part2-simplifying

Matteo Sanchez-Dahl, Latifa Al Jlayel

2022-09-21

# R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

## Including Plots

You can also embed plots, for example:

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

library(igraph)

##   
## Attaching package: 'igraph'

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

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

library(sna)

## Loading required package: statnet.common

##   
## Attaching package: 'statnet.common'

## The following objects are masked from 'package:base':  
##   
## attr, order

## Loading required package: network

##   
## 'network' 1.17.2 (2022-05-20), part of the Statnet Project  
## \* 'news(package="network")' for changes since last version  
## \* 'citation("network")' for citation information  
## \* 'https://statnet.org' for help, support, and other information

##   
## Attaching package: 'network'

## The following objects are masked from 'package:igraph':  
##   
## %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,  
## get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,  
## is.directed, list.edge.attributes, list.vertex.attributes,  
## set.edge.attribute, set.vertex.attribute

## sna: Tools for Social Network Analysis  
## Version 2.7 created on 2022-05-09.  
## copyright (c) 2005, Carter T. Butts, University of California-Irvine  
## For citation information, type citation("sna").  
## Type help(package="sna") to get started.

##   
## Attaching package: 'sna'

## The following objects are masked from 'package:igraph':  
##   
## betweenness, bonpow, closeness, components, degree, dyad.census,  
## evcent, hierarchy, is.connected, neighborhood, triad.census

nodes <- as.matrix(read.table('data\_subelj\_jdk/ent\_subelj\_jdk\_jdk\_class\_name.txt'))  
edges <- as.matrix(read.table('data\_subelj\_jdk/out\_subelj\_jdk\_jdk.txt', sep = "", header = F, skip = 2))  
edges

## V1 V2  
## [1,] 1 2  
## [2,] 1 3  
## [3,] 1 4  
## [4,] 1 5  
## [5,] 1 5  
## [6,] 1 6  
## [7,] 1 4  
## [8,] 1 5  
## [9,] 1 4  
....

net <- graph\_from\_edgelist(edges, directed = TRUE)  
V(net)$name <- nodes  
net

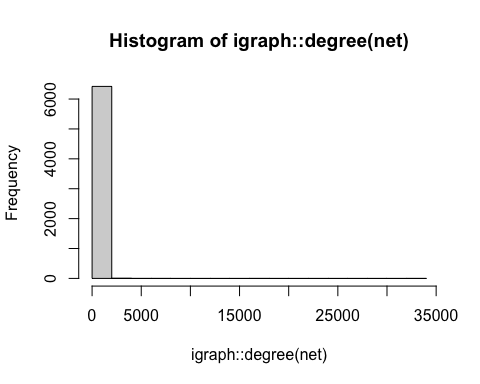
## IGRAPH bc27a41 DN-- 6434 150985 --   
## + attr: name (v/c)  
## + edges from bc27a41 (vertex names):  
## [1] java.applet.Applet->java.awt.Panel   
## [2] java.applet.Applet->java.awt.Dimension  
## [3] java.applet.Applet->java.net.URL   
## [4] java.applet.Applet->java.lang.String   
## [5] java.applet.Applet->java.lang.String   
## [6] java.applet.Applet->java.util.Locale   
## [7] java.applet.Applet->java.net.URL   
## [8] java.applet.Applet->java.lang.String   
## + ... omitted several edges

plot(net)



In this histogram, 6423 nodes fall into the first bucket. That means they have degree less than 2000. From this histogram we can tell that the nodes in the network differ greatly in terms of their degree. From documentation about the data set (<http://konect.cc/networks/subelj_jdk/>), we know that the the node with the maximum degree in the network has a degree (dmax) of 32,530. It’s difficult to get a good sense of the data set will all the nodes, so we want to explore sub-graphs of the network to further elucidate which nodes to focus our analysis on. We must choose the handful of nodes in the network that we can analyze such that those nodes retain the most important information in the data set and we can begin to make generalizations about the entire data set from the results of analyzing these nodes.

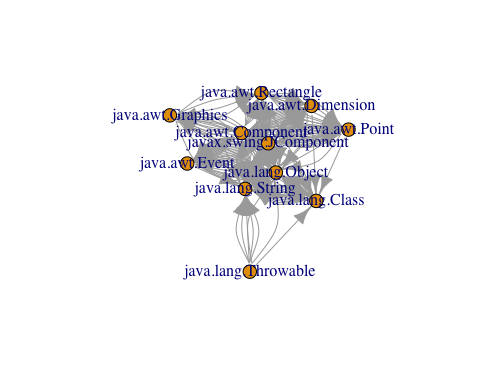
histogram <- hist(igraph::degree(net))



Here we make a subgraph of the nodes that have a degree higher than 2000. There are 11 nodes that meet this criteria. We can conclude these nodes are the most important nodes in the software class dependency network of the JDK 1.6.0.7 framework since they have the highest degrees in the entire network, i.e. many classes depend on them since an edge between nodes indicates there exists a dependency between the two classes. This makes sense since a String class is needed to compile the source code into executable files with binary machine code and Java is an OOP language, which means the Object class is key to its functionality.

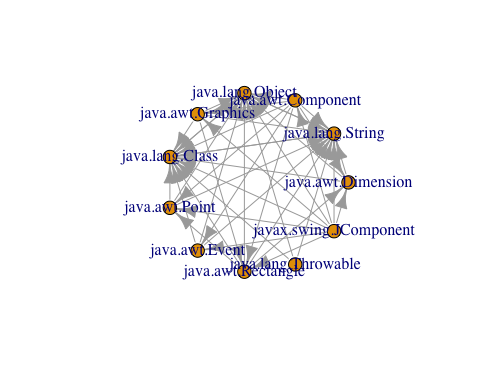
EDIT: We can conclude these 4 nodes are some of the most important nodes in the software class dependency network of the JDK 1.6.0.7 framework since they remain after simplifying the graph with the weights of the vertices. They also have some of the highest degree centrality. This makes sense since a String class is needed to compile the source code into executable files with binary machine code and Java is an OOP language, which means the Object class is key to its functionality.

sg2000 <- induced\_subgraph(net, igraph::V(net)[igraph::degree(net)>2000])  
plot(sg2000)



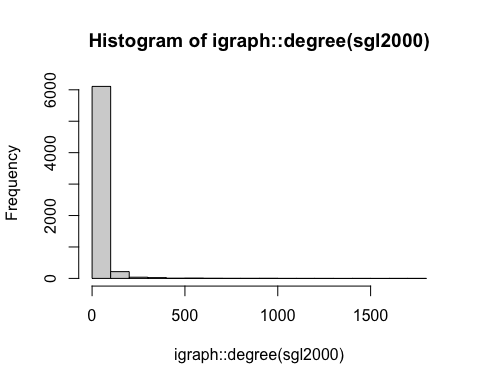
Here we use the simplify( ) function to create a new attribute by summing the old ones. We plot the resulting graph with the circle layout to better visualize how these core 11 nodes are connected to each out.

net\_2000 = simplify(sg2000,  
 remove.multiple = TRUE,  
 edge.attr.comb = "sum"  
 )  
l = layout\_in\_circle(net\_2000)  
plot(net\_2000, layout = l)



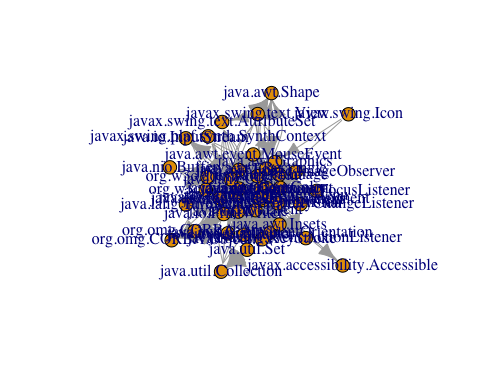
Here we make a subgraph of the 6423 nodes that have a degree less than 2000 to better investigate what the network looks like without the 11 highest degree nodes. Form the histogram, we can observe that the majority of the nodes (6108) have less than 100 degrees, 6323 nodes have a degree less than 200, and 6390 nodes have degree lass than 500.

sgl2000 <- induced\_subgraph(net, igraph::V(net)[igraph::degree(net)<2000])  
histl200 <- hist(igraph::degree(sgl2000))



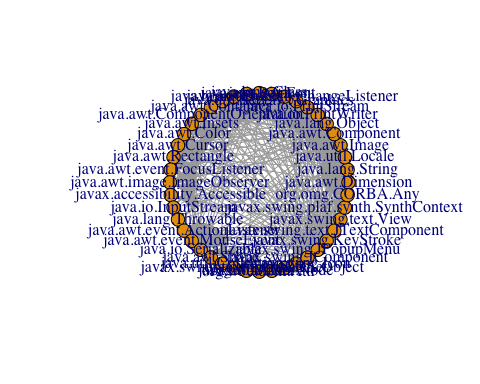
Here we make a subgraph of all the nodes in the network that have a degree greater than 500. 44 nodes in the network meet this criteria.

sg500 <- induced\_subgraph(net, igraph::V(net)[igraph::degree(net)>500])  
plot(sg500)

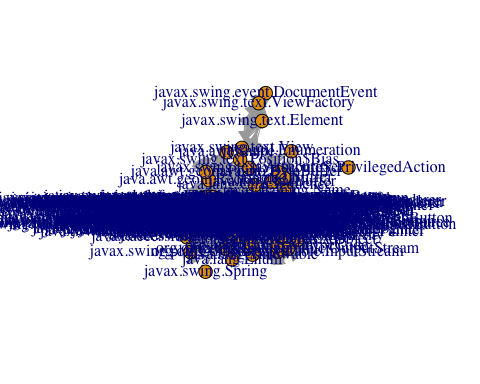


Here we use the simplify( ) function to create a new attribute by summing the old ones. We plot the resulting graph with the circle layout to better visualize how these core 44 nodes are connected to each out.

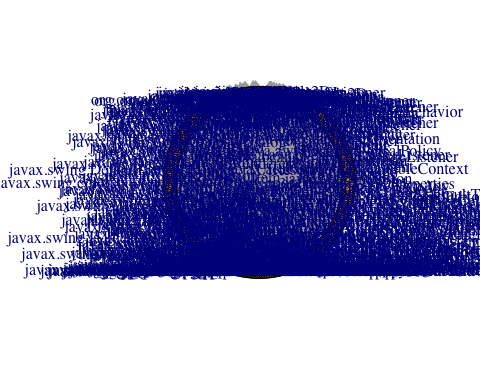
net\_simple = simplify(sg500,  
 remove.multiple = TRUE,  
 edge.attr.comb = "sum"  
 )  
l <- layout\_in\_circle(net\_simple)  
plot(net\_simple, layout = l)



sg200 <- induced\_subgraph(net, igraph::V(net)[igraph::degree(net)>200])  
plot(sg200)

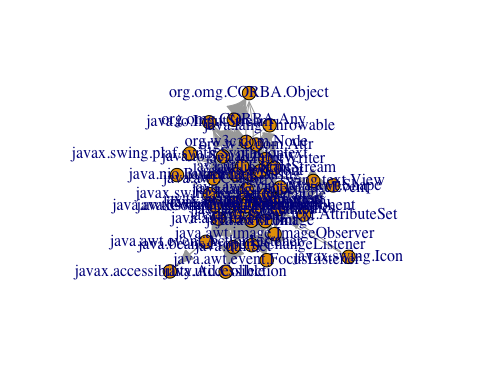


l <- layout\_in\_circle(sg200)  
plot(sg200, layout = l)



final simplified data used for subsequent section

plot(sg500)



l <- layout\_in\_circle(sg500)  
plot(sg500, layout = l)

