# Two-mode networks

A two-mode network consists of two sets of units (e. g. people and events), relation connects the two sets, e. g. participation of people in social events.

There exist several two-mode networks:

- *Membership in institutions* - people, institutions, *is a member*, e.g. directors and commissioners on the boards of corporations.

- *Voting for suggestions* - polititians, suggestions, *votes for*.

- *'Buying articles in the shop'*, where first set consists of consumers, the second of articles, the connection tells which *article was bought by a consumer*.

- *Readers and magazines*.

- *Citation network*, where first set consists of authors, the second set consists of articles/papers, connection is a relation *author cites a paper*.

- *Co-autorship networks* - authors, papers, *is a (co)author*.

A corresponding graph is called *bipartite graph* – lines connect only vertices from one to vertices from another set – inside sets there are no connections.

# Example: participation of women in social events: davis.net – example from UCINET dataset:



Pajek - shadow [0.00,1.00]

# Transforming two-mode networks to ordinary valued networks

Two-mode network can be transformed to 'ordinary' network, where units are only units from first or only units from the second set.

The previous two-mode network (women-events) can be transformed to ordinary network, where units are women. Two women are in relation (in the corresponding graph there exists an undirected line) if they took part in at least one common event. The line value tells the number of events where both of them took place. Loop values represent total number of events for each woman.

But if we transform network to ordinary network where units are events, the two events are in relation (in the corresponding graph there exists an undirected line) if there exists at least one woman who took part in both events. The line value between two events tells the number of women who took place in both events. Loop values represent total number of women present at each event.
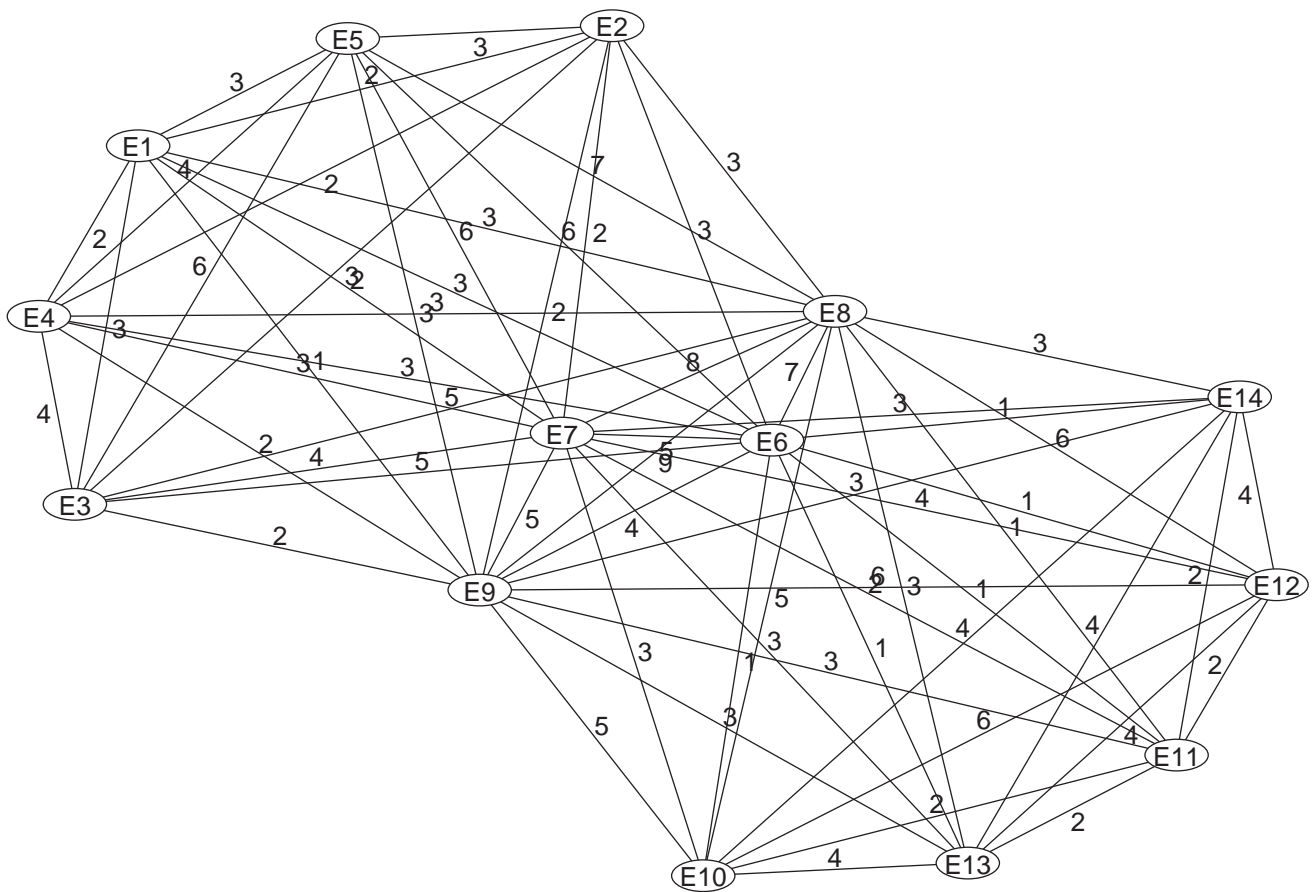
# Network of women and number of common events (without loops)

# Matrix representation
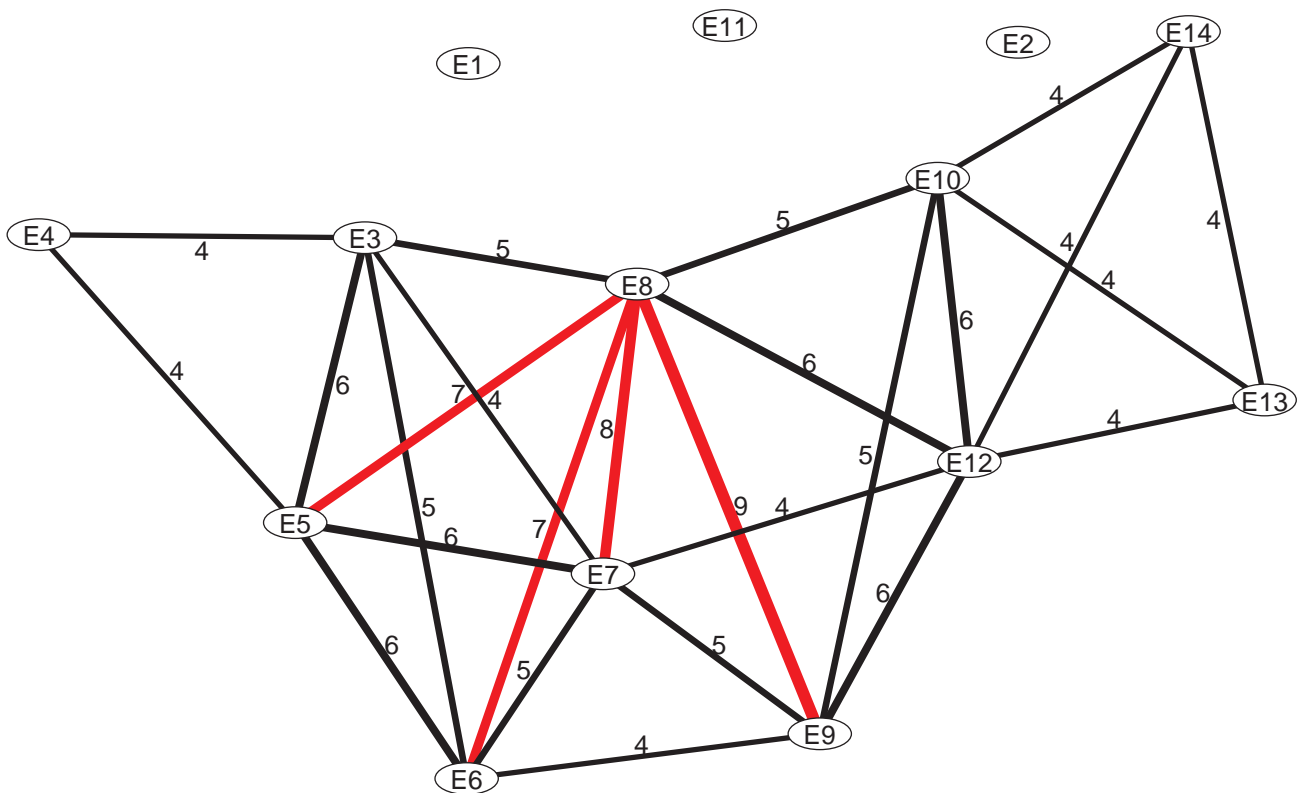
Pajek - shadow [0.00,8.00]
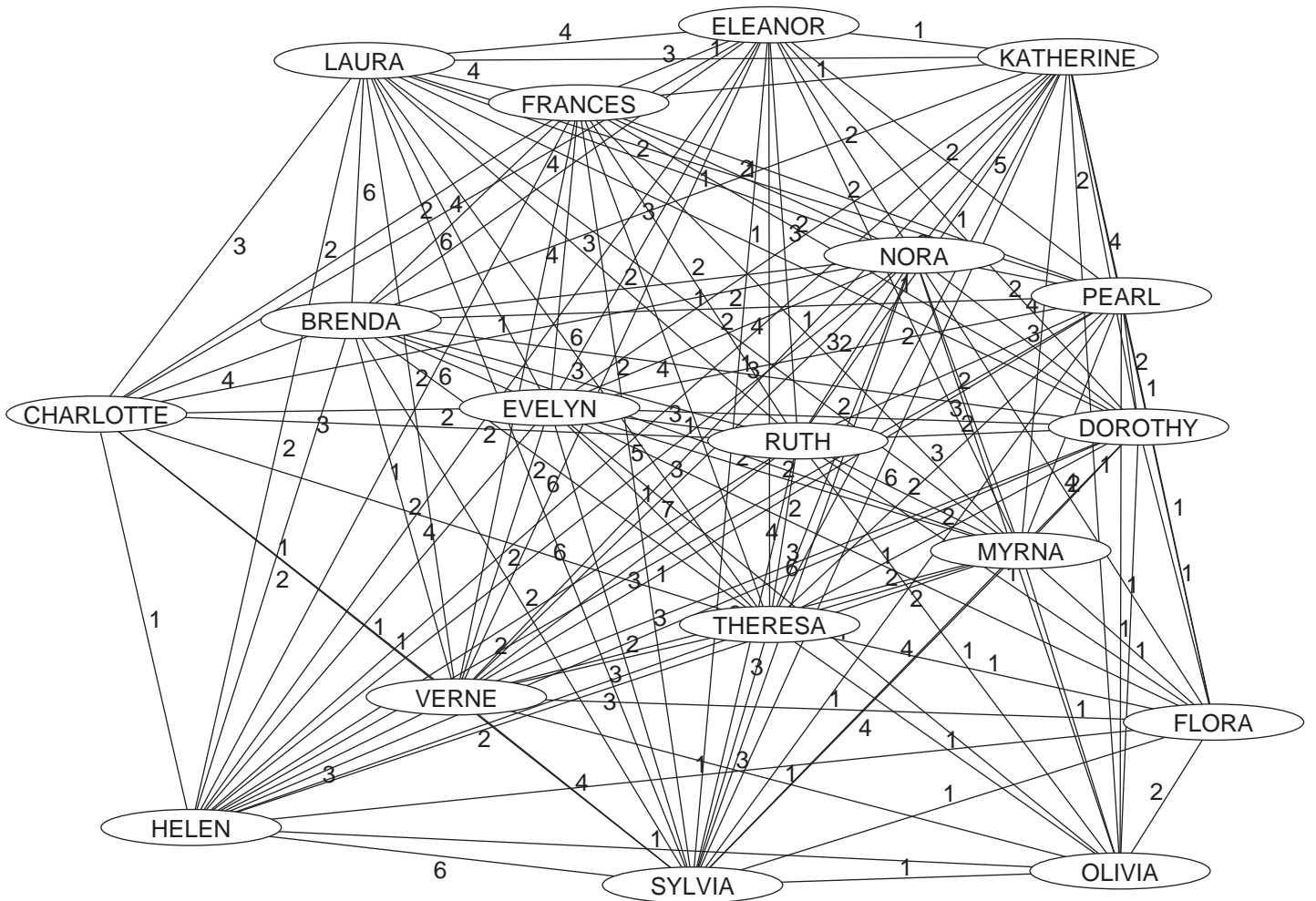
# Number of common events at least 3
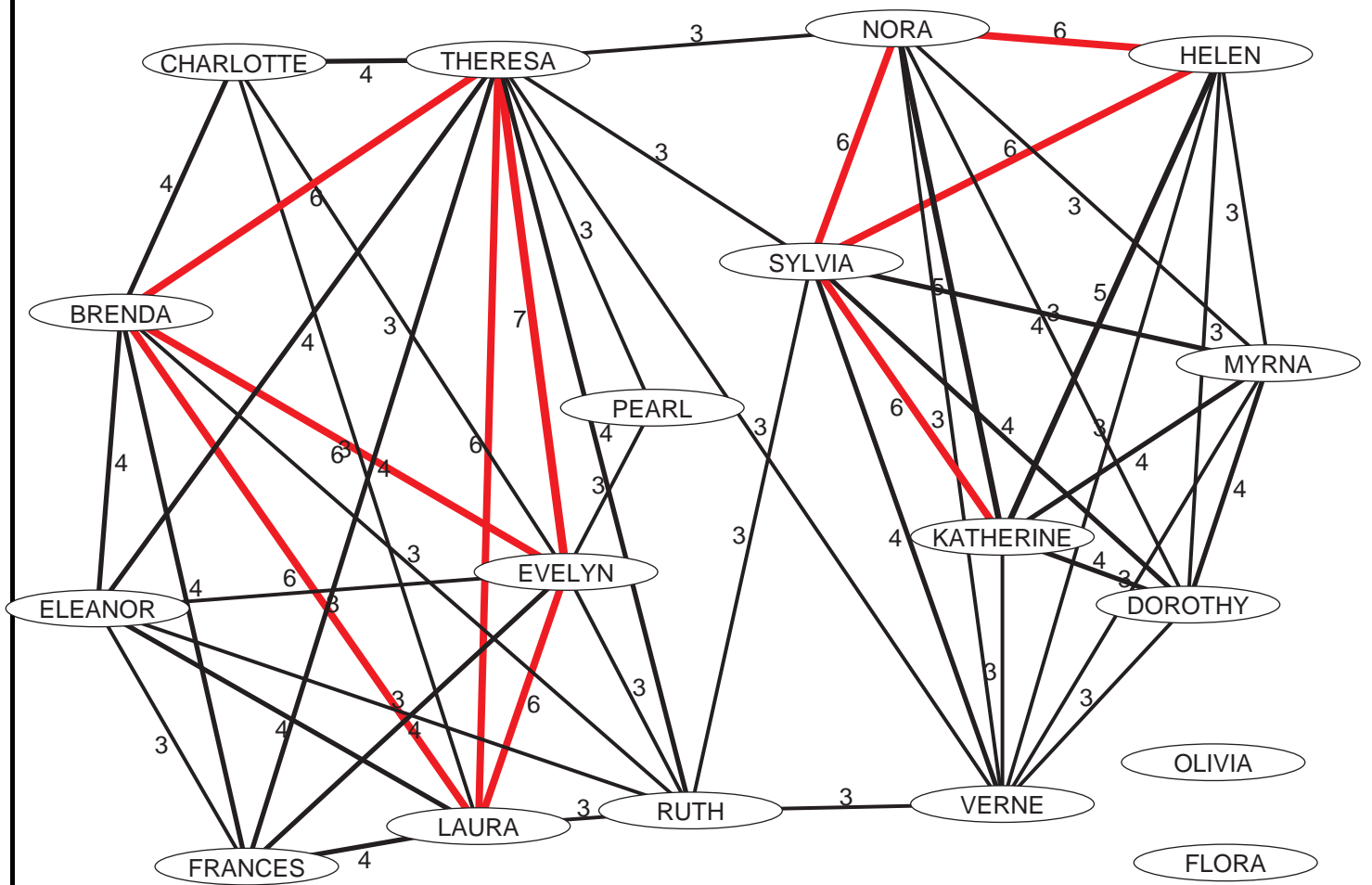
# Network of events and number of women taking part in connected events (without loops)

# Matrix representation

Pajek - shadow [0.00,14.00]

# Number of common women at least 4

# Transforming two-mode networks to ordinary networks in Pajek

A two-mode network is defined on an input file in the following way (Davis.net):

```
*Vertices 32 18
 1 EVELYN            27 E9
 2 LAURA             28 E10
 3 THERESA           29 E11
 4 BRENDA            30 E12
 5 CHARLOTTE         31 E13
 6 FRANCES           32 E14
 7 ELEANOR           *Edgeslist
 8 PEARL             1 19 20 21 22 23 24 26 27
 9 RUTH              2 19 20 21 23 24 25 26
10 VERNE             3 20 21 22 23 24 25 26 27
11 MYRNA             4 19 21 22 23 24 25 26
12 KATHERINE         5 21 22 23 25
13 SYLVIA            6 21 23 24 26
14 NORA              7 23 24 25 26
15 HELEN             8 24 26 27
16 DOROTHY           9 23 25 26 27
17 OLIVIA            10 25 26 27 30
18 FLORA             11 26 27 28 30
19 E1                12 26 27 28 30 31 32
20 E2                13 25 26 27 28 30 31 32
21 E3                14 24 25 27 28 29 30 31 32
22 E4                15 25 26 28 29 30 31 32
23 E5                16 26 27 28 30
24 E6                17 27 29
25 E7                18 27 29
26 E8
```

**Explanation**

The only difference comparing to ordinary networks is that we have to specify the total number of vertices (in our case 32) and number of vertices that belong to the first subset (in our case 18 women). First, all vertices from the first subset must be listed (women) and afterwards all vertices from the second subset (events). Partition into the two subsets is obtained using Net/Partitions/2-Mode, where value 1 is given to vertices from the first subset (women), and value 2 to vertices from the second subset (events).

**Transforming to valued networks**

The network is transformed into an ordinary network, where the vertices are elements from the first subset (in our case women), using Net/Transform/2-Mode to 1-Mode/Rows.
If we want to get a network with elements from the second subset we use Net/Transform/2-Mode to 1-Mode/Columns.
Network with or without loops can be generated: Net/Transform/2-Mode to 1-Mode/Include Loops.

We can generate network with values on lines (in our case number of common events) or network with multiple lines – for each common event a line between corresponding two women: Net/Transform/2-Mode to 1-Mode/Multiple Lines

For our purposes we will always generate a valued network, sometime the loops will be useful sometimes not.

## Obtaining picture of valued network

We store values of loops (e.g. total number of events for each women) into vector using Net/Vector/Get Loops and use later this vector to determine sizes of vertices (Draw/Draw-Vector). After network with values on lines is generated we can visualize it in different ways:

- picture of complete network: First we draw network using Energy drawing, providing that option Options/Values of Lines/Similarities is selected (vertices connected with higher values will be drawn closer).

  Values of lines can be shown using different widths of lines (Options/Lines/Different Widths) and/or greyscale (Options/Lines/GreyScale).

  After that we export picture to SVG, where we can interactively add lines according to their values: Export/SVG/LineValues/Nested Classes.

- picture of the most important part of the network: only lines with values which are large enough are kept.

  Distribution of line values can obtained using

Info/Network/Line Values.

According to the distribution, we remove lines with low vaues from the network using Net/Transform/Remove/lines with value/lower than and entering the limit value.

# Analysis of obtained valued network

We will use concept of an $m$-slice: a subnetwork defined by the value of lines. Concept is similar to $k$-cores:

An $m$-**slice** is a maximal subnetwork containing the lines with value equal to or greater than $m$ and the vertices incident with these lines.

# $m$-**slices in Pajek**

If you just want to know the $m$-slices to which the vertices belong, you may use the commands in the submenu Net/Partitions/Valued Core, provided that the network does not contain multiple lines or loops. Make sure that the option Use max instead of sum is checked in the submenu before you execute the First Threshold and Step.

Choose Input or Output because we are analyzing an undirected network. In a dialog box, you must specify the first

threshold, which is the upper limit of the lowest class. Note that each class includes its upper limit. The default threshold, 0, is the correct choice. A second dialog box asks for the step value, which is the class width. It is one by default and this is a good value for $m$-slices because it creates a class for each consecutive value.

Now, Pajek creates a partition with class numbers corresponding to the highest $m$-slice each vertex belongs to. In addition, Pajek reports a frequency table with the distribution of vertices over $m$-slices (denoted by $m$). Note that loops (if present) are taken into consideration in the computation, so **remove** them from the network first, otherwise isolated vertices may be regarded as having one or more neighbors.

Finally, Pajek offers a powerful tool to show $m$-slices interactively on the web with the Export/SVG submenu. Select the command Nested Classes from the submenu Line Values to obtain an HTML-file with a drawing of the network. If you open this file in an internet browser with the SVG plug-in installed, you will see the layout and a set of checkboxes to its right. Each checkbox is associated with a class of lines, which you specified in a dialog box. If you deselect a checkbox, all lines with values up to and including the deselected class are removed from the picture as well as all vertices which are

isolated when the lines are removed (try women/events).

# Slovenian magazines and journals

Over 100.000 people have been asked which magazines and journals they read (survey conducted in 1999 and 2000, source CATI Center Ljubljana). They listed 124 different magazines and journals. The collected data can be represented as 2-mode network:

```
                Delo       Dnevnik     Sl.novice ...
Reader1          X                        X        ...
Reader2                       X                    ...
Reader3          X                                 ...
..........     .....      .......      .......... ...
```

From 2-mode network reader/journal we generated ordinary network, where the vertices are journals

- undirected edge with value a between journals means the number of readers of both journals.

- loop on selected journal means the number of all readers that read this journal.

Obtained matrix (A):

```
            Delo     Dnevnik     Sl.novice ...
Delo       20714        3219          4214 ...
Dnevnik                15992          3642 ...
Sl. novice                          31997 ...
........   ......      .....      ..... ...
```

Because of the huge differences in number of readers of different journals it is very hard to make any conclusions about journals connectivity according to raw data. First we have to normalize network. There exist several types of normalizations:

$$\text{Geo}_{ij} = \frac{a_{ij}}{\sqrt{a_{ii}a_{jj}}}$$

$$\text{Input}_{ij} = \frac{a_{ij}}{a_{jj}} \qquad \text{Output}_{ij} = \frac{a_{ij}}{a_{ii}}$$

$$\text{Min}_{ij} = \frac{a_{ij}}{\min(a_{ii}, a_{jj})} \qquad \text{Max}_{ij} = \frac{a_{ij}}{\max(a_{ii}, a_{jj})}$$

$$\text{MinDir}_{ij} = \begin{cases} \frac{a_{ij}}{a_{ii}} & a_{ii} \leq a_{jj} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{MaxDir}_{ij} = \begin{cases} \frac{a_{ij}}{a_{jj}} & a_{ii} \leq a_{jj} \\ 0 & \text{otherwise} \end{cases}$$

In all cases each element of the matrix is divided by the proper diagonal element(s).

## Geo normalization

Using Geo normalization we divide elements of the matrix by geometric mean of both diagonal elements. In our case this normalization is a measure for connectivity (correlation) among journals (see, revije3.htm, revije4.htm).

## MinDir normalization

MinDir normalization makes two things:

- it converts undirected network to directed - the arcs point from the journal with less readers to the journals with higher number of readers

- the value on the arc corresponds to the percentage of readers of the first journal, who read also second one.

In our case this normalization measures the dependance between journals (see figures revije5.htm, revije6.htm). The size of vertices corresponds to number of readers. The colors determine weakly connected components.

# Normalisations in Pajek

If we want to use normalisations we must obtain the ordinary network with option **include loops** checked and **multiple lines** not checked.

Normalisations are available at:
Net/Transform/2-Mode to 1-Mode/Normalize 1-Mode.

After normalization is done we first remove loops using Net/Transform/Remove/loops (since we are not interested in loops and all loops have line value 1).

The obtained network is much to dense (every vertex is connected to several other vertices), therefore we check the distribution of line values (Info/Network/Line Values) and remove lines with low line values:
Net/Transform/Remove/lines with value/lower than

After that some vertices can be isolated (without any neighbours), so we remove the isolated vetices
Net/Transform/Reduction/Degree/All and type 1 and answer Yes to additional question.

The result usually consists of several weakly connected components (pieces), therefore we compute weakly connected components and draw network according to the components:

Draw/Draw-Partition.

Afterwards we can export pictures to SVG using Export/SVG/Line Values/Nested Classes.

# Islands

In a network where some properties (values) of vertices or lines are known we can find *islands*. Islands are called *vertex islands* if values of vertices are given, and they are called *line islands* if values of lines are given.

Lets take the network obtained from two mode network. As we know, in such network values of lines are given, therefore we can find line islands – clusters of vertices, connected with lines having higher values than values of lines going out (inside islands line values are higher than between islands).

We must also select the smallest and the largest size of island allowed. In Pajek line islands are computed using

Net/Partitions/Islands/Line Weights

Example: Davis.net – compute line islands of size 2 to 6 for both networks obtained from a two mode network

# Blockmodeling

Blockmodeling can be applied to 2-mode networks as well. We must specify number of vertices in first and in second mode. Example Davis.net.

# Examples

- Analyse and draw two-mode network Davis.net.

- Analyse and draw network with values on lines revije.net (number of readers of magazines), which was obtained from two-mode network. Use different normalizations and export pictures to SVG.

- Compute line islands for networks Davis.net and revije.net.