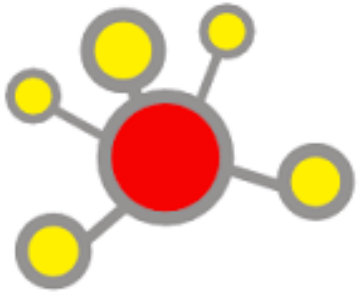
Igraph Tutorial

python™



Some good sources of graph data

- http://www-personal.umich.edu/~mejn/netdata/
- http://snap.stanford.edu/data/
- https://networkdata.ics.uci.edu/resources.php
- http://konect.uni-koblenz.de/

- And tons more. Go information age!

Reading Graphs in From Files

- To load a formatted graph file into igraph

- In R: graph <- read.graph("[PATH]/mygraph.gml", "gml")
- In Python: graph = Graph.Read("[PATH]/mygraph.gml", "gml")

- Be sure to specify the correct format for your graph
- For some formats it is also a good idea to specify directed=false for undirected graphs. (or direction will be assumed on edges)

Exporting Graphs to Files

- To export a graph object to a formatted file

- In R: write.graph(graph, “[PATH]/mygraph.gml", "gml")
- In Python: Graph.write(graph, “[PATH]/mygraph.gml“, "gml")

- Note: Yes, in Python the read function is capitalized and write is not.
  - Although there are now convenience functions with and without capitalization

Network File Formats

- edgelist: pairs of vertex IDs that have an edge
- pajek: text format where each element is in a line followed by edges.
- graphml: XML based file format
- gml: simple textual format
- ncol: weighted edgelist used for large graphs
- lgl: large graph layout
- dimacs: mainly used for network flow
- graphdb: binary graph database format
- Details: http://igraph.org/r/doc/read_graph.html

An Example: word connection graph

- Example from http://www.r-bloggers.com/an-example-of-social-network-analysis-with-r-using-package-igraph/
- Term document matrix
  - Each word is a row
  - Each document is a column
  - Nonzeroes indicate the word is in the document.
  - Value can be binary or a count
- We're making a graph of this matrix.
  - A useful visualization of word connectedness
  - Approach could be used generally with other types of (sparse) matrices

Matrix Formatting

```
# load termDocMatrix
load("termDocMatrix.rdata")

# inspect part of the matrix
termDocMatrix[5:10,1:20]
```

Matrix Formatting

```
# change it to a Boolean matrix
termDocMatrix[termDocMatrix>=1] <- 1

# transform into a term-term adjacency matrix
#Note:(%*% = matrix product) (for python users, use numpy.dot)
termMatrix <- termDocMatrix %*% t(termDocMatrix)

# inspect terms numbered 5 to 10
termMatrix[5:10,5:10]
```

Building a graph

```
library(igraph)

# build an undirected, weighted graph from the above matrix
g <- graph.adjacency(termMatrix, weighted=T, mode = 'undirected')

print(g)

#alternative: read graph
g = read.graph("TermNetwork.gml","gml")
```

If you are using Python

```
from igraph import *

g =Graph.Read(f="TermNetwork.gml", format="gml")

print(summary(g))
```

Exercises

remove self loops

set labels (to the name attribute) and degrees of vertices

Layouts

- Too many to list (http://igraph.org/c/doc/igraph-Layout.html#idm470928403040)

- igraph_layout_random – uniformly at random on a plane
- igraph_layout_grid – in a grid
- igraph_layout_circle – nodes organized in a circle by ID
- igraph_layout_bipartite – standard for bipartite graphs
- igraph_layout_fruchterman_reingold – FDL, flexible and usually attractive
- igraph_layout_kamada_kawai – also FDL, usually faster and messier than FR
- igraph_layout_lgl – for large graphs

Exercises

set seed to make the layout reproducible

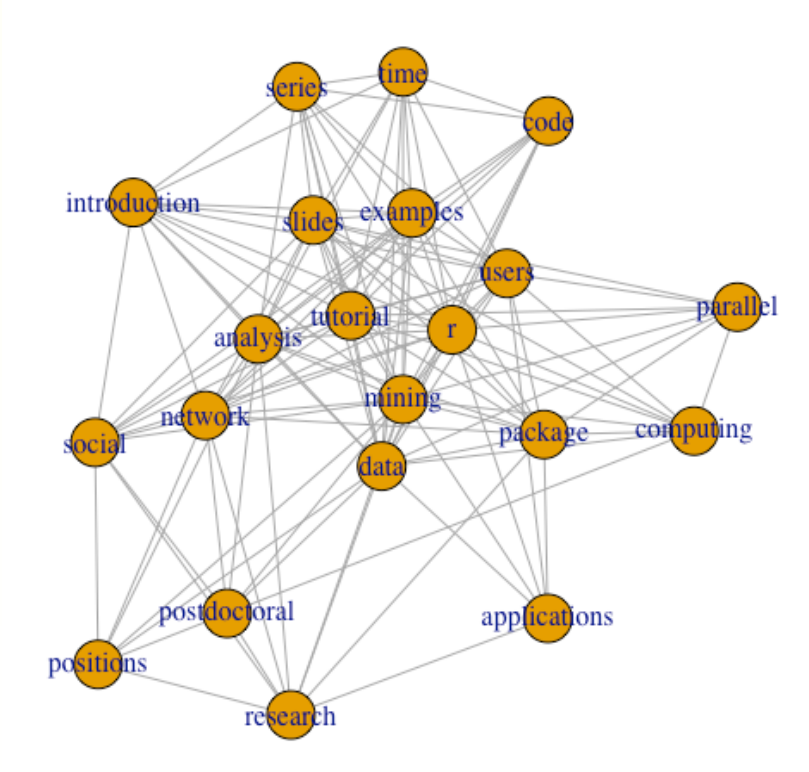
#make fruchterman reingold layout

#make kamada kawai layout

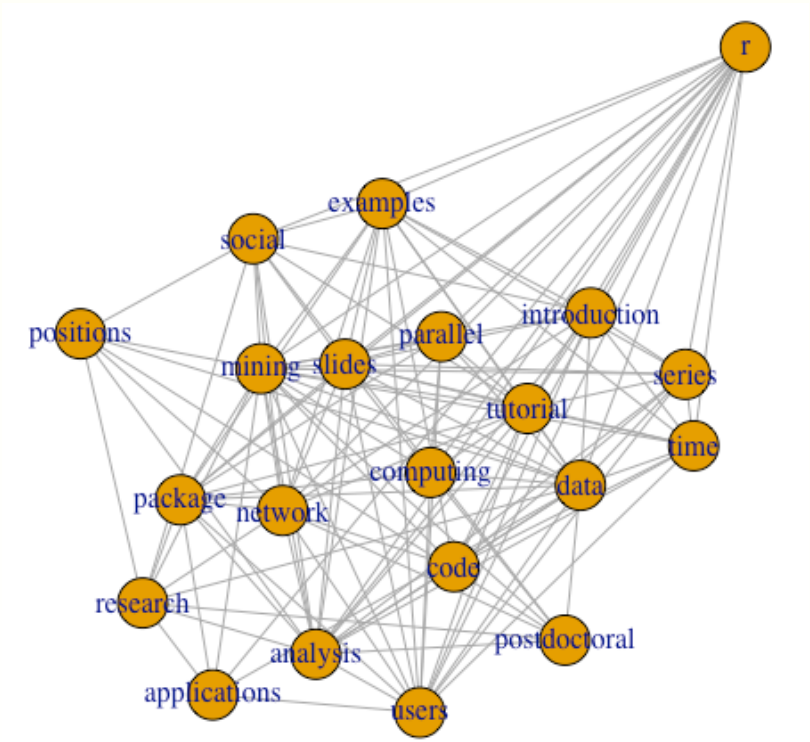
#plot layouts 1 and 2

Exercises

Layouts:



layout.fruchterman.reingold



layout.kamada.kawai

Exercises

#set vertex label size (label.cex) proportional to degree

#change label color using rgb()

#change vertex color using rgb()

#remove outlines (frame.color)

Exercises

#given, calculate edge thicknesses based on weight

#set edge color

#set edge widths to calculated values

plot the graph in layout1, note: tkplot lets you drag the nodes, handy

Structural Properties

#find if the network is connected

#get the network diameter

#get the maximum degree

#plot the network's (cumulative) degree distribution