http://nwcommands.org
http://nwcommands.org

About

Here you find 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.grundi@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.
WORKSHOPS

14 November, Florence, Italian Stata Group
11/12 and 18/19 December, Cologne, University of Cologne
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

---

ThomasGrund / nwcommands

<table>
<thead>
<tr>
<th>File</th>
<th>Date</th>
<th>Description</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>3sept2014</td>
<td></td>
<td>18 hours</td>
</tr>
<tr>
<td>demo</td>
<td>3sept2014</td>
<td></td>
<td>18 hours</td>
</tr>
<tr>
<td>development</td>
<td>3sept2014</td>
<td></td>
<td>18 hours</td>
</tr>
<tr>
<td>extract_valuelabels.ado</td>
<td></td>
<td>Initialize Git</td>
<td>2 months</td>
</tr>
<tr>
<td>_nrevalnotexp.ado</td>
<td></td>
<td>Initialize Git</td>
<td>2 months</td>
</tr>
<tr>
<td>_nswsyntax.ado</td>
<td></td>
<td></td>
<td>18 hours</td>
</tr>
<tr>
<td>_nswsyntax_other.ado</td>
<td></td>
<td>Initialize Git</td>
<td>2 months</td>
</tr>
<tr>
<td>_opts_oneof.ado</td>
<td></td>
<td></td>
<td>24 days</td>
</tr>
<tr>
<td>animate.ado</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>animate.sthlp</td>
<td>3sept2014</td>
<td></td>
<td>18 hours</td>
</tr>
</tbody>
</table>
INSTALLATION

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

Or

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

. nwininstall, all
help nwcommands

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[NW-1]</td>
<td>Introduction and concepts</td>
</tr>
<tr>
<td>[NW-2]</td>
<td>Topical list of network commands</td>
</tr>
<tr>
<td>[NW-3]</td>
<td>Alphabetical list of network commands</td>
</tr>
<tr>
<td>[NW-4]</td>
<td>Getting started</td>
</tr>
<tr>
<td>[NW-5]</td>
<td>Network programming</td>
</tr>
<tr>
<td>[NW-6]</td>
<td>Install Stata menus/dialogs</td>
</tr>
</tbody>
</table>

*! Date        : 11sept2015
*! Version     : 1.4.8
*! Authors     : Thomas U. Grund
*! Contact     : thomas.u.grund@gmail.com
*! Web         : http://nwcommands.org
*! Bugs        : mailto:bug@nwcommands.org
Checking `nwcommands-ext` consistency and verifying not installing into /Users/thomasgrund/Library/Application Support/.

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 Stata
- `nwcommands-help` Social Network Analysis Using Stata
- `nwcommands-dlg` Social Network Analysis Using Stata
- `nwcommands-ext` Social Network Analysis Using Stata
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
**ADJACENCY MATRIX**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
# ADJACENCY MATRIX

The diagram on the left represents a graph with vertices 1 through 7, and the adjacency matrix on the right shows the connections between these vertices. The matrix indicates whether there is an edge between two vertices:

- A 1 in the matrix indicates an edge exists between the corresponding vertices.
- A 0 indicates no edge exists.

The adjacency matrix for the graph is as follows:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

This matrix accurately reflects the graph structure, with edges connecting vertices 1-3, 2-3, and 3-4.
ADJACENCY LIST

<table>
<thead>
<tr>
<th>ego</th>
<th>alter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>
ADJACENCY LIST

**Diagram:**
- Nodes: 1, 2, 3, 4, 5, 6, 7
- Edges:
  - 1 to 2
  - 1 to 3
  - 2 to 4
  - 2 to 5
  - 5 to 6
  - 6 to 7

**Table:**

<table>
<thead>
<tr>
<th>ego</th>
<th>alter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>3, 1</td>
</tr>
<tr>
<td>6</td>
<td>4, 1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>6, 1</td>
</tr>
</tbody>
</table>
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, nwcbelowte, nwcollapse, nwexpand, nwreplace, nwrecode, nwunab and more.
• Stata intuition just works.
<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mynet</td>
<td>Just one network</td>
</tr>
<tr>
<td>mynet1 mynet2</td>
<td>Two networks</td>
</tr>
<tr>
<td>mynet*</td>
<td>All networks starting with mynet</td>
</tr>
<tr>
<td>*net</td>
<td>All networks ending with net</td>
</tr>
<tr>
<td>my*t</td>
<td>All networks starting with my and ending with t</td>
</tr>
<tr>
<td>my~t</td>
<td>One network starting with my and ending with t</td>
</tr>
<tr>
<td>my?t</td>
<td>All networks starting with my and ending with t</td>
</tr>
<tr>
<td>mynet1-mynet6</td>
<td>mynet1, mynet2, ..., mynet6</td>
</tr>
<tr>
<td>_all</td>
<td>All networks in memory</td>
</tr>
</tbody>
</table>
“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]
```
. nwset _all
. nwplot, lab
nwset ego alter, edgelist
nwplot, lab
LIST ALL NETWORKS

```
.nwds
network     network_1

.nwset
(2 networks)
```

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
NODE ATTRIBUTES
NODE ATTRIBUTES

```
webnwuse florentine, nwclear
```

- 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
```

<table>
<thead>
<tr>
<th></th>
<th>wealth</th>
<th>priorates</th>
<th>seat</th>
<th>_nodelab</th>
<th>_nodevar</th>
<th>_nodeid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>53</td>
<td>1</td>
<td>acciaiuoli</td>
<td>acciaiuoli</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
<td>65</td>
<td>1</td>
<td>albizzi</td>
<td>albizzi</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>55</td>
<td>0</td>
<td>0</td>
<td>barbadori</td>
<td>barbadori</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>44</td>
<td>12</td>
<td>1</td>
<td>bischeri</td>
<td>bischeri</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>22</td>
<td>1</td>
<td>castellani</td>
<td>castellani</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>ginori</td>
<td>ginori</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>21</td>
<td>1</td>
<td>guadagni</td>
<td>guadagni</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>lamberteschi</td>
<td>lamberteschi</td>
<td>8</td>
</tr>
</tbody>
</table>

- 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)
```
. nwsummarize network_1

Network name: network_1
Network id: 1
Directed: true
Nodes: 5
Arcs: 4
Minimum value: 0
Maximum value: 1
Density: .2
.nsummarize glasgow1, detail

Network name: glasgow1
Network id: 1
Directed: true
Nodes: 50
Arcs: 113
Minimum value: 0
Maximum value: 1
Density: .0461224489795918
Reciprocity: .527027027027027
Transitivity: .3870967741935484
Betweenness centralization: .0821793002915452
Indegree centralization: .119533527696793
Outdegree centralization: .0570595585172845
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

OBTAIN TIE VALUES

```bash
doctest: >>> nwsummarize network_1, matonly
```

```text
. nwsummarize network_1, matonly
```
# Obtain Tie Values

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

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

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mynet</td>
<td>The whole network.</td>
</tr>
<tr>
<td>mynet[2,3]</td>
<td>Specific tie value; toe that node 3 received from node 2.</td>
</tr>
<tr>
<td>mynet[(2::4),3]</td>
<td>All ties that node 3 receives from nodes 2 to 4.</td>
</tr>
<tr>
<td>mynet[(2::4),(3::4)]</td>
<td>All ties that nodes 3 to 4 receive from nodes 2 to 4.</td>
</tr>
</tbody>
</table>
| 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
```

---

<table>
<thead>
<tr>
<th></th>
<th>_nodeid</th>
<th>net1</th>
<th>net2</th>
<th>net3</th>
<th>net4</th>
<th>net5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Vars:** 8  **Order:** Dataset  **Obs:** 5  **Length:** 1  **Filter:** Off
. webnwuse florentine, nwclear

Loading successful
(2 networks)

  flobusiness
  flomarriage

. nwtabulate flomarriage

<table>
<thead>
<tr>
<th>Network: flomarriage</th>
<th>Directed: false</th>
</tr>
</thead>
<tbody>
<tr>
<td>flomarriage</td>
<td>Freq.</td>
</tr>
<tr>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>Total</td>
<td>120</td>
</tr>
</tbody>
</table>
TABULATE TWO NETWORKS

```
. nwtabulate fomarriage flobusiness

Network 1: fomarriage Directed: false
Network 2: flobusiness Directed: false

<table>
<thead>
<tr>
<th>fomarriage</th>
<th>flobusiness</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>93</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>Total</td>
<td>105</td>
<td>15</td>
</tr>
</tbody>
</table>
. nwtabulate flomarriage seat
(0 observations deleted)

Network: flomarriage  Directed: false
Attribute: seat

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

<table>
<thead>
<tr>
<th>from_seat</th>
<th>to_seat</th>
<th>0</th>
<th>1</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>8</td>
<td>32</td>
<td>40</td>
</tr>
</tbody>
</table>

E-I Index: -.2  p-value: .22
GANG NETWORK

```
.webnwuse gang, nwclear
Loading successful
(2 networks)
_____________
gang_valued
gang
```
. nwtabulate gang_valued

<table>
<thead>
<tr>
<th>gang_valued</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1,116</td>
<td>77.99</td>
<td>77.99</td>
</tr>
<tr>
<td>1</td>
<td>182</td>
<td>12.72</td>
<td>90.71</td>
</tr>
<tr>
<td>2</td>
<td>92</td>
<td>6.43</td>
<td>97.13</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>1.75</td>
<td>98.88</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1.12</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Total 1,431 100.00
RECODE TIE VALUES

.nwrecode gang_valued (2/4 = 99)

(gang_valued: 266 changes made)

.nwtabulate gang_valued

<table>
<thead>
<tr>
<th>gang_valued</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1,116</td>
<td>77.99</td>
<td>77.99</td>
</tr>
<tr>
<td>1</td>
<td>182</td>
<td>12.72</td>
<td>90.71</td>
</tr>
<tr>
<td>99</td>
<td>133</td>
<td>9.29</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>1,431</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>
FLORENTINE FAMILIES

Marriage ties

Business ties
REPLACE TIE VALUES

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

<table>
<thead>
<tr>
<th>flomarriage</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>83.33</td>
<td>83.33</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>10.00</td>
<td>93.33</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>6.67</td>
<td>100.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>120</strong></td>
<td><strong>100.00</strong></td>
<td></td>
</tr>
</tbody>
</table>

[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

`. nogen both = (flobusiness & flomarriage)`

`. nwtabulate both`

<table>
<thead>
<tr>
<th></th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>112</td>
<td>93.33</td>
<td>93.33</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>6.67</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>120</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>
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 netfcn2, 3) produce networks with an expression netexp. A network expression is very similar to normal expressions in Stata.
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
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

<table>
<thead>
<tr>
<th></th>
<th>Mutual</th>
<th>Asym</th>
<th>Null</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>39</td>
<td>35</td>
<td>1151</td>
</tr>
</tbody>
</table>

Reciprocity: .527027027027027
TRIAD CENSUS
TRIAD CENSUS

111U

012

120U

021D

003 012 102 021D 021U 021C 111D 111U
030T 030C 201 120D 120U 120C 210 300
```
.nwtriads glasgow1

Triad census:  glasgow1

<table>
<thead>
<tr>
<th></th>
<th>003</th>
<th>012</th>
<th>021D</th>
<th>021U</th>
</tr>
</thead>
<tbody>
<tr>
<td>16243</td>
<td>1470</td>
<td>5</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>021C</td>
<td>030T</td>
<td>030C</td>
<td>102</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>5</td>
<td>0</td>
<td>1724</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>120D</th>
<th>120U</th>
<th>120C</th>
<th>111D</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>2</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>111U</td>
<td>201</td>
<td>210</td>
<td>300</td>
<td></td>
</tr>
</tbody>
</table>

|     | 30   | 15   | 9    | 5    |

Transitivity: .3870967741935484
```
FLORENTINE FAMILIES

Who are the neighbors?
NEIGHBORS

. webnwuse florentine, nwclear

. nwneighbor flomarriage, ego(albizzi)

Network: flomarriage

Ego : albizzi
Neighbors : ginori, guadagni, medici
```plaintext
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

wealth = 3

wealth = 146
What is the average wealth of the “albizzi’s” network neighbors?
. nwmcontext flomarriage, attribute(wealth) stat(mean) generate(wmean)

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

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

. list _nodelab w*

<table>
<thead>
<tr>
<th>_nodelab</th>
<th>wealth</th>
<th>wmean</th>
<th>wmax</th>
<th>wmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>acciaiuoli</td>
<td>10</td>
<td>103</td>
<td>103</td>
<td>103</td>
</tr>
<tr>
<td>albizzi</td>
<td>36</td>
<td>47.66666</td>
<td>103</td>
<td>8</td>
</tr>
<tr>
<td>barbadori</td>
<td>55</td>
<td>61.5</td>
<td>103</td>
<td>20</td>
</tr>
<tr>
<td>bischeri</td>
<td>44</td>
<td>67.66666</td>
<td>146</td>
<td>8</td>
</tr>
<tr>
<td>castellani</td>
<td>20</td>
<td>83.33334</td>
<td>146</td>
<td>49</td>
</tr>
<tr>
<td>statistic</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>Mean of <em>varname</em> over network neighbors; default.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Maximum of <em>varname</em> over network neighbors.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>min</td>
<td>Minimum of <em>varname</em> over network neighbors.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sum</td>
<td>Sum of <em>varname</em> over network neighbors.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sd</td>
<td>Standard deviation of <em>varname</em> over network neighbors.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>meanego</td>
<td>Mean of <em>varname</em> over network neighbors and <em>ego</em>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maxego</td>
<td>Maximum of <em>varname</em> over network neighbors and <em>ego</em>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>minego</td>
<td>Minimum of <em>varname</em> over network neighbors and <em>ego</em>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sumego</td>
<td>Sum of <em>varname</em> over network neighbors and <em>ego</em>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sdego</td>
<td>Standard deviation of <em>varname</em> over network neighbors and <em>ego</em>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>context</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>outgoing</td>
<td>Network neighbors of node <em>ego</em> are all nodes <em>alter</em> who receive a tie from <em>ego</em>; default.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>incoming</td>
<td>Network neighbors of node <em>ego</em> are all nodes <em>alter</em> who send a tie to <em>ego</em>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>both</td>
<td>Network neighbors of node <em>ego</em> are all nodes <em>alter</em> who either receive or send a tie to/from <em>ego</em>.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DISTANCE

FAR FAR AWAY
DISTANCE

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

How can we calculate the distance?

• Matrix $\mathbf{y}$ indicates which row actor is directly connected to which column actor.

• The squared matrix $\mathbf{y}^2$ indicates which row actor can reach which column actor in two steps.

• The matrix $\mathbf{y}^l$ indicates who reaches whom in $l$ steps.

\[
\mathbf{y}^2 = \begin{pmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
\end{pmatrix}
\times
\begin{pmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
\end{pmatrix}
= \begin{pmatrix}
0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]
DISTANCE

\[
d_{ij} = \\
\begin{bmatrix}
0 & 1 & 1 & 2 & 2 \\
1 & 0 & 2 & 1 & 1 \\
1 & 2 & 0 & 3 & 3 \\
2 & 1 & 3 & 1 & 0
\end{bmatrix}
\]

average 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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>geodesic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>15</td>
<td>12.50</td>
<td>12.50</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>16.67</td>
<td>29.17</td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td>29.17</td>
<td>58.33</td>
</tr>
<tr>
<td>3</td>
<td>32</td>
<td>26.67</td>
<td>85.00</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>12.50</td>
<td>97.50</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>2.50</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Total 120 100.00
How can one get from the “peruzzi” to the “medici”? 
PATHS

. nopath 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**

```
lnwpath flomarriage, ego(peruzzi) alter(medici) generate(mypath)
```

Network: flomarriage

<table>
<thead>
<tr>
<th>Ego</th>
<th>Alter</th>
<th>Shortest path length</th>
<th>Selected length</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 (peruzzi)</td>
<td>9 (medici)</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

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

```
flobusiness
flomarriage
mypath_1
mypath_2
```

Diagram showing relationships between acciaiuoli, albizzi, barbadori, bischeri, castellani, ginori, guadagni, lamberteschi, medici, pazzi, peruzzi, pucci, ridolfi, salviati, strozzi, tornabuoni.
PATHS

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

```
mypath_1 = 0
mypath_1 = 1
```
PATHS OF SPECIFIC LENGTH

. npath 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

Well connected actors are in a structurally advantageous position.

• Getting jobs
• Better informed
• Higher status
• ...

What is “well-connected?”
Degree centrality

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

\[ C_{odegree}(i) = \sum_{j=1}^{N} y_{ij} \quad C_{idegree}(i) = \sum_{j=1}^{N} y_{ji} \]
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?

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

\( l_{ij} \) is the shortest path between \( i \) and \( j \)
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

\[ C_{\text{closeness}}(i) = \frac{1}{Farness(i)} \]

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

\[ C_{\text{closeness}}(b) = \frac{1}{\left[ \frac{1}{4} (1 + 2 + 2 + 2) \right]} = \frac{4}{7} \]
**BETWEENNESS CENTRALITY**

**Betweenness centrality**

- How many shortest paths go through an individual?

\[
C_{betweenness}(a) = 6 \\
C_{betweenness}(b) = 0
\]
Betweenness 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>_between</td>
<td>16</td>
<td>19.5</td>
<td>24.60111</td>
<td>0</td>
<td>95</td>
</tr>
</tbody>
</table>

. list _nodelab _between

<table>
<thead>
<tr>
<th>_nodelab</th>
<th>_between</th>
</tr>
</thead>
<tbody>
<tr>
<td>acciaiuoli</td>
<td>0</td>
</tr>
<tr>
<td>albizzi</td>
<td>38.66667</td>
</tr>
<tr>
<td>barbadori</td>
<td>17</td>
</tr>
<tr>
<td>bischeri</td>
<td>19</td>
</tr>
</tbody>
</table>
SIMULATION
RANDOM NETWORK

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

nwsmall 10, k(2) shortcuts(3)
PREFERENTIAL ATTACHMENT NETWORK

nwpref 10, prob(.5)
HOMOPHILY NETWORK

homophily = 5

homophily = -5

nwhomophily gender, density(0.05) homophily(5)
. webnwuse gang
. nplot gang, color(Birthplace)
nwplot gang, color(Birthplace) symbol(Prison) size(Arrests)
. webnwuse klas12
. nmovie 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?

GRAPH CORRELATION

Network 1

\[ \begin{bmatrix} a & b & c \\ a & 0 & 1 \\ b & 0 & 0 \\ c & 1 & 0 \end{bmatrix} \]

Network 2

\[ \begin{bmatrix} a & b & c \\ a & 0 & 0 \\ b & 0 & 1 \\ c & 1 & 0 \end{bmatrix} \]
Network 1

Transform adjacency matrix in a dataset of dyads.

\[
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}
\]
GRAPH CORRELATION

Network 1

\[
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0 \\
\end{bmatrix}
\]

= 

\[
\begin{bmatrix}
1 \\
0 \\
0 \\
1 \\
0 \\
\end{bmatrix}
\]

Network 2

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
\end{bmatrix}
\]

= 

\[
\begin{bmatrix}
0 \\
0 \\
1 \\
1 \\
\end{bmatrix}
\]
## Graph Correlation

<table>
<thead>
<tr>
<th>row</th>
<th>col</th>
<th>net1</th>
<th>net2</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>c</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>a</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>b</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$$corr(\text{net1}, \text{net2}) = 0.333$$
GRAPH CORRELATION

Florentine Marriage Network

Florentine Business Ties

\[
\text{corr}_{obs} = 0.372
\]
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

\[ \text{corr}_{obs} = 0.372 \]

2 Distribution of test-statistic under null hypothesis

\[ \text{corr}_{random} = \text{?} \]
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

\[
\begin{pmatrix}
-1 & 0 & 1 \\
1 & -1 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
-1 & 1 & 1 \\
0 & -1 & 0 \\
1 & 1 & -1 \\
0 & 0 & 0
\end{pmatrix}
\]
GRAPH CORRELATION

\[ corr_{obs} = 0.372 \]
GRAPH CORRELATION

\[ \text{corr} = -0.034 \]
GRAPH CORRELATION

Florentine Marriage Network

Florentine Business Ties

\[ corr = -0.101 \]
Corr(flobusiness, flomarriage)

based on 100 QAP permutations of network flobusiness

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.

. 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

<table>
<thead>
<tr>
<th></th>
<th>Coef.</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>same_Birthplace</td>
<td>0.859192</td>
<td>.005</td>
</tr>
<tr>
<td>same_Residence</td>
<td>0.186923</td>
<td>.41</td>
</tr>
<tr>
<td>absdist_Arrests</td>
<td>-0.036064</td>
<td>.095</td>
</tr>
<tr>
<td>_cons</td>
<td>-2.447445</td>
<td></td>
</tr>
</tbody>
</table>
EXPONENTIAL RANDOM GRAPH MODELS
ERGM

$Y_{ij}^c = \text{all dyads other than } Y_{ij}$

logit\left[P\left(Y_{ij} = 1 \mid n \text{ actors}, Y_{ij}^c \right)\right] = \sum_{k=1}^{K} \theta_k \delta s_k(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!

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

\[ Y = \text{random variable}, \text{ a randomly selected network from the pool of all potential networks} \]

\[ y = \text{observed variable}, \text{ here observed network} \]

\[ \theta = \text{parameters}, \text{ to be estimated} \]

\[
P(Y = y | \theta) = \frac{e^{\theta^T s(y)}}{c(\theta)}
\]

Probability to draw ‘our’ observed network \( y \) from all potential networks

A score given to our network \( y \) using some parameters \( \theta \) and the network features \( s \) of \( y \)

A score given to all other networks we could have observed
ERGM: INTERPRETATION

ERGM’s ultimately give you an estimate for various parameters $\theta_k$, 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 $\theta_k$. 
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:

\[
P(Y = y|\theta) \propto e^{(\theta_{edges}s_{edges}(y) + \theta_{2stars}s_{2stars}(y) + \theta_{triangles}s_{triangles}(y))}
\]
. nwgem gang, formula(edges + nodematch("Birthplace") + gwesp(0.5, fixed=T))

Exponential random graph analysis

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of vertices</td>
<td>54</td>
</tr>
<tr>
<td>Number of edges/arcs</td>
<td>133</td>
</tr>
<tr>
<td>Directed</td>
<td>FALSE</td>
</tr>
<tr>
<td>Estimation</td>
<td>MLE</td>
</tr>
<tr>
<td>Iterations</td>
<td>3 out of 20</td>
</tr>
<tr>
<td>MCMC sample size</td>
<td>4096</td>
</tr>
<tr>
<td>AIC</td>
<td>741.4</td>
</tr>
<tr>
<td>BIC</td>
<td>757.2</td>
</tr>
</tbody>
</table>

| 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    |
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 \( \theta \) are
- Attached to network statistics \( s \)
- 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 \( \theta \) 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 \textit{endogenous} => dyadic dependence model