Introduction to Social Network Analysis with R

Part 2: Basic SNA with R

Michał Bojanowski
m.bojanowski@uw.edu.pl
www.bojanorama.pl/snar:start

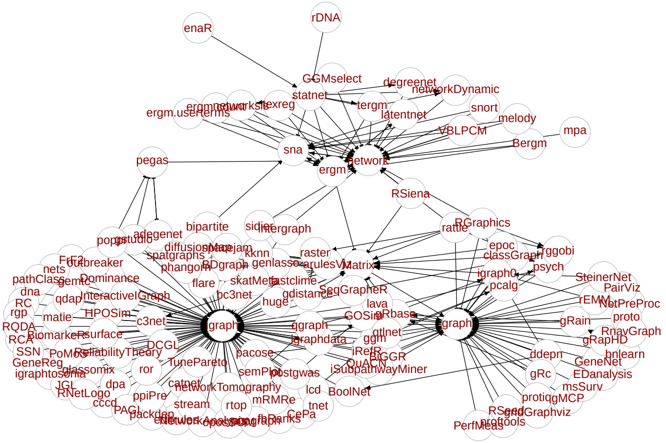
ICM, University of Warsaw

July 1, 2014
EUSN 2014, Barcelona

Outline of Part 2

Focus on igraph with pointers to network.

Network of network-related packages in R



Data used in part 2

`io`	Directed weighted graph of commodity flows between 21 industrial sectors in US (source: Bureau of Economic Analysis)
`ibe43`	Primary school classroom network, "With whom would you like to play with?". Source: (Polish) Institute for Educational Research (IBE)

Outline

Network objects

Packages igraph and network provide dedicated types of objects (classes) for storing network data:

- Package igraph provides objects of class "igraph".
- Package network provides objects of class "network".

Apart from storing relational data (nodes and ties), objects can also store node-, tie-, and network-level variables (called attributes).

"igraph" objects can be created

- from scratch using `graph` function.
- from adjacency matrices using `graph.adjacency`.
- from edgelists using `graph.edgelist`.
- from edge and vertex data frames using `graph.data.frame`.
- from specialized file formats: Pajek, GraphML, etc. with `read.graph`.

Basic properties of networks

- Network size and number of edges: `vcount` and `ecount`.
- `graph.density`
- Extracting relational information with `get.adjacency`, `get.edgelist`
- "Simplifying" networks by removing loops (self-edges) and/or multiple edges with `simplify`.

Vertex- / edge- / graph-level attributes

Attributes can be used to store additional information on nodes (e.g. gender), ties (e.g. value, strength), or network as a whole.

- Retrieve attributes with `get.vertex.attribute`, `get.edge.attribute`, and `get.graph.attribute`.
- Set attributes with `set.vertex.attribute`, `set.edge.attribute`, and `set.graph.attribute`.

▸ `rsna.R`: Basic properties

Outline

Packages igraph and network

Package igraph

- Binary directed or undirected networks
- Vertex/edge/network attributes
- Multiple ties per dyad
- Loops
- Bipartite networks

Package network adds

- Hypergraphs
- Encoding missingness of nodes/ties

Using them together can give some headaches. . .

igraph and network clashes and how to avoid them

There are several function name conflicts between igraph and network.

- The order in which packages are loaded matters.
- How to make sure that a proper version of the function is used?

Two suggested strategies:

1. Always detach the package that you are not about to use.
2. Explicitly specify from which package the conflicting function should be used with the `::` operator.

▸ `rsna.R`: Packages igraph and network

Outline

Conversions igraph <=> network with intergraph

Package intergraph provides two functions for converting "igraph" objects to "network" and vice versa:

- `asIgraph`
- `asNetwork`

All the attributes are copied appropriately.

▸ `rsna.R`: Package intergraph

Outline

Vertex and edge sequences

Vertex and edge sequences allow for

- Vertex and edge subscripting.
- Retrieve and set attributes of vertexes/edges.
- Identify edges based on incident vertexes and vice versa.

Vertex and edge sequences are created using functions `V` and `E` respectively. For example (`E` works in a similar way):

```
V(g)[ i ]$attrname
```

- `g` is an "igraph" object
- `attrname` is an optional name of vertex attribute
- Within `[ ]` we can
  - Subscript vertexes just like elements of a vector.
  - Use special functions exploiting adjacency information.

▸ rsna.R: Vertex and edge sequences

Outline

Subgraphs and components

Vertex and edge sequences are useful when we want to create a subgraph depending on the values of vertex or edge attributes.

- Create subgraphs using `induced.subgraph`, `delete.edges`, or `delete.vertices`.

Function `clusters` identifies weakly and strongly connected components, which can extracted using `induced.subgraph`.

▸ `rsna.R`: Subgraphs and components

Outline

Customizing network visualizations

- Several network layouts, see help of `layout`.
- Customizing graphical elements
  - Node size, shape, color
  - Edge color, width, curvature
  - Vertex label color, font, size

  See `help("igraph.plotting")`.
- Vertex and edge attributes like `color` or `size` etc. are interpreted like corresponding arguments to `plot`.

▸ `rsna.R`: Visualization

Outline

Selected SNA descriptives

- Dyad census (`dyad.census`)
- Triad census (`triad.census`)
- Network diameter: `diameter`, `get.diameter`
- Centrality indices: `betweenness`, `evcent`, `closeness`.
- Network segregation: computing mixing matrix and E-I index.

▸ `rsna.R`: SNA