National Research University Higher School of Economics, Faculty of Social Sciences

Introduction to Network Analysis, Spring 2019

Seminar 4, 08/02/2019

Dmitry Zaytsev, PhD and Valentina Kuskova, PhD

(with deep appreciation to all used sources; references available in text and upon request)

Welcome to the fourth seminar! As we are getting more comfortable in SNA programming, we are starting to get into more and more complex analysis. Please make sure to complete all assignments, or you will start falling behind. As usual, before attempting to run any code, please make sure you have every necessary package installed and pay attention not to have conflicting packages loaded at once. Package *igraph* is notorious for conflicing with *network* and emph{sna.}

Contents of today's seminar:

- In your Seminar 4 folder, you will find the following files:
  1. This file, "Seminar 4," in .pdf format.
  2. Archived SouthernWomen.zip file and archived HiTech.zip file. You will need to extract and load both folders with all the files in them into your working directory. So your working directory will contain not the individual files, but two folders with files.
- For today's assignment, you will need the following packages (remember, R is case sensitive, make sure, when installing packages, to keep the appropriate case:
  1. rmarkdown
  2. RColorBrewer
  3. network
  4. sna
  5. igraph. *Please note:* this package may conflict with "network" package, especially on graphics. Therefore, when using one, it is always wise to unload the other (using command "detach(package:packagename)", or you may get an error in execution.

  Remember that to use a certain package, you need to make sure that the library is installed (using install.packages("packagename") command). To use a loaded package, you call it with "library.(packagename)" command.
- After completing today's seminar, you should be able to accomplish the following:
  1. Learn how to work with affiliation networks.
  2. Acquire additional skills for network graphs.
  3. Compare and interpret obtained network results. That's right, you are already network analysts!

Seminar 4 assignment. Please reproduce the R code in this document and answer the questions that are marked as *Assignment questions.* Please include your answers right after each question in your RMarkdown file. You should submit both the .Rmd and the .pdf files with results.

Your assignment is due on Friday, February 15, at 23.59.

Loading data folders and creating several working directories within one

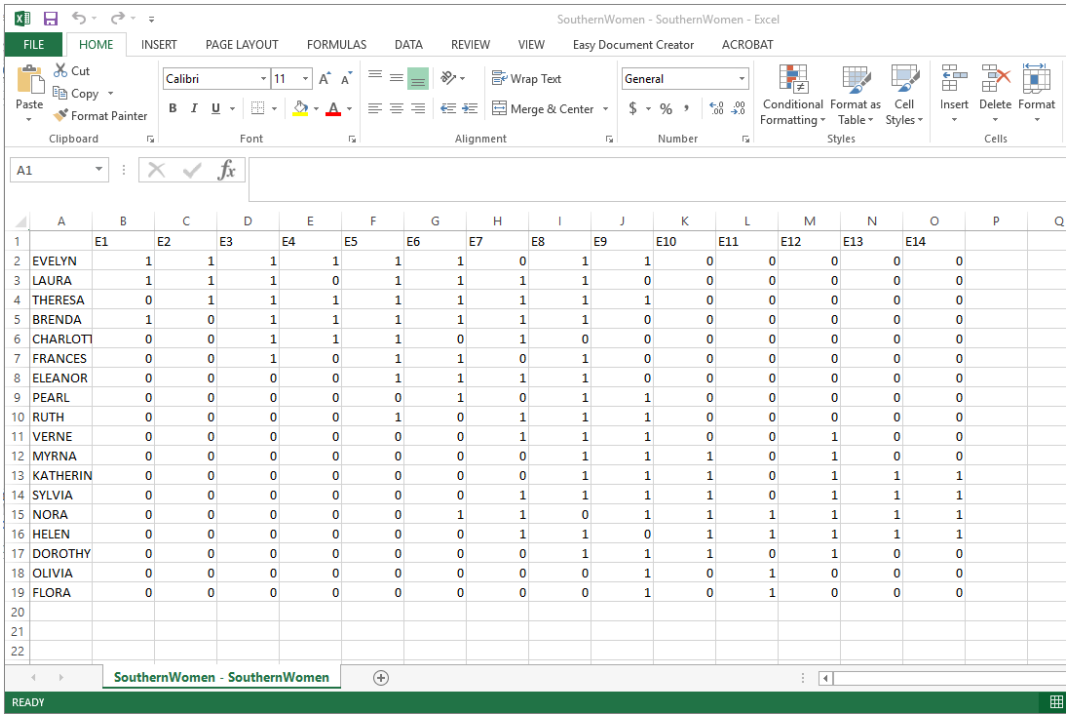
Please make sure your working directory is set to a folder where the data files are located; otherwise, you will be forced to provide a full path to data every time.

In seminar 2, we learned how to load several different types of data. What if we have several data files, located in several folders, and we need to switch between these folders all the time? Changing working directory every time is not very practical; typing in full file path is time-consuming. It turns out, we can load a path name to a folder we need as an object within R, and then just use the needed object to call a specific file from a specific folder.

First, set your working directory to what you need it to be (and where your unzipped data folders are located). Then, feed it to an object:

```
wd<-getwd()
```

We will be working with the file, SouthernWomen.tsv, for this assignment. It is a well-known classical dataset of women attending different events. In the folder "SouthernWomen" you will find two files, R workspace data file (we won't be using it for now, but you have to load it using File-Open File command), and the ".tsv" file, which is new to us. The file extension, ".tsv," stands for "tab-separated file," and you can open it with Excel to see what's inside.



		E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14
2	EVELYN	1	1	1	1	1	1	0	1	1	0	0	0	0	0
3	LAURA	1	1	1	0	1	1	1	1	0	0	0	0	0	0
4	THERESA	0	1	1	1	1	1	1	1	1	0	0	0	0	0
5	BRENDA	1	0	1	1	1	1	1	1	0	0	0	0	0	0
6	CHARLOTI	0	0	1	1	1	0	1	0	0	0	0	0	0	0
7	FRANCES	0	0	1	0	1	1	0	1	0	0	0	0	0	0
8	ELEANOR	0	0	0	0	1	1	1	1	0	0	0	0	0	0
9	PEARL	0	0	0	0	0	1	0	1	1	0	0	0	0	0
10	RUTH	0	0	0	0	1	0	1	1	1	0	0	0	0	0
11	VERNE	0	0	0	0	0	0	1	1	1	0	0	1	0	0
12	MYRNA	0	0	0	0	0	0	0	1	1	1	0	1	0	0
13	KATHERIN	0	0	0	0	0	0	0	1	1	1	0	1	1	1
14	SYLVIA	0	0	0	0	0	0	1	1	1	1	0	1	1	1
15	NORA	0	0	0	0	0	1	1	0	1	1	1	1	1	1
16	HELEN	0	0	0	0	0	0	1	1	0	1	1	1	1	1
17	DOROTHY	0	0	0	0	0	0	0	1	1	1	0	1	0	0
18	OLIVIA	0	0	0	0	0	0	0	0	1	0	1	0	0	0
19	FLORA	0	0	0	0	0	0	0	0	1	0	1	0	0	0

What we see is a matrix of data, but not the symmetric adjacency matrix that we are used to - it's an *incidence* matrix for a bi-modal, or an affiliation, network. So rows are women, and columns - events that women attended; values in cells indicate *incidence* of the arc. This is what we will be workingw with today, but first, let's load it into R.

We need to tell R where to look for the data. In our case, it is the folder named "SouthernWomen;" the file we need is the "Southernwomen.tsv" file. Let's load *its name* into an object and create another object that will join together two folder. We will have our working directory and the directory for the file:

```
SWFile <- "/SouthernWomen/SouthernWomen.tsv" #this is the name of the file

#Let's link two objects, the working directory and the name of the file, to make the full file path:
SWFilePath <- paste(wd, SWFile, sep = "")
```

Now, when we need file from our working directory, we'll call it from there; when we need our SouthernWomen dataset, we'll call it from SWFilePath. Next, let's load the data into R; to read file in the .tsv format, you'll need command "read.table:"

```
# Option sep indicates separater, we have a header, and names of rows are in the first column:
SWrawdata<-read.table(SWFilePath, sep = "\t", header = TRUE, row.names = 1)
SWrawdata
```

```
##           E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14
## EVELYN     1  1  1  1  1  1  0  1  1   0   0   0   0   0
## LAURA      1  1  1  0  1  1  1  1  0   0   0   0   0   0
## THERESA    0  1  1  1  1  1  1  1  1   0   0   0   0   0
## BRENDA     1  0  1  1  1  1  1  1  0   0   0   0   0   0
## CHARLOTTE  0  0  1  1  1  0  1  0  0   0   0   0   0   0
## FRANCES    0  0  1  0  1  1  0  1  0   0   0   0   0   0
## ELEANOR    0  0  0  0  1  1  1  1  0   0   0   0   0   0
## PEARL      0  0  0  0  0  1  0  1  1   0   0   0   0   0
## RUTH       0  0  0  0  1  0  1  1  1   0   0   0   0   0
## VERNE      0  0  0  0  0  0  1  1  1   0   0   1   0   0
## MYRNA      0  0  0  0  0  0  0  1  1   1   0   1   0   0
## KATHERINE  0  0  0  0  0  0  0  1  1   1   0   1   1   1
## SYLVIA     0  0  0  0  0  0  1  1  1   1   0   1   1   1
## NORA       0  0  0  0  0  1  1  0  1   1   1   1   1   1
## HELEN      0  0  0  0  0  0  1  1  0   1   1   1   1   1
## DOROTHY    0  0  0  0  0  0  0  1  1   1   0   1   0   0
## OLIVIA     0  0  0  0  0  0  0  0  1   0   1   0   0   0
## FLORA      0  0  0  0  0  0  0  0  1   0   1   0   0   0
```

Assignment task. In your Seminar 4 folder, there is another file with data, HiTech. Create the code that will do the following:

- Create a directory that will consist of a working directory name and the HiTech folder name, so that it opens the HiTech folder.
- Create file paths to separate data files "GivesAdviceTo," "IsFriendOf," "ReportsTo," and "HiTechAtt.""

Plotting affiliation data

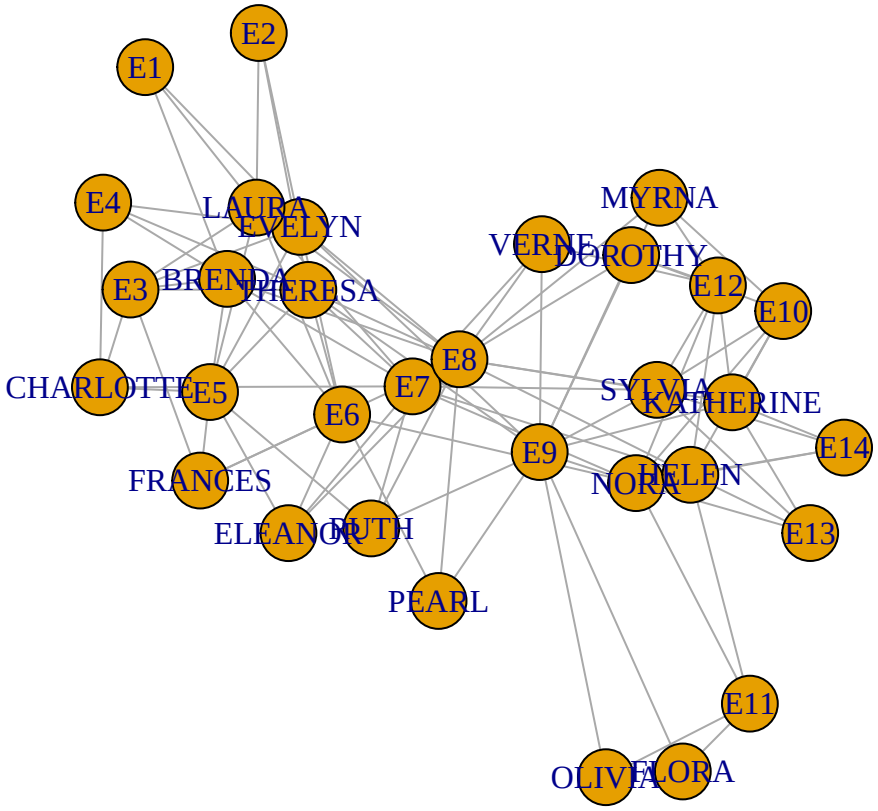
The matrix for SouthernWomen above is the incidence matrix, and to convert it to a network (in this case, a bimodal network) we'll be using an *igraph* native command, "graph_from_incidence_matrix." So, load an the required *igraph* package and execute the command:

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
SWnet<-graph_from_incidence_matrix(SWrawdata)

# Of course, the first thing we do with our data is look at it, so
par(mar=c(0,0,0,0))
plot(SWnet)
```

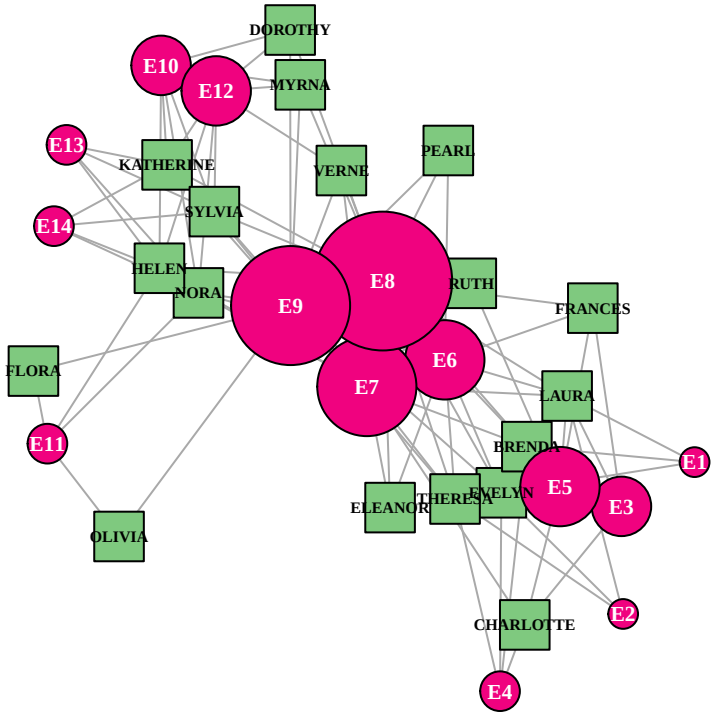


Now, that's ugly. We know that there are two different types of nodes - women and events, but in this graph, it's difficult to distinguish between the two. We suggest we modify the graph a bit, so that it looks a little more presentable. Since you are well familiar with plotting techniques, we are not providing too many details on the code below.

We do, however, want to mention one thing. It is obvious that the graph above does *NOT* contain arrows - that is, network appears undirected. But... someone might say, it should be directed! Women go *to* parties! Well, though it appears directed theoretically, if you think about it, it's the same logic as with married couples - if a women went *to* a party, it automatically means that the party contained that woman. Moreover, when the direction is that obvious, it makes no sense to draw arrows - they will only make the graph too busy.

Some code is commented out; it is for you to understand better what we are doing with the graph. Those of you who skipped ahead and learned a lot about igraph options, do not need to run this code.

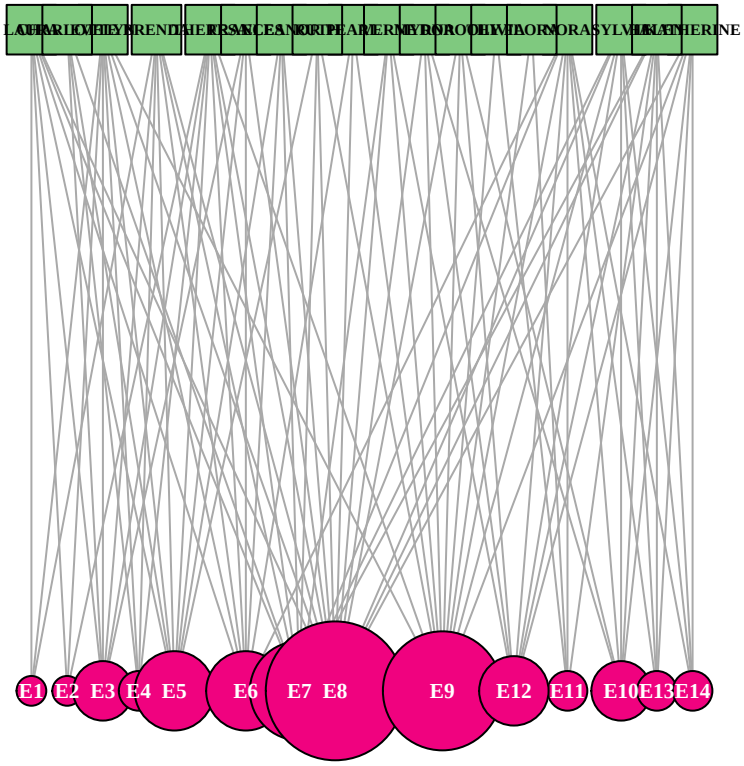
```
## V(SWnet)$type
## V(SWnet)$name

library(RColorBrewer)
colors<-brewer.pal(8, 'Accent') #select and load a palette

# Next, change attributes we want.
# Doing it one at a time, as we are sure you've figured out,
# is much easier than inside the plot function
V(SWnet)$color <- c(colors[1],colors[6])[V(SWnet)$type+1] #set two different colors for two node types
V(SWnet)$shape <- c("square", "circle")[V(SWnet)$type+1] #change shapes based on node types
V(SWnet)$label.color<-c("black","white")[V(SWnet)$type+1]
V(SWnet)$label.cex<-c(0.5, 0.7)[V(SWnet)$type+1]
V(SWnet)$label.font=2

# Calculate the indegree of events:
V(SWnet)$indegree <- degree(SWnet, mode = "in") #calculate indegree

#Set the size of the event node scaled by indegree (*3 for scale), leave women all same size:
V(SWnet)$size<-ifelse(V(SWnet)$type==TRUE,V(SWnet)$indegree*3,15)

# Plot the graph
par(mar=c(0,0,0,0))
plot(SWnet)
```



Hmmm.... not sure we like the result. There is another option that we would like to show you, the bipartite

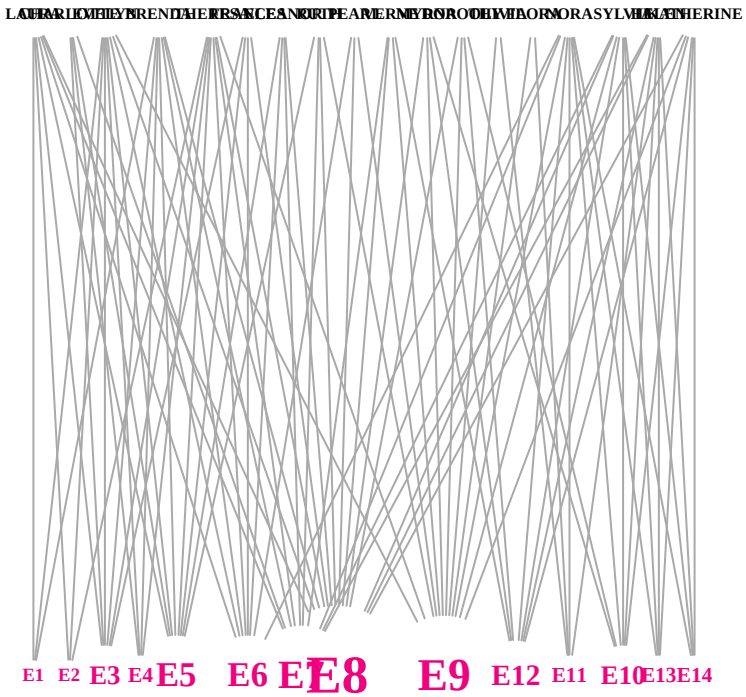
option:

```
par(mar=c(0,0,0,0))
plot(SWnet, layout=layout.bipartite)
```



OK, this clearly separates the nodes, but the result is not much better. So I'll experiment some more:

```
V(SWnet)$shape='none'
V(SWnet)$label.color<-c("black",colors[6])[V(SWnet)$type+1]
V(SWnet)$label.cex<-ifelse(V(SWnet)$type==TRUE,0.25+V(SWnet)$indegree/10,0.5)

par(mar=c(0,0,0,0))
plot(SWnet, layout=layout.bipartite)
```



Assignment task. Apparently, creating such graph is not as easy as it looks! You give it a try. Using the plotting function in *igraph*, improve my graph for the SouthernWomen data by changing at least three characteristics of the graph.

Finally, we obviously understand that some women went to the same parties, and some of the parties contained some of the women. Women who went to the same parties are connected to each other (think of it as them having a chance to meet), and the events are connected to each other (think of it as advertisements for one event was distributed at another and available to women who went to it). So, from the bipartite network we can create two adjacency matrices, one for women and one for events. This is generally not recommended, because we lose information; however, if you save all your data, you should be just fine.

You can do this one of two ways. First, you can project a two-mode matrix into a one-mode by multiplying the matrix by its transpose. Obviously, we will be working with the raw data file that we pulled earlier (this is not always easier than attempting to convert networks back to data, but there is also absolutely no reason to do so - R keeps your old objects in memory:

```
Women.only<-as.matrix(SWrawdata)%*%t(as.matrix(SWrawdata)) # Matrix of women only
Events.only<-t(as.matrix(SWrawdata))%*%as.matrix(SWrawdata) # Matrix of events only
```

Assignment task. Display the contents of the Women.only and Events.only matrices we created above.

- Explain what data in these matrices mean.
- What are the benefits and problems with separating incidence matrices and creating adjacency matrices

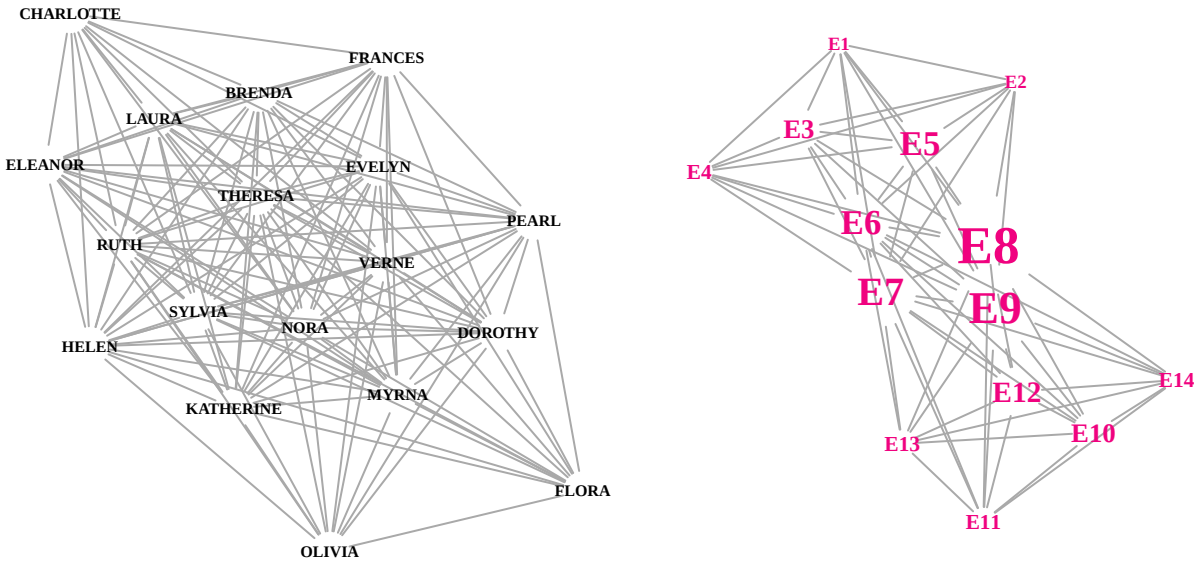
out of them?

There is, of course, more than one way of doing something. You can separate graphs into two using the built-in *igraph* command, "bipartite.projection." It allows you to separate the two graphs and graphs them using either the default, or the options you have set in earlier plotting:

```
SWnet.sep<-bipartite.projection(SWnet) # separate the two networks
```

Now, because it's a projection, inside the SWnet.sep we now have two lists, or projections. By default, system calls them proj1 and proj2. So, let's plot them:

```
par(mar=c(0,0,0,0), mfrow=c(1,2))
plot(SWnet.sep$proj1)
plot(SWnet.sep$proj2)
```



Notice that the system, of course, has kept all the changes I've made to the plotting parameters, and the resulting networks are different from the default with respect to all graph attributes.

Assignment task. For the network SouthernWomen, please calculate the following network characteristics and briefl explain what they mean:

- Indegree, outdegree, total degree;
- Centrality: degree, betweenness, closeness, eigenvector, page rank; correlations between these measures.
- Transitivity.
