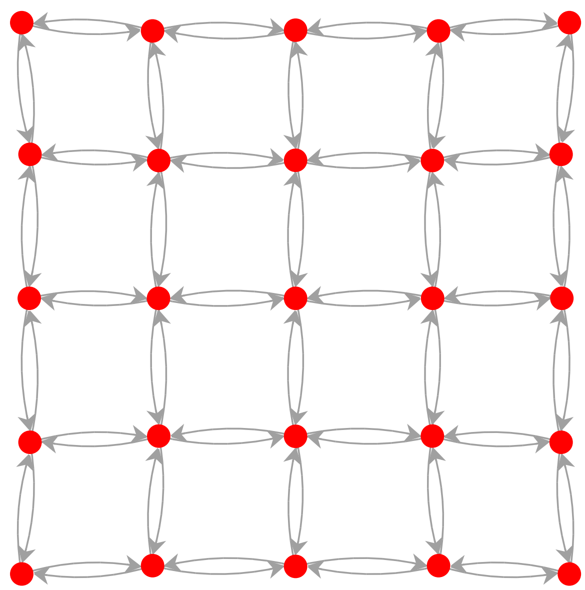
nwrandom 15, prob(.1)



nwrandom 15, prob(.5)

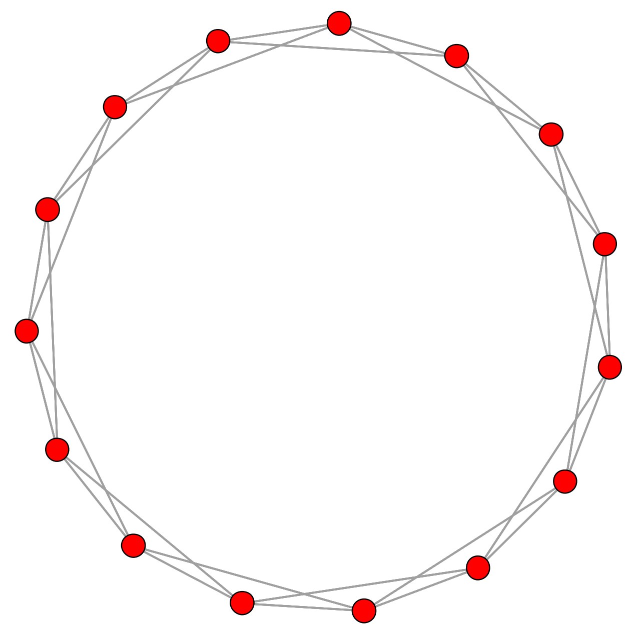
Each tie has the same probability to exist, regardless of any other ties.

LATTICE



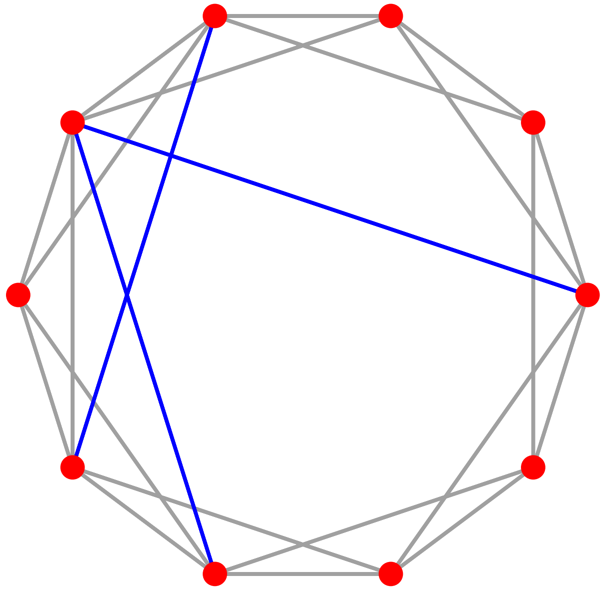
nwlattice 5 5

RING LATTICE



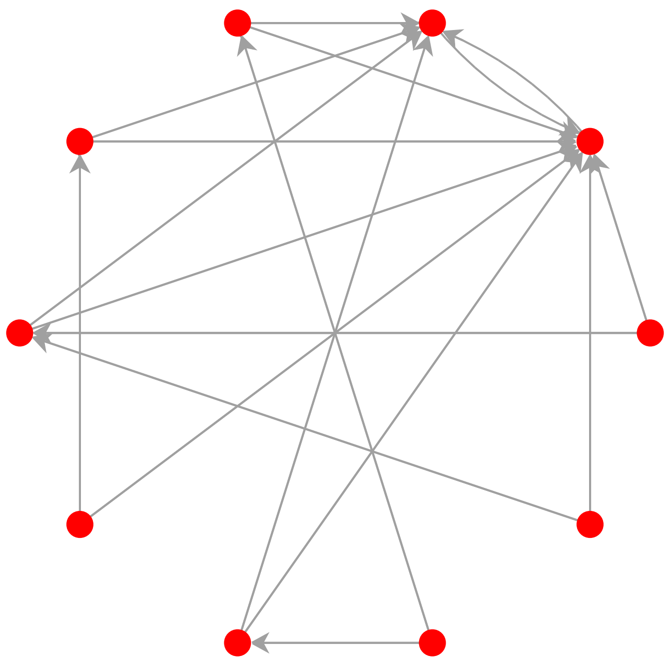
nwring 15, k(2)

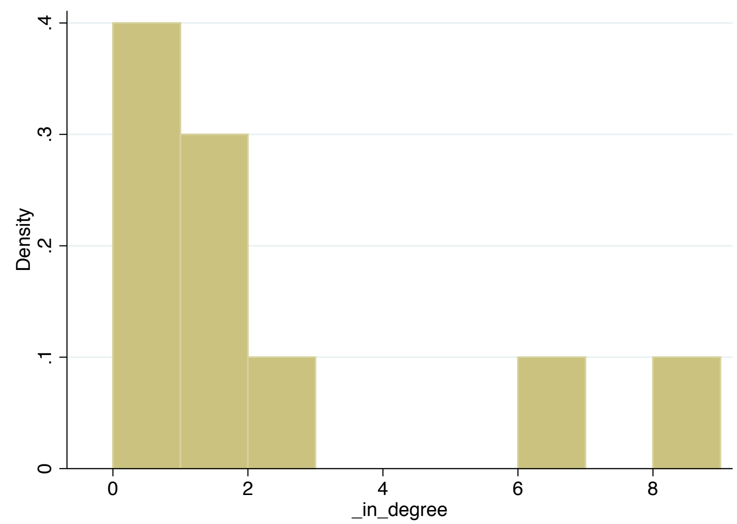
SMALL WORLD NETWORK



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

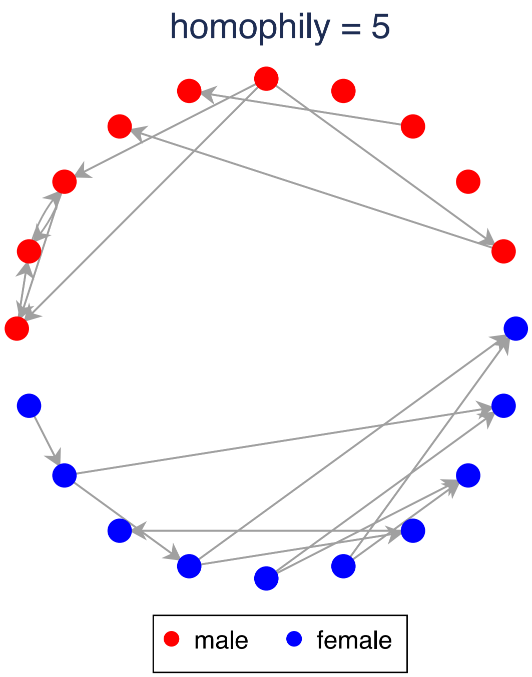
PREFERENTIAL ATTACHMENT NETWORK

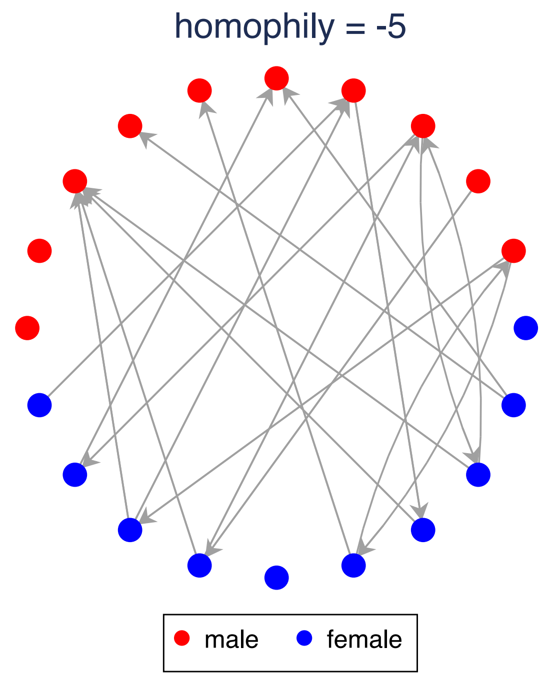





```
nwpref 10, prob(.5)
```

HOMOPHILY NETWORK

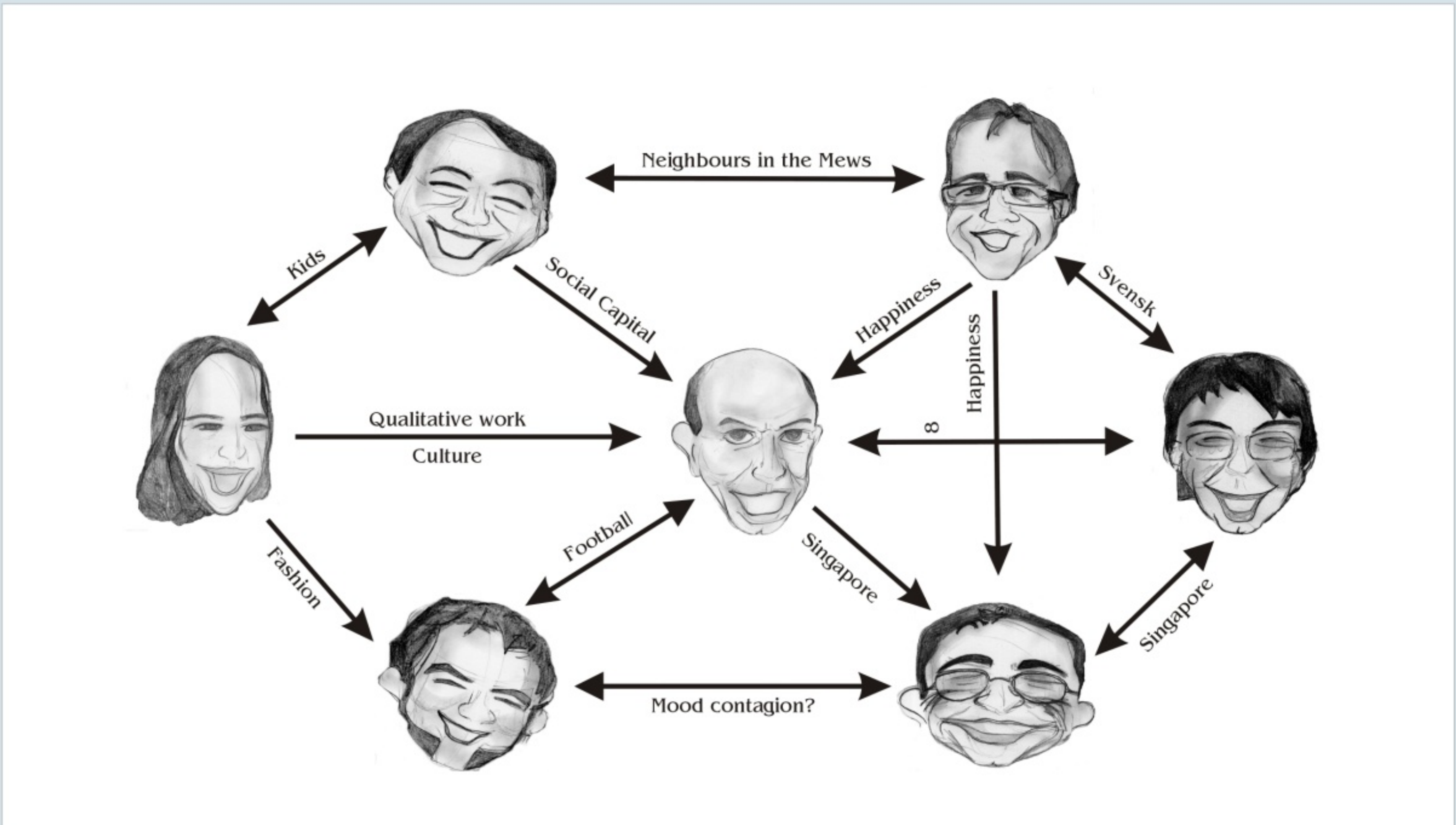


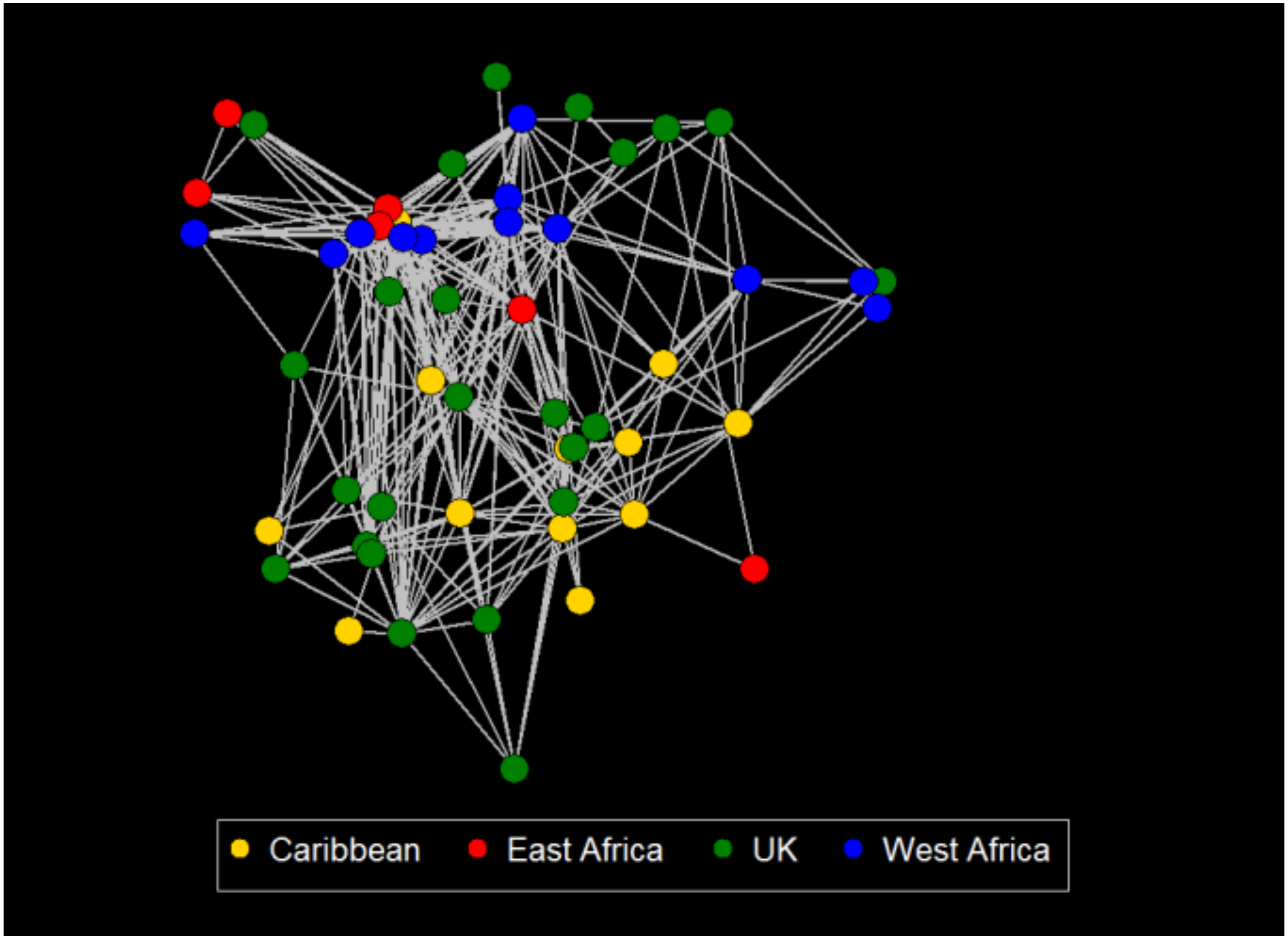



```
nwhomophily gender, density(0.05) homophily(5)
```

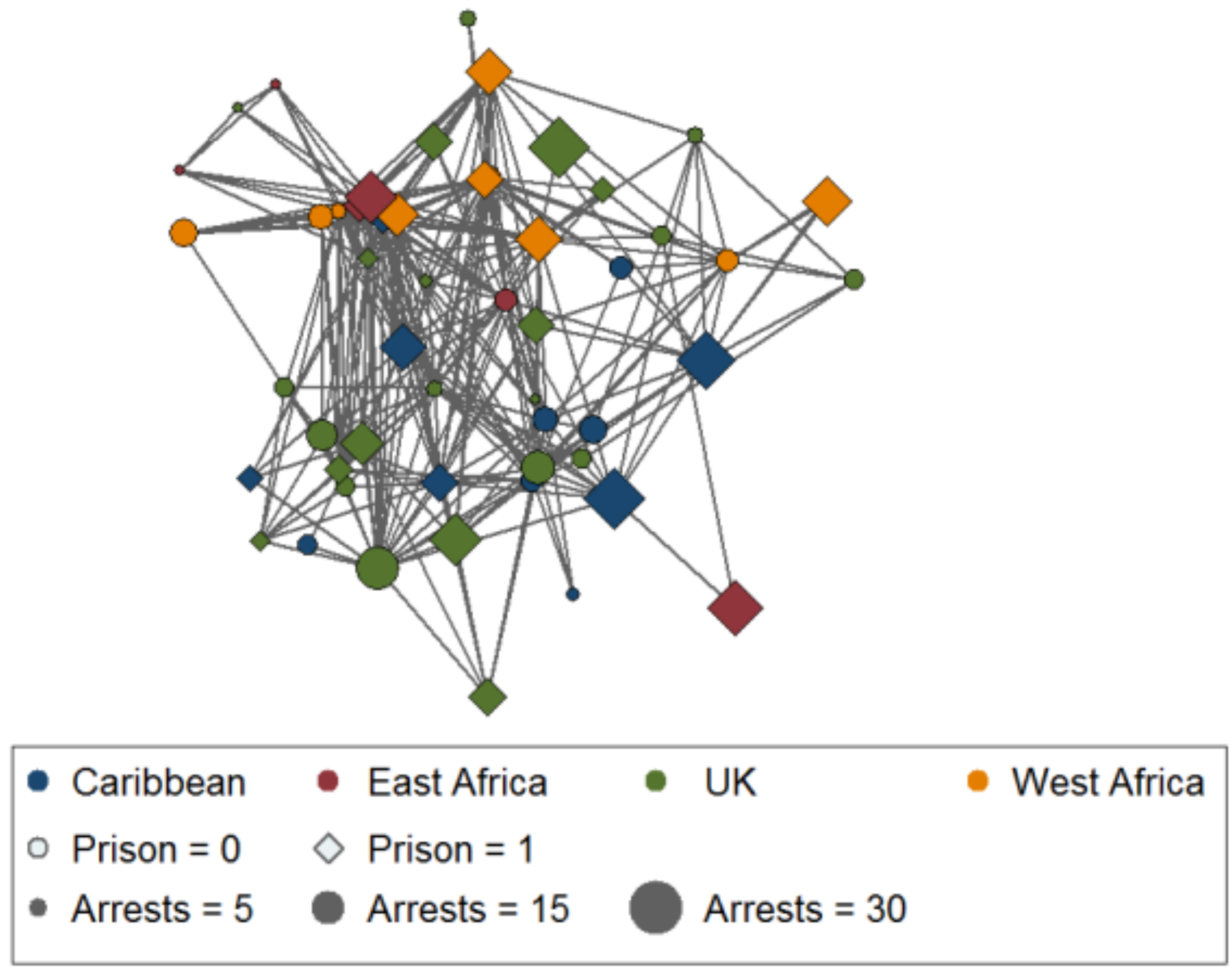
VISUALIZATION

Nuffield Network 2008

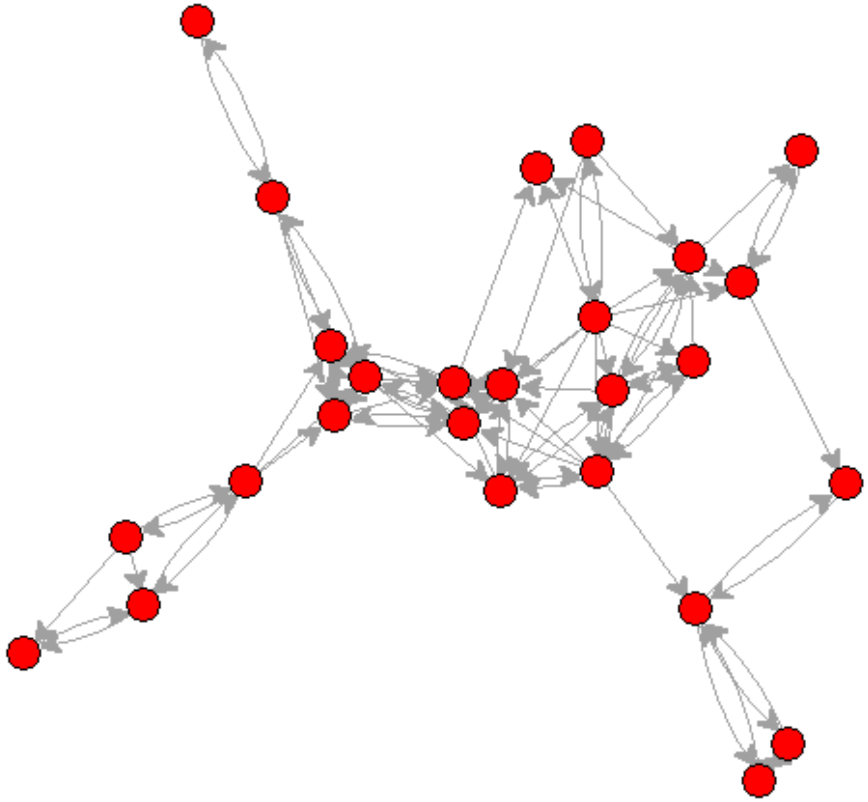




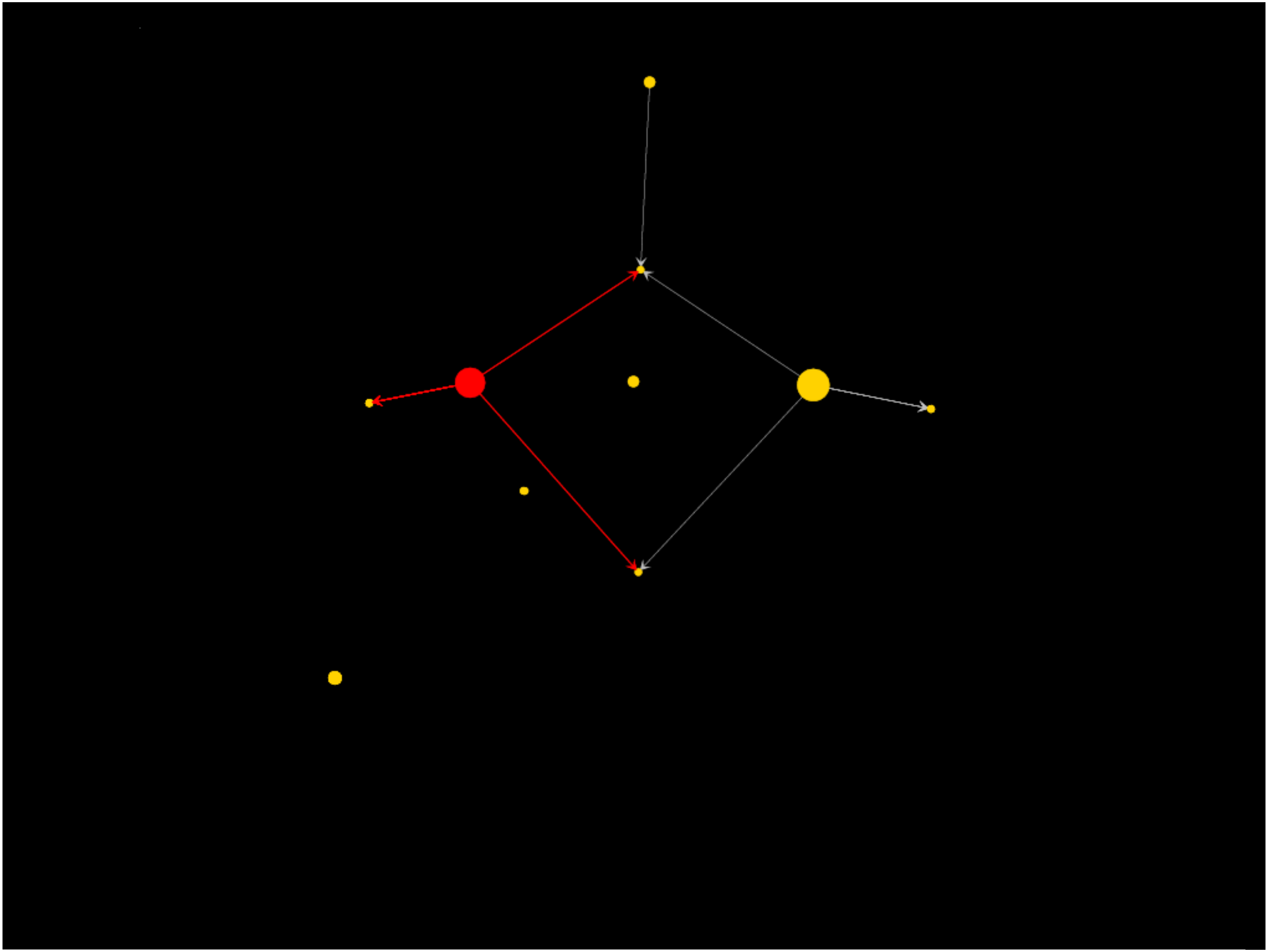

```
. webnwuse gang
. nwplot gang, color(Birthplace)
```

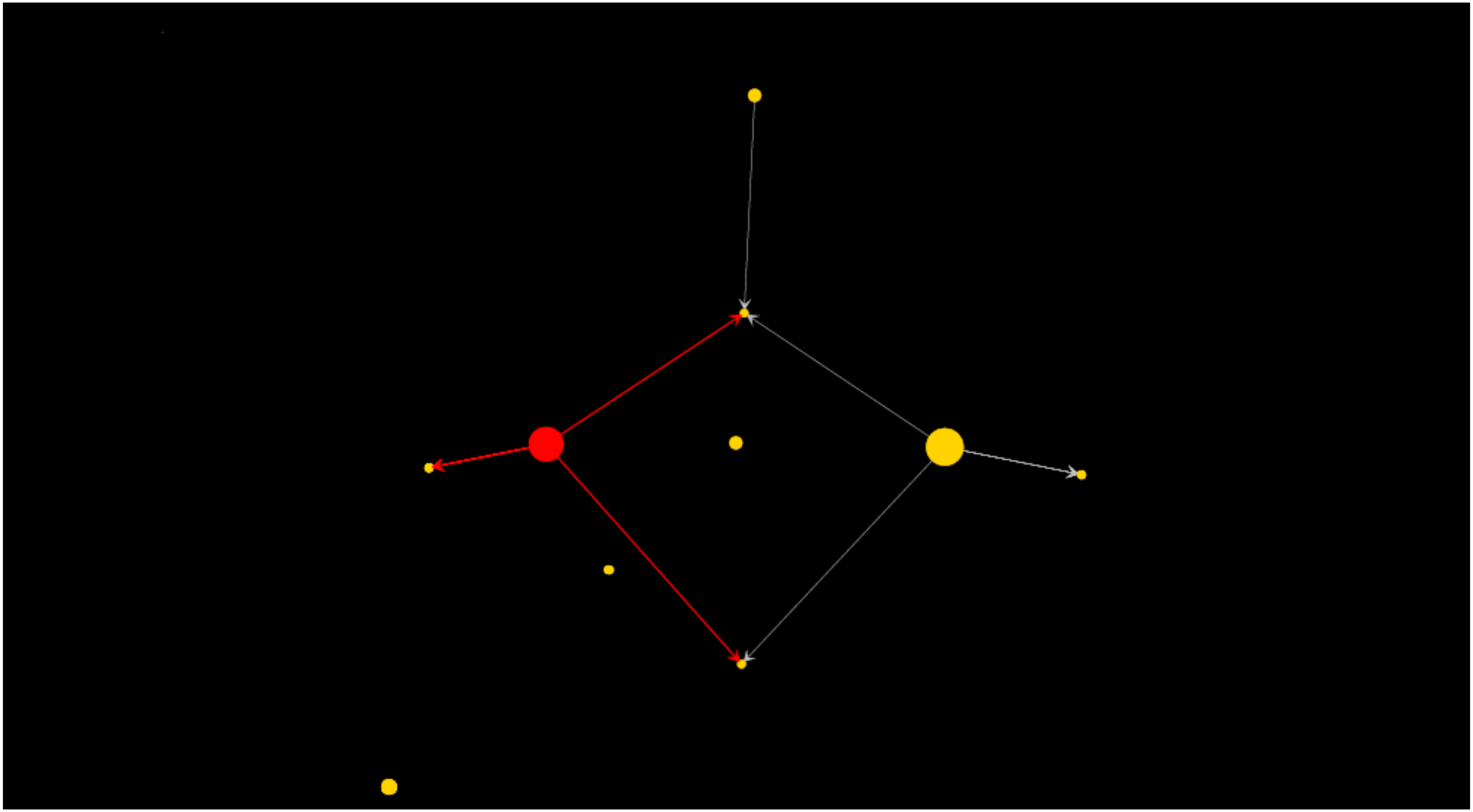



```
nwplot gang, color(Birthplace) symbol(Prison) size(Arrests)
```

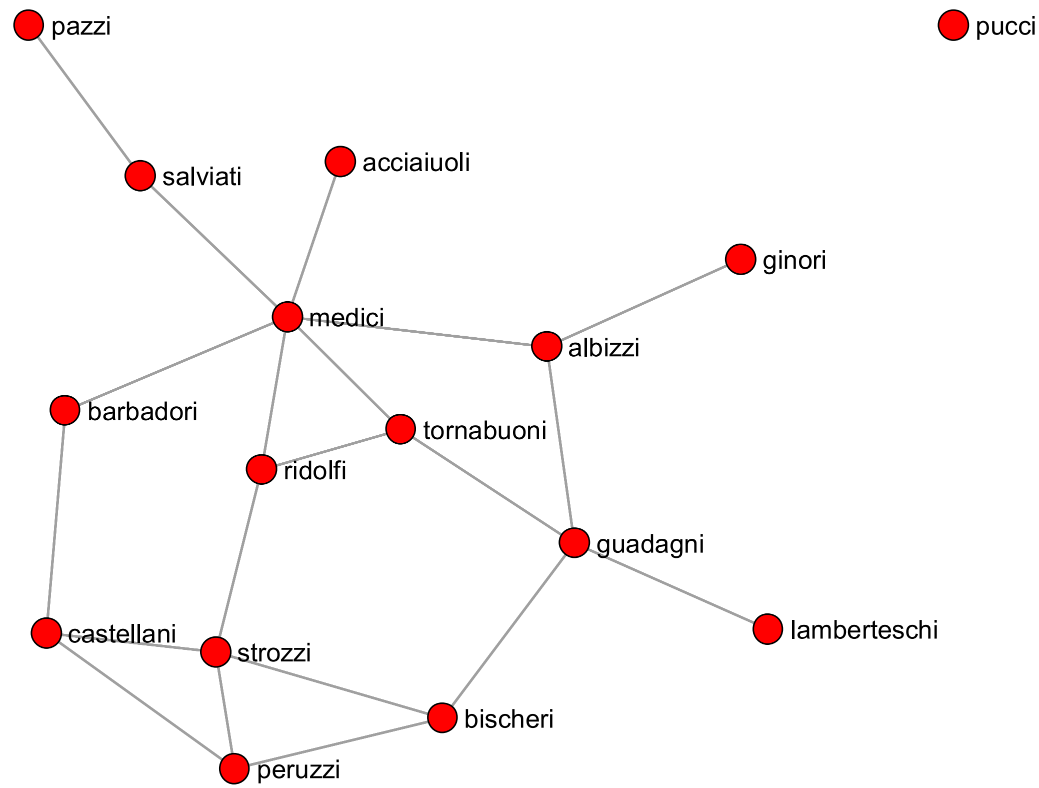


```
. webnwuse klas12
. nwmovie klas12_wave1-klas12_wave4
```

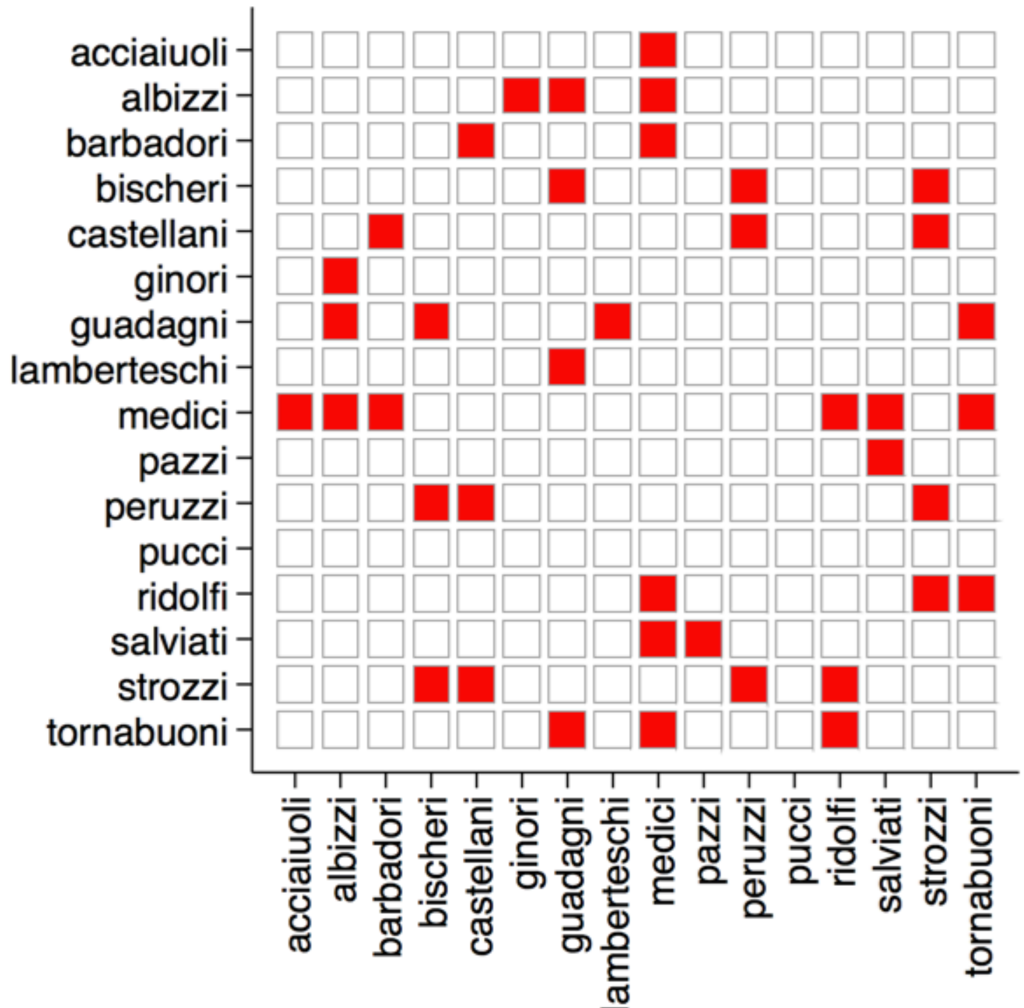




```
. nwmovie _all, colors(col_t*) sizes(siz_t*) edgecolors(edge_t*)
```




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

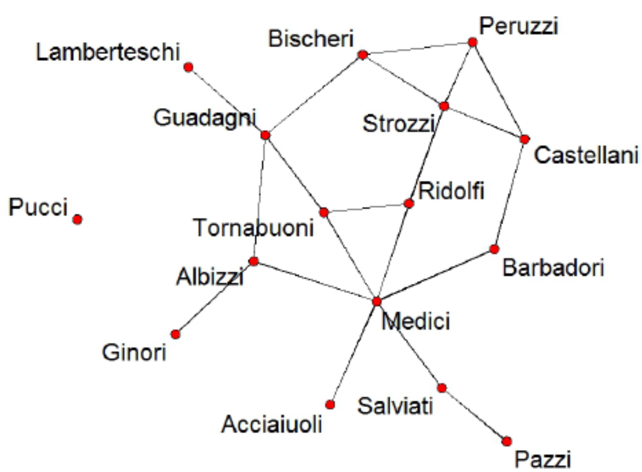



```
. nwplotmatrix flomarriage, lab
```

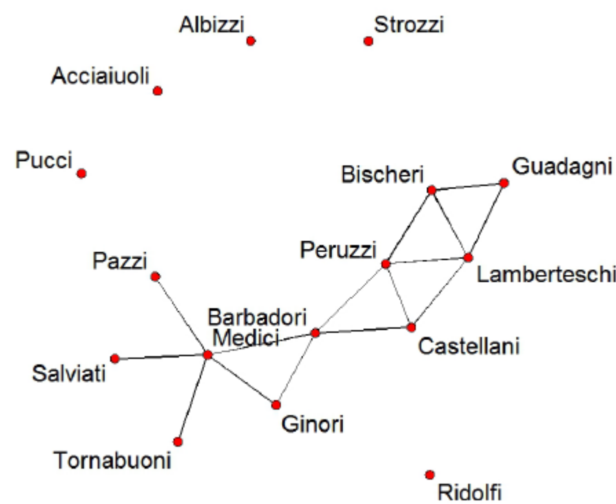
HYPOTHESIS TESTING

Is a particular network pattern more (or less) prominent than expected?

Question: Is there more or less correlation between these two networks than expected?



Florentine Marriage Network

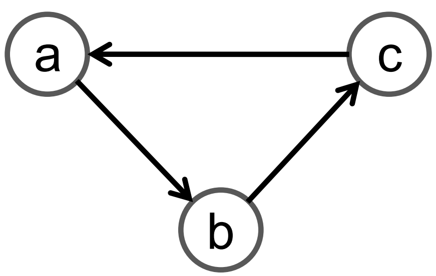


Florentine Business Ties

Padgett, J. and Ansell, C. (1993) Robust Action and the Rise of the Medici, 1400-1434. *American Journal of Sociology* 98: 1259-1319

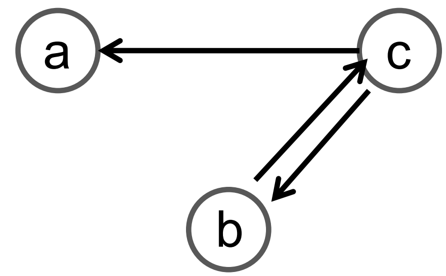
GRAPH CORRELATION

Network 1



$$\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$$

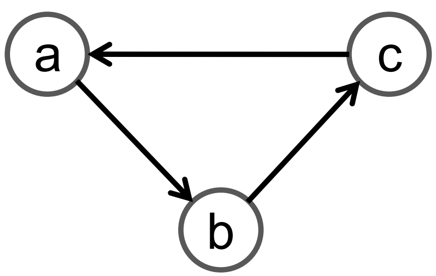
Network 2



$$\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{array}$$

GRAPH CORRELATION

Network 1



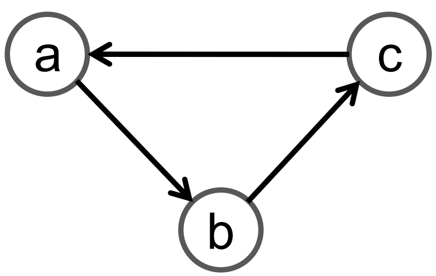
$$
\begin{array}{c c}
 & \begin{array}{ccc} a & b & c \end{array} \\
\begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}
\end{array}
=
$$

Transform adjacency matrix in a dataset of dyads.

row	col	net1
a	b	1
a	c	0
b	a	0
b	c	1
c	a	1
c	b	0

GRAPH CORRELATION

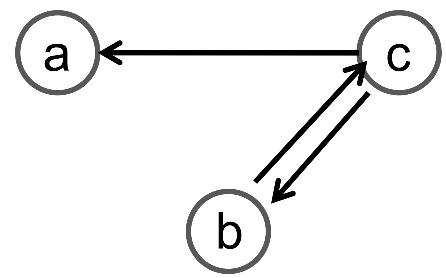
Network 1



$$
\begin{array}{c c}
 & \begin{array}{ccc} a & b & c \end{array} \\
\begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}
\end{array} =
$$

net1
1
0
0
1
1
0

Network 2

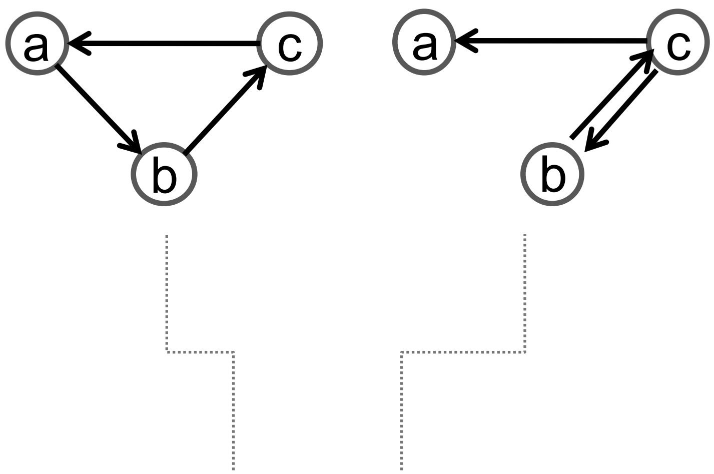


$$
\begin{array}{c c}
 & \begin{array}{ccc} a & b & c \end{array} \\
\begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}
\end{array} =
$$

net2
0
0
0
1
1
1

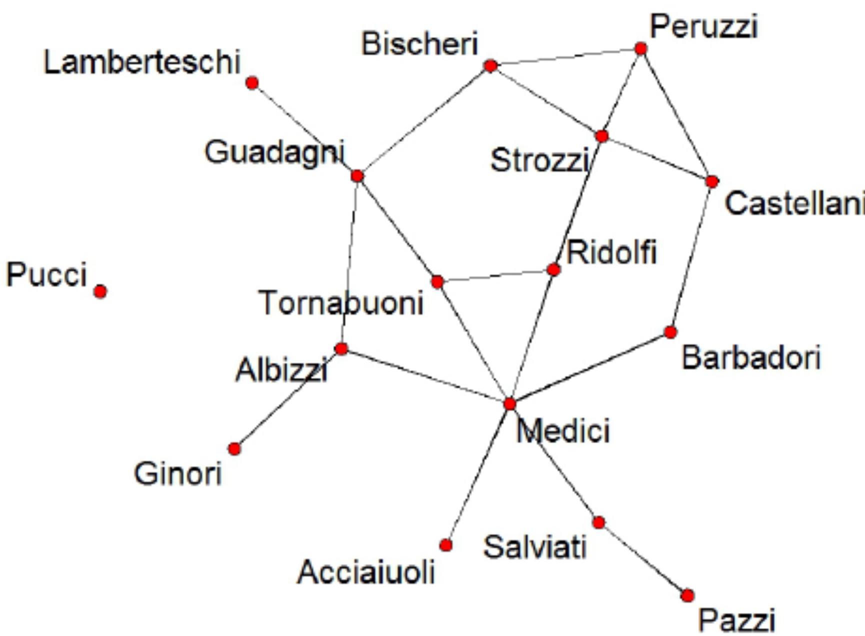
GRAPH CORRELATION

row	col	net1	net2
a	b	1	0
a	c	0	0
b	a	0	0
b	c	1	1
c	a	1	1
c	b	0	1

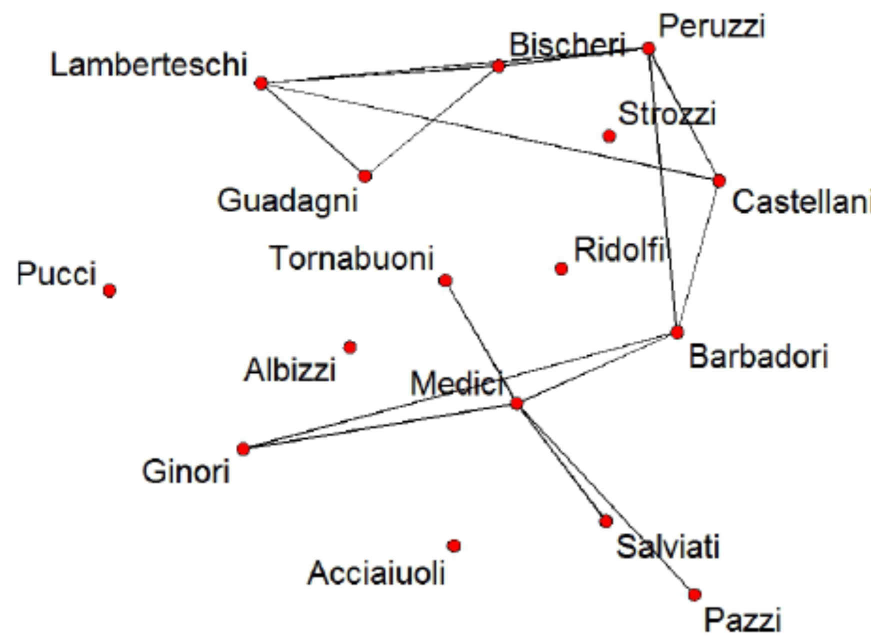


$$corr(net1, net2) = 0.333$$

GRAPH CORRELATION



Florentine Marriage Network



Florentine Business Ties

$$corr_{obs} = 0.372$$

Is this a lot?

Problem: We do not know how much correlation we should expect by chance given the marriage and the business network!

1

Test-statistic

$$corr_{obs} = 0.372$$

2

Distribution of test-statistic under null hypothesis

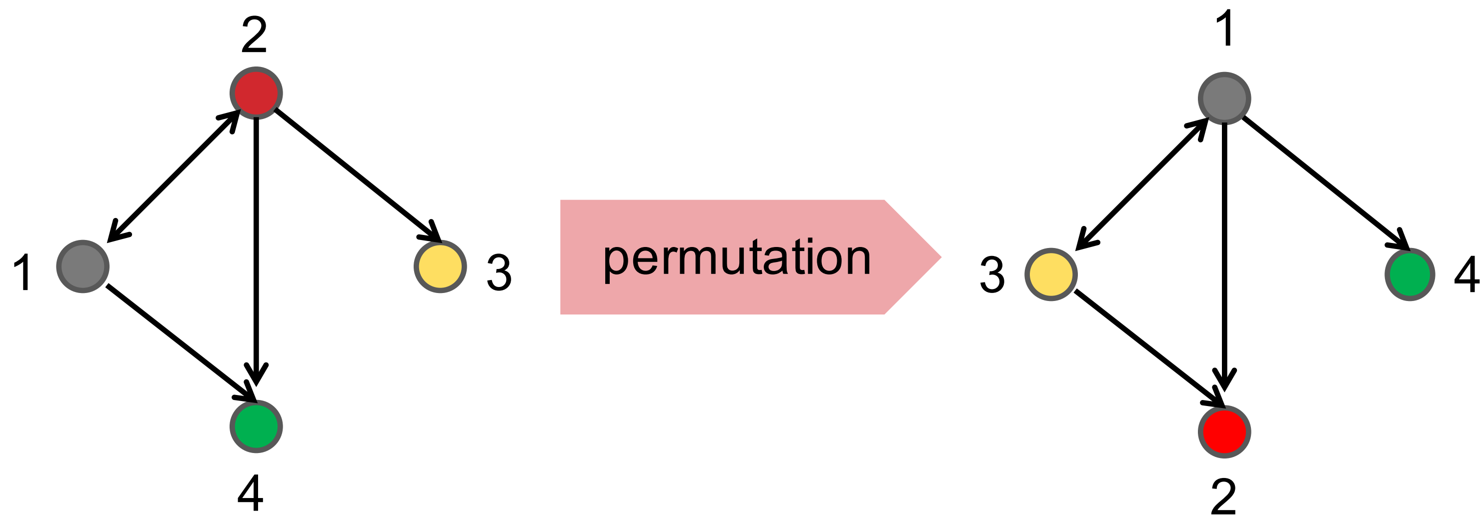
$$corr_{random} = ??$$

QUADRATIC ASSIGNMENT PROCEDURE

- Scramble the network by permuting the actors (randomly re-label the nodes), i.e. the actual network does not change, however, the position each node takes does.
- Re-calculate the test-static on the permuted networks and compare it with test-statistic on the unscrambled network.

Network structure is 'controlled' for. Keeps dependencies.

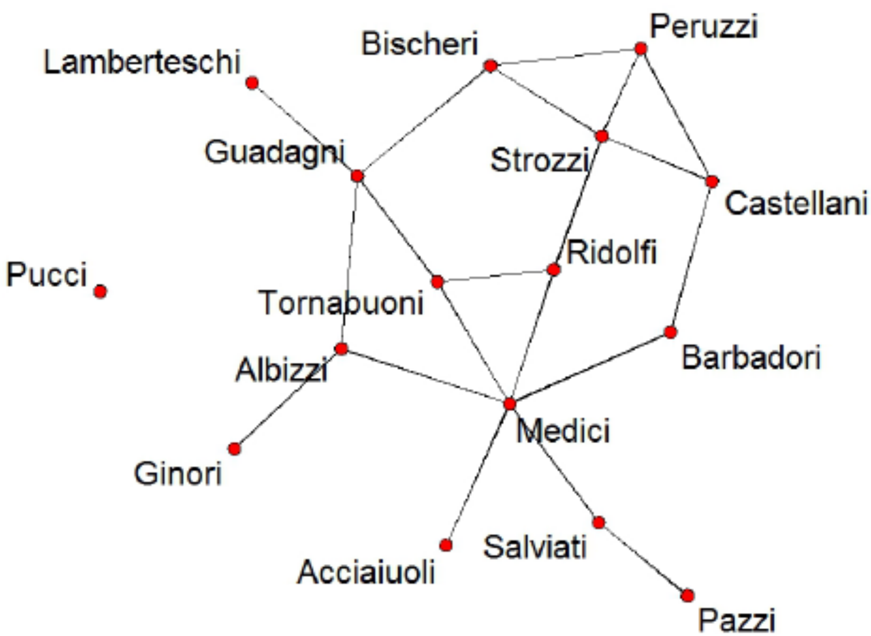
PERMUTATION TEST



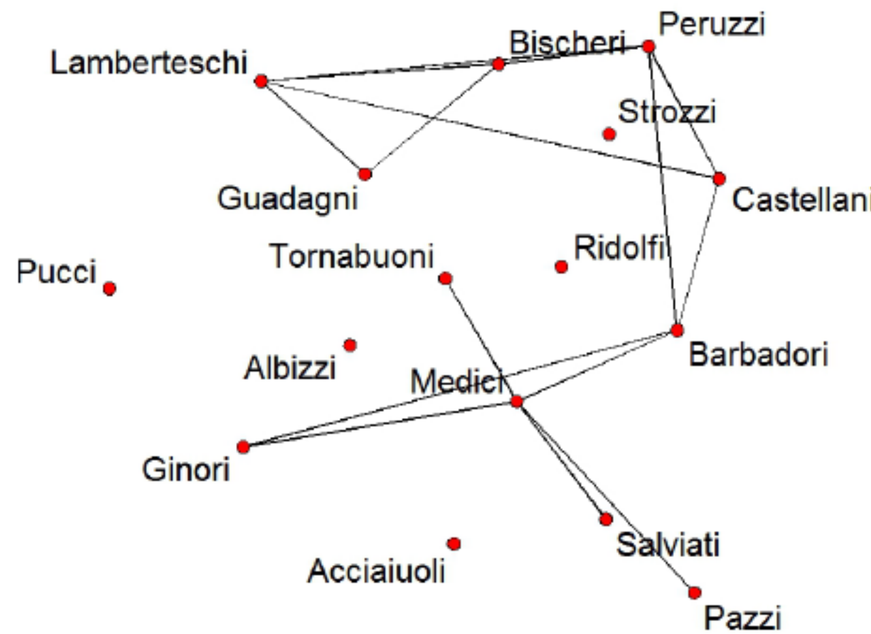
-	1	0	1
1	-	1	1
0	0	-	0
0	0	0	-

-	1	1	1
0	-	0	0
1	1	-	0
0	0	0	-

GRAPH CORRELATION



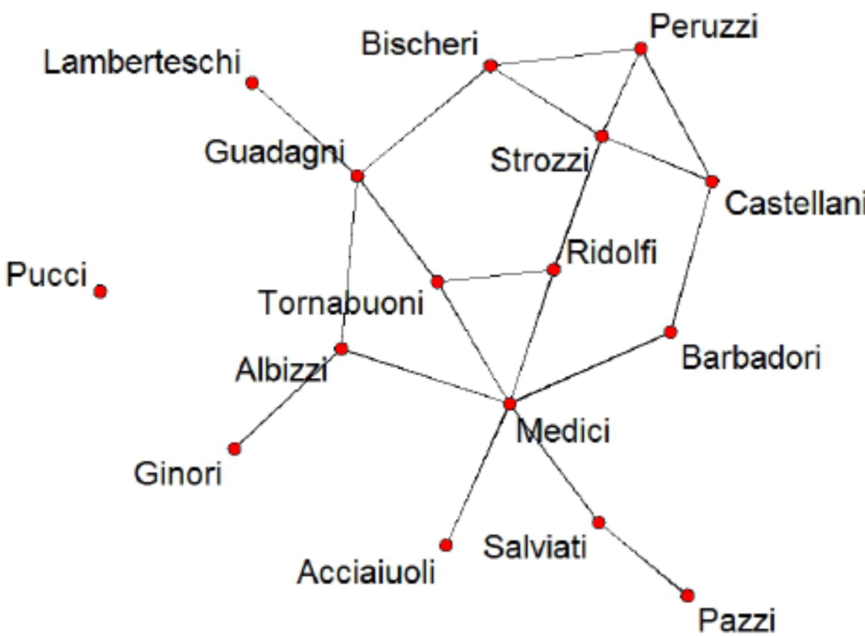
Florentine Marriage Network



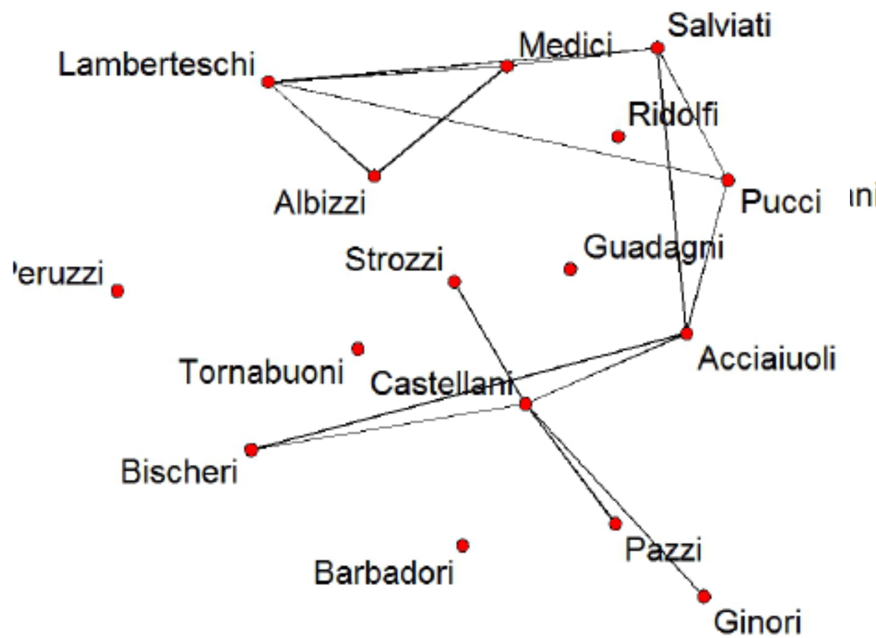
Florentine Business Ties

$$corr_{obs} = 0.372$$

GRAPH CORRELATION



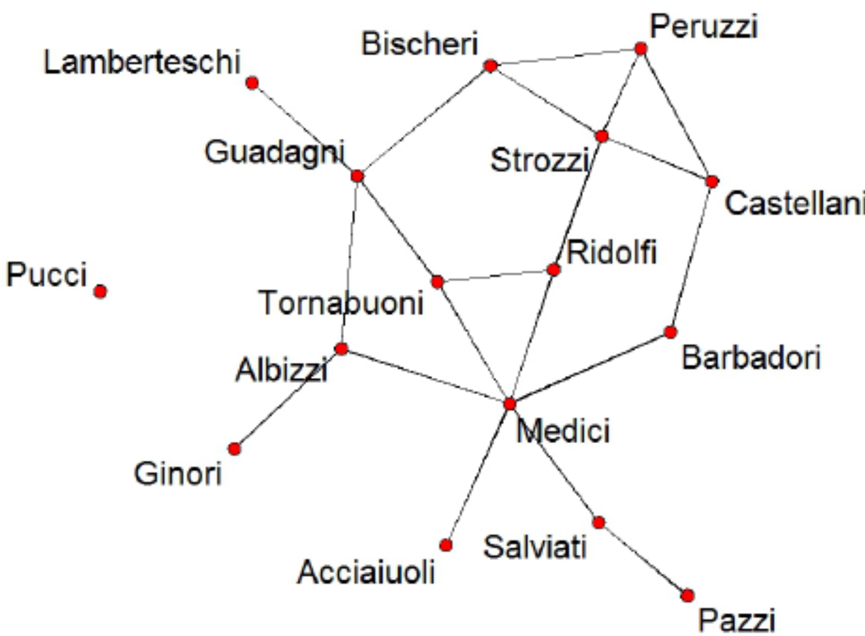
Florentine Marriage Network



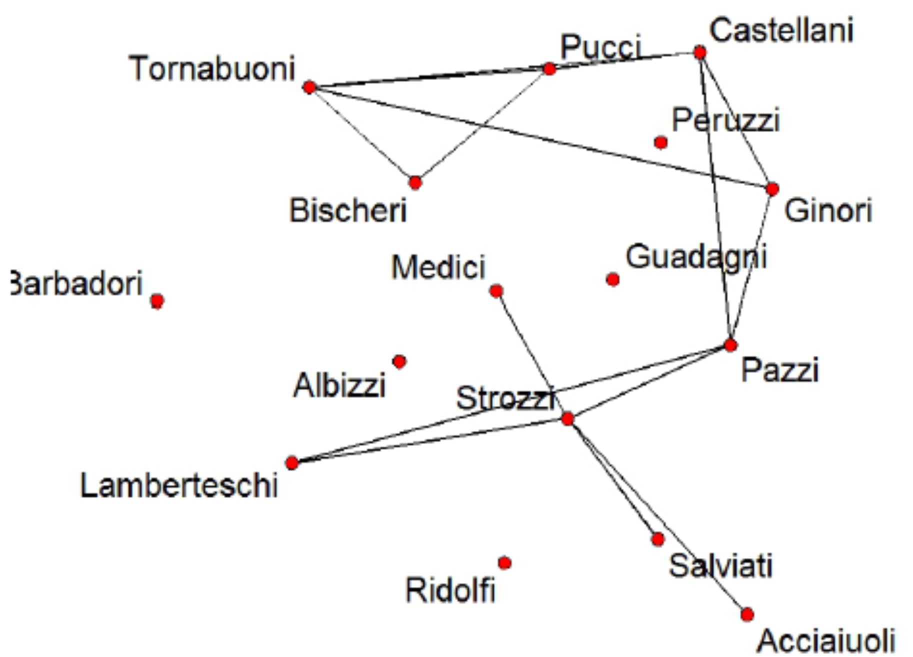
Florentine Business Ties

$$corr = -0.034$$

GRAPH CORRELATION



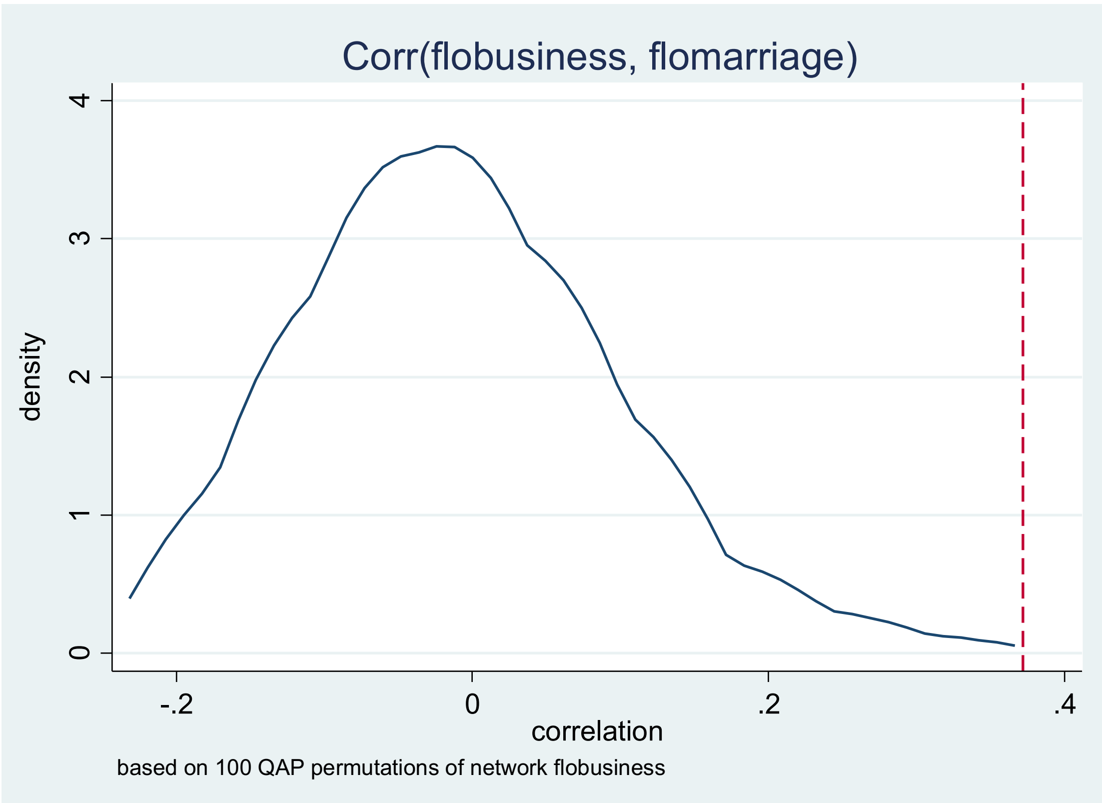
Florentine Marriage Network



Florentine Business Ties

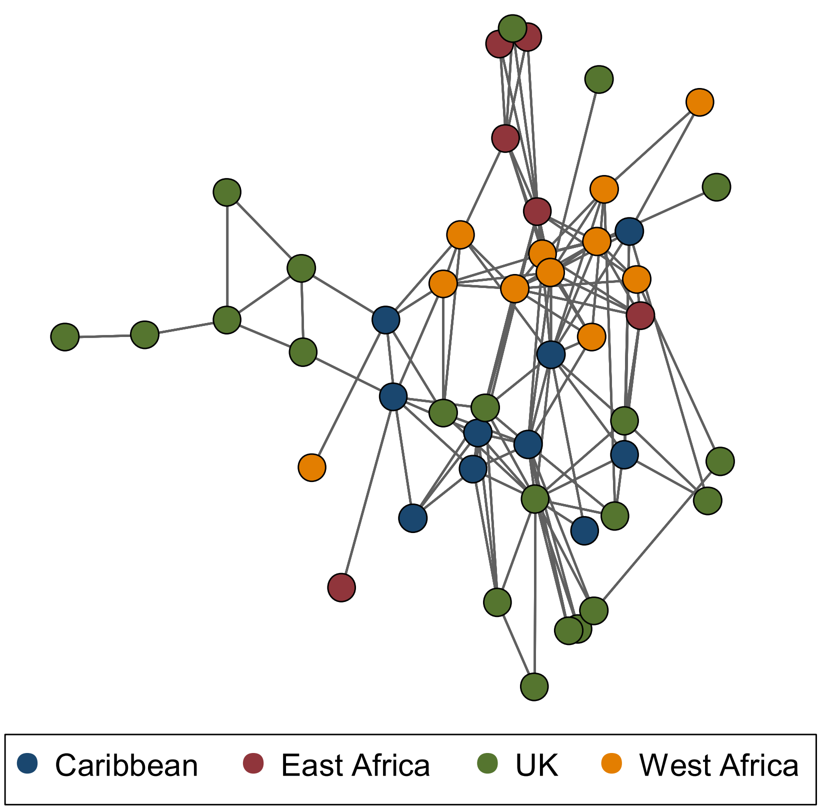
$$corr = -0.101$$

GRAPH CORRELATION




```
nwcorrelate flobusiness flomarriage, permutations(100)
```

Question: Are co-offending ties between gang members from the same ethnicity more likely than ties between gang members from different ethnicities?



QAP REGRESSION

- We can use the QAP principle to run
  1. Dyad-level logistic regression on dyadic dataset
  2. Permute network many times
  3. Run dyad-level logistic regression on permuted networks
  4. Compare regression estimate from unscrambled network with regression estimates obtained with permuted networks to derive standard errors.

For example:. Grund, T. and Densley, J. (2012) Ethnic Heterogeneity in the Activity and Structure of a Black Street Gang. ***European Journal of Criminology*, Vol. 9, Issue 3, pp. 388-406.**

```
. nwqap gang Birthplace Residence Arrests, mode(same same absdist) permutations(200)

Permutation: 1 out of 200
Permutation: 50 out of 200
Permutation: 100 out of 200
Permutation: 150 out of 200
Permutation: 200 out of 200


Multiple Regression Quadratic Assignment Procedure

  Estimation              =  QAP
  Regression              =  logit
  Permutations            =  200
  Number of vertices      =  54
  Number of edges         =  133
```

gang	Coef.	P-value
same_Birthplace	.859192	.005
same_Residence	.186923	.41
absdist_Arrests	-.036064	.095
_cons	-2.447445	

EXPONENTIAL RANDOM GRAPH MODELS

ERGM

Y_{ij}^{c} = all dyads other than Y_{ij}

Amount by which the feature $s_k(y)$ changes when Y_{ij} is toggled from 0 to 1.

$$\text{logit}\left[P\left(Y_{ij} = 1 \middle| n\ actors, Y_{ij}^{c}\right)\right] = \sum_{k=1}^{K} \theta_k \boxed{\delta s_k(\boldsymbol{y})}$$

Probability that there is a tie from *i* to *j*.

Given, *n* actors AND the rest of the network, excluding the dyad in question!

ERGM

$\boldsymbol{Y} = \boldsymbol{random\ variable}$, a randomly selected network from the pool of all potential networks

$\boldsymbol{y} = \boldsymbol{observed\ variable}$, here observed network

$\boldsymbol{\theta} = \boldsymbol{parameters}$, to be estimated

$$P(\boldsymbol{Y} = \boldsymbol{y}|\theta) = \frac{e^{\left(\theta^T s(\boldsymbol{y})\right)}}{c(\theta)}$$

A score given to our network **y** using some parameters θ and the network features **s** of **y**

Probability to draw 'our' observed network **y** from all potential networks

A score given to all other networks we could have observed

ERGM: INTEPRETATION

ERGM's ultimately give you an estimate for various parameters θ_k, which mean…

If a potential tie $Y_{ij} = 1$ (between *i* and *j*) would change the network statistic s_k by one unit.

This changes the log-odds for the tie Y_{ij} to actually exist by θ_k.

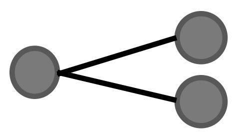
EXAMPLE

Consider an ERGM for an undirected network with parameters for these three statistics:

1) number of edges

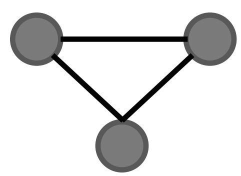
$$s_{edges}(y) = \sum y_{ij}$$

2) number of 2-stars



$$s_{2stars}(y) = \sum y_{ij}\, y_{ik}$$

3) number of triangles



$$s_{triangles}(y) = \sum y_{ij}\, y_{jk} y_{ik}$$

Then the 3-parameter ERG distribution function is:

$$P(\boldsymbol{Y} = \boldsymbol{y}|\theta) \propto e^{\left(\theta_{edges} s_{edges}(y) + \theta_{2stars} s_{2stars}(y) + \theta_{triangles} s_{triangles}(y)\right)}$$

```
. nwergm gang, formula(edges + nodematch("Birthplace") + gwesp(0.5, fixed=T))
```

```
Exponential random graph analysis

 Number of vertices      =  54
 Number of edges/arcs    =  133
 Directed                =  FALSE
 Estimation              =  MLE
 Iterations              =  3 out of 20
 MCMC sample size        =  4096
 AIC                     =  741.4
 BIC                     =  757.2
```

network	Observed	Coef.	Std.Err.	MCMC%	P>|z|
edges	133	-4.585	.235	0	0
nodematch.Birthplace	63	.518	.122	0	0
gwesp.fixed.0.5	165.121	1.434	.151	0	0

ERGM FEATURES

- Think of ERG models as a probability distribution on a (huge) space of all possible networks.

- The observed network is modelled as if it has been drawn from this distribution.

- The model parameters θ are
  - Attached to network statstictis s
  - These statistics in general correspond to subgraph counts (local patterns, ’motifs’)
  - The parameters describe the relative prevalence of the corresponding subgraph in ’generating’ the total graph.

- The parameters θ are estimated in such a way that each change of a tie (during the process of ’generating’ a network) is considered for the next ties that could change. Structure is ***endogenous*** **=> dyadic dependence model**

SOCIAL NETWORK ANALYSIS USING STATA

11 September 2015
UK Stata Group, London

Thomas Grund
thomas.u.grund@gmail.com

www.grund.co.uk
www.nwcommands.org