

Chapter 3

Nick Lauerman

13 February 2020

Contents

Libraries and data used	1
Libraries	1
Data	4
Creating and managing networks	4
creating net in netstat	4
Managing Node and Tie Attributes	7
Node attrib	7
Tie Attrib	8
Creating network object in igraph	9
Going back and forth between statman and igraph	10
importing network data	10
Common network data tasks	13
Filtering networks daved on vertex or edge attribute values	13
Filtering based on Node Values	13
remove isolates	15
filtering based on edge values	16
transforming directed to non-directed network	21

Libraries and data used

Libraries

```
library(UserNetR)
library(statnet)

## Loading required package: tergm
## Loading required package: ergm
## Loading required package: network

## network: Classes for Relational Data
## Version 1.16.0 created on 2019-11-30.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##           Mark S. Handcock, University of California -- Los Angeles
##           David R. Hunter, Penn State University
##           Martina Morris, University of Washington
##           Skye Bender-deMoll, University of Washington
```

```

## For citation information, type citation("network").
## Type help("network-package") to get started.

##
## ergm: version 3.10.4, created on 2019-06-10
## Copyright (c) 2019, Mark S. Handcock, University of California -- Los Angeles
##           David R. Hunter, Penn State University
##           Carter T. Butts, University of California -- Irvine
##           Steven M. Goodreau, University of Washington
##           Pavel N. Krivitsky, University of Wollongong
##           Martina Morris, University of Washington
##           with contributions from
##           Li Wang
##           Kirk Li, University of Washington
##           Skye Bender-deMoll, University of Washington
##           Chad Klumb
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergm").

## NOTE: Versions before 3.6.1 had a bug in the implementation of the bd()
## constraint which distorted the sampled distribution somewhat. In
## addition, Sampson's Monks datasets had mislabeled vertices. See the
## NEWS and the documentation for more details.

## NOTE: Some common term arguments pertaining to vertex attribute and
## level selection have changed in 3.10.0. See terms help for more
## details. Use 'options(ergm.term=list(version="3.9.4"))' to use old
## behavior.

## Loading required package: networkDynamic

##
## networkDynamic: version 0.10.1, created on 2020-01-16
## Copyright (c) 2020, Carter T. Butts, University of California -- Irvine
##           Ayn Leslie-Cook, University of Washington
##           Pavel N. Krivitsky, University of Wollongong
##           Skye Bender-deMoll, University of Washington
##           with contributions from
##           Zack Almquist, University of California -- Irvine
##           David R. Hunter, Penn State University
##           Li Wang
##           Kirk Li, University of Washington
##           Steven M. Goodreau, University of Washington
##           Jeffrey Horner
##           Martina Morris, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("networkDynamic").

##
## tergm: version 3.6.1, created on 2019-06-12
## Copyright (c) 2019, Pavel N. Krivitsky, University of Wollongong
##           Mark S. Handcock, University of California -- Los Angeles
##           with contributions from
##           David R. Hunter, Penn State University
##           Steven M. Goodreau, University of Washington

```

```

##           Martina Morris, University of Washington
##           Nicole Bohme Carnegie, New York University
##           Carter T. Butts, University of California -- Irvine
##           Ayn Leslie-Cook, University of Washington
##           Skye Bender-deMoll
##           Li Wang
##           Kirk Li, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("tergm").

## Loading required package: ergm.count

##
## ergm.count: version 3.4.0, created on 2019-05-15
## Copyright (c) 2019, Pavel N. Krivitsky, University of Wollongong
##           with contributions from
##           Mark S. Handcock, University of California -- Los Angeles
##           David R. Hunter, Penn State University
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergm.count").

## NOTE: The form of the term 'CMP' has been changed in version 3.2 of
## 'ergm.count'. See the news or help('CMP') for more information.

## Loading required package: sna

## Loading required package: statnet.common

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

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

## sna: Tools for Social Network Analysis
## Version 2.5 created on 2019-12-09.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
## For citation information, type citation("sna").
## Type help(package="sna") to get started.

## Loading required package: tsna

##
## statnet: version 2019.6, created on 2019-06-13
## Copyright (c) 2019, Mark S. Handcock, University of California -- Los Angeles
##           David R. Hunter, Penn State University
##           Carter T. Butts, University of California -- Irvine
##           Steven M. Goodreau, University of Washington
##           Pavel N. Krivitsky, University of Wollongong
##           Skye Bender-deMoll
##           Martina Morris, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("statnet").

## unable to reach CRAN

```

```
#library(igraph) will be loaded in flow to prevent interference with statnet  
library(intergraph)
```

Data

```
data("ICTS_G10")  
data("DHHS")
```

Creating and managing networks

creating net in netstat

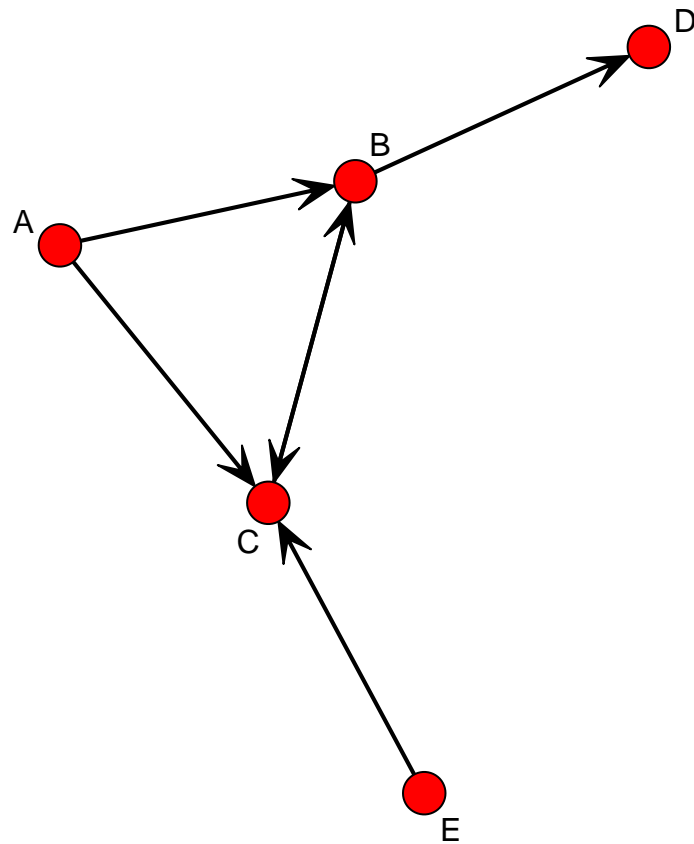
```
netmat1 <- rbind(c(0,1,1,0,0),  
                 c(0,0,1,1,0),  
                 c(0,1,0,0,0),  
                 c(0,0,0,0,0),  
                 c(0,0,1,0,0))  
rownames(netmat1) <- c("A","B","C","D","E")  
colnames(netmat1) <- c("A","B","C","D","E")  
net1 <- network(netmat1, matrix.type = "adjacency")  
class(net1)
```

```
## [1] "network"
```

```
summary(net1)
```

```
## Network attributes:  
##   vertices = 5  
##   directed = TRUE  
##   hyper = FALSE  
##   loops = FALSE  
##   multiple = FALSE  
##   bipartite = FALSE  
##   total edges = 6  
##   missing edges = 0  
##   non-missing edges = 6  
##   density = 0.3  
##  
## Vertex attributes:  
##   vertex.names:  
##     character valued attribute  
##     5 valid vertex names  
##  
## No edge attributes  
##  
## Network adjacency matrix:  
##   A B C D E  
## A 0 1 1 0 0  
## B 0 0 1 1 0  
## C 0 1 0 0 0  
## D 0 0 0 0 0  
## E 0 0 1 0 0
```

```
gplot(net1,
      vertex.col = 2,
      displaylabels = TRUE)
```



```
netmat2 <- rbind(c(1,2),
                 c(1,3),
                 c(2,3),
                 c(2,4),
                 c(3,2),
                 c(5,3))
net2 <- network(netmat2,
                 matrix.type = "edgelist")
network.vertex.names(net2) <- c("A", "B", "C", "D", "E")
summary(net2)
```

```
## Network attributes:
##   vertices = 5
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges = 6
##   missing edges = 0
##   non-missing edges = 6
##   density = 0.3
##
## Vertex attributes:
##   vertex.names:
##     character valued attribute
##     5 valid vertex names
##
## No edge attributes
##
## Network adjacency matrix:
##   A B C D E
## A 0 1 1 0 0
## B 0 0 1 1 0
## C 0 1 0 0 0
## D 0 0 0 0 0
## E 0 0 1 0 0
```

```
as.sociomatrix(net1)
```

```
##   A B C D E
## A 0 1 1 0 0
## B 0 0 1 1 0
## C 0 1 0 0 0
## D 0 0 0 0 0
## E 0 0 1 0 0
```

```
class(as.sociomatrix(net1))
```

```
## [1] "matrix"
```

```
all(as.matrix(net1) == as.sociomatrix(net1))
```

```
## [1] TRUE
```

```
as.matrix(net1, matrix.type = "edgelist")
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    2
## [3,]    1    3
## [4,]    2    3
## [5,]    5    3
## [6,]    2    4
## attr(,"n")
## [1] 5
## attr(,"vnames")
## [1] "A" "B" "C" "D" "E"
```

Managing Node and Tie Attributes

Node attrib

```
set.vertex.attribute(net1,
                     "gender",
                     c("F", "F", "M", "F", "M"))
net1 %v% "alldeg" <- degree(net1)
list.vertex.attributes(net1)

## [1] "alldeg"      "gender"      "na"          "vertex.names"
summary(net1)
```

```
## Network attributes:
##   vertices = 5
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges = 6
##   missing edges = 0
##   non-missing edges = 6
##   density = 0.3
##
## Vertex attributes:
##
##   alldeg:
##     numeric valued attribute
##     attribute summary:
##       Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1.0     1.0     2.0     2.4     4.0     4.0
##
##   gender:
##     character valued attribute
##     attribute summary:
##   F M
## 3 2
##   vertex.names:
##     character valued attribute
##     5 valid vertex names
##
## No edge attributes
##
## Network adjacency matrix:
##   A B C D E
## A 0 1 1 0 0
## B 0 0 1 1 0
## C 0 1 0 0 0
## D 0 0 0 0 0
## E 0 0 1 0 0
```

```
get.vertex.attribute(net1, "gender")
```

```
## [1] "F" "F" "M" "F" "M"
```

```
net1 %v% "alldeg"
```

```
## [1] 2 4 4 1 1
```

Tie Attrib

```
list.edge.attributes(net1)
```

```
## [1] "na"
```

```
set.edge.attribute(net1,  
  "rndval",  
  runif(network.size(net1),0,1))  
list.edge.attributes(net1)
```

```
## [1] "na"      "rndval"
```

```
summary(net1 %e% "rndval")
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## 0.3258 0.4438 0.4763 0.5315 0.6020 0.8337
```

```
summary(get.edge.attribute(net1, "rndval"))
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## 0.3258 0.4438 0.4763 0.5315 0.6020 0.8337
```

```
netval1 <- rbind(c(0,2,3,0,0),  
  c(0,0,3,1,0),  
  c(0,1,0,0,0),  
  c(0,0,0,0,0),  
  c(0,0,2,0,0))  
netval1 <- network(netval1,  
  matrix.type = "adjacency",  
  ignore.eval = FALSE,  
  names.eval = "like")  
network.vertex.names(netval1) <- c("A","B","C","D","E")  
list.edge.attributes(netval1)
```

```
## [1] "like" "na"
```

```
get.edge.attribute(netval1, "like")
```

```
## [1] 2 1 3 3 2 1
```

```
as.sociomatrix(netval1)
```

```
##      A B C D E  
## A 0 1 1 0 0  
## B 0 0 1 1 0  
## C 0 1 0 0 0  
## D 0 0 0 0 0  
## E 0 0 1 0 0
```

```
as.sociomatrix(netval1)
```

```
##      A B C D E  
## A 0 1 1 0 0  
## B 0 0 1 1 0
```



```
## C 0 1 0 0 0
## D 0 0 0 0 0
## E 0 0 1 0 0
```

Creating network object in igraph

```
detach(package:statnet, unload = TRUE)
detach(package:sna, unload = TRUE)
detach(package:tsna, unload = TRUE)
detach(package:ergm.count, unload = TRUE)
detach(package:tergm, unload = TRUE)
detach(package:networkDynamic, unload = TRUE)
detach(package:ergm, unload = TRUE)
detach(package:network, unload = TRUE)

## Warning: 'network' namespace cannot be unloaded:
## namespace 'network' is imported by 'intergraph' so cannot be unloaded

detach(package:statnet.common, unload = TRUE)
library(igraph)

##
## Attaching package: 'igraph'

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

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

inet1 <- graph.adjacency(netmat1)
class(inet1)

## [1] "igraph"

summary(inet1)

## IGRAPH 84868b6 DN-- 5 6 --
## + attr: name (v/c)

#str(inet1)
inet2 <- graph.edgelist(netmat2)
summary(inet2)

## IGRAPH 8488120 D--- 5 6 --
V(inet2)$name <- c("A","B","C","D","E")
E(inet2)$val <- c(1:6)
summary(inet2)

## IGRAPH 8488120 DN-- 5 6 --
## + attr: name (v/c), val (e/n)

#str(inet2)
```

Going back and forth between statman and igraph

```
library(intergraph)
```

```
class(net1)
```

```
## [1] "network"
```

```
netligraph <- asIgraph(net1)  
class(netligraph)
```

```
## [1] "igraph"
```

```
#str(netligraph)
```

importing network data

```
detach(package:igraph, unload = TRUE)  
library(statnet)
```

```
## Loading required package: tergm
```

```
## Loading required package: ergm
```

```
## Loading required package: network
```

```
## network: Classes for Relational Data
```

```
## Version 1.16.0 created on 2019-11-30.
```

```
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
```

```
##           Mark S. Handcock, University of California -- Los Angeles
```

```
##           David R. Hunter, Penn State University
```

```
##           Martina Morris, University of Washington
```

```
##           Skye Bender-deMoll, University of Washington
```

```
## For citation information, type citation("network").
```

```
## Type help("network-package") to get started.
```

```
##
```

```
## ergm: version 3.10.4, created on 2019-06-10
```

```
## Copyright (c) 2019, Mark S. Handcock, University of California -- Los Angeles
```

```
##           David R. Hunter, Penn State University
```

```
##           Carter T. Butts, University of California -- Irvine
```

```
##           Steven M. Goodreau, University of Washington
```

```
##           Pavel N. Krivitsky, University of Wollongong
```

```
##           Martina Morris, University of Washington
```

```
##           with contributions from
```

```
##           Li Wang
```

```
##           Kirk Li, University of Washington
```

```
##           Skye Bender-deMoll, University of Washington
```

```
##           Chad Klumb
```

```
## Based on "statnet" project software (statnet.org).
```

```
## For license and citation information see statnet.org/attribution
```

```
## or type citation("ergm").
```

```
## NOTE: Versions before 3.6.1 had a bug in the implementation of the bd()
```

```
## constriant which distorted the sampled distribution somewhat. In
```

```
## addition, Sampson's Monks datasets had mislabeled vertices. See the
```

```
## NEWS and the documentation for more details.
```

```

## NOTE: Some common term arguments pertaining to vertex attribute and
## level selection have changed in 3.10.0. See terms help for more
## details. Use 'options(ergm.term=list(version="3.9.4"))' to use old
## behavior.

## Loading required package: networkDynamic

##
## networkDynamic: version 0.10.1, created on 2020-01-16
## Copyright (c) 2020, Carter T. Butts, University of California -- Irvine
##           Ayn Leslie-Cook, University of Washington
##           Pavel N. Krivitsky, University of Wollongong
##           Skye Bender-deMoll, University of Washington
##           with contributions from
##           Zack Almquist, University of California -- Irvine
##           David R. Hunter, Penn State University
##           Li Wang
##           Kirk Li, University of Washington
##           Steven M. Goodreau, University of Washington
##           Jeffrey Horner
##           Martina Morris, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("networkDynamic").

##
## tergm: version 3.6.1, created on 2019-06-12
## Copyright (c) 2019, Pavel N. Krivitsky, University of Wollongong
##           Mark S. Handcock, University of California -- Los Angeles
##           with contributions from
##           David R. Hunter, Penn State University
##           Steven M. Goodreau, University of Washington
##           Martina Morris, University of Washington
##           Nicole Bohme Carnegie, New York University
##           Carter T. Butts, University of California -- Irvine
##           Ayn Leslie-Cook, University of Washington
##           Skye Bender-deMoll
##           Li Wang
##           Kirk Li, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("tergm").

## Loading required package: ergm.count

##
## ergm.count: version 3.4.0, created on 2019-05-15
## Copyright (c) 2019, Pavel N. Krivitsky, University of Wollongong
##           with contributions from
##           Mark S. Handcock, University of California -- Los Angeles
##           David R. Hunter, Penn State University
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergm.count").

## NOTE: The form of the term 'CMP' has been changed in version 3.2 of
## 'ergm.count'. See the news or help('CMP') for more information.

```

```

## Loading required package: sna
## Loading required package: statnet.common
##
## Attaching package: 'statnet.common'
## The following object is masked from 'package:base':
##
##     order
##
## sna: Tools for Social Network Analysis
## Version 2.5 created on 2019-12-09.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
## For citation information, type citation("sna").
## Type help(package="sna") to get started.
## Loading required package: tsna
##
## statnet: version 2019.6, created on 2019-06-13
## Copyright (c) 2019, Mark S. Handcock, University of California -- Los Angeles
##           David R. Hunter, Penn State University
##           Carter T. Butts, University of California -- Irvine
##           Steven M. Goodreau, University of Washington
##           Pavel N. Krivitsky, University of Wollongong
##           Skye Bender-deMoll
##           Martina Morris, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("statnet").
## unable to reach CRAN
netmat3 <- rbind(c("A","B"),
                c("A","C"),
                c("B","C"),
                c("B","D"),
                c("C","B"),
                c("E","C"))
net.df <- data.frame(netmat3)
net.df

##   X1 X2
## 1  A  B
## 2  A  C
## 3  B  C
## 4  B  D
## 5  C  B
## 6  E  C

write.csv(net.df,
          file = "MyData.csv",
          row.names = FALSE)
net.edge <- read.csv(file = "MyData.csv")
net_import <- network(net.edge,
                      matrix.type = "edgelist")
summary(net_import)

```

```

## Network attributes:
##   vertices = 5
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges = 6
##   missing edges = 0
##   non-missing edges = 6
##   density = 0.3
##
## Vertex attributes:
##   vertex.names:
##     character valued attribute
##     5 valid vertex names
##
## No edge attributes
##
## Network adjacency matrix:
##   A B C D E
## A 0 1 1 0 0
## B 0 0 1 1 0
## C 0 1 0 0 0
## D 0 0 0 0 0
## E 0 0 1 0 0
gden(net_import)

## [1] 0.3

```

Common network data tasks

Filtering networks daved on vertex or edge attribute values

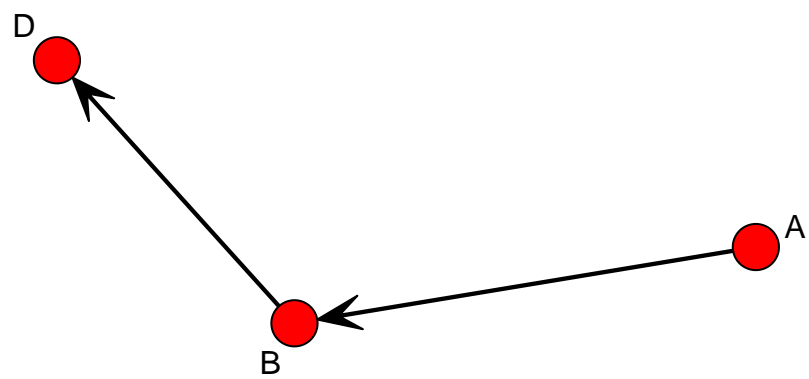
Filtering based on Node Values

```

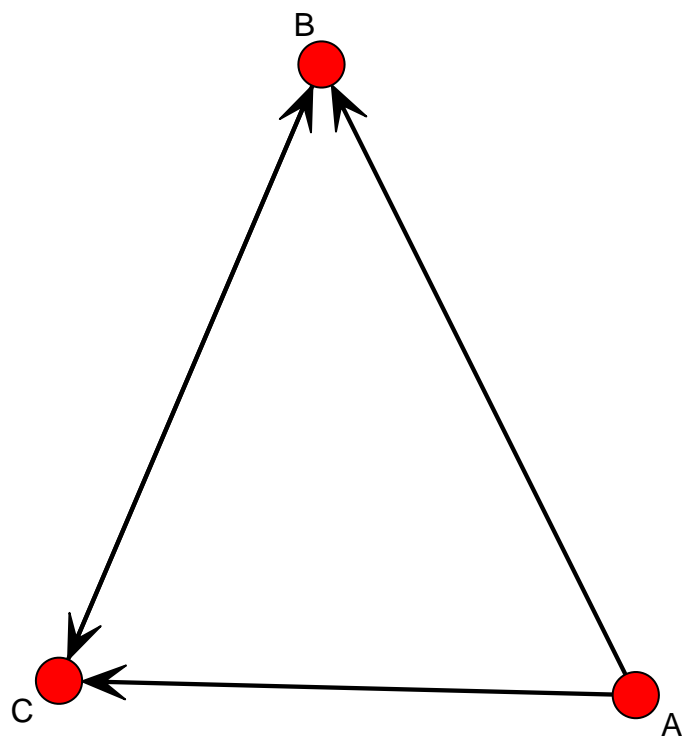
n1F <- get.inducedSubgraph(net1,
                           which(net1 %v% "gender" == "F"))
n1F[,]

##   A B D
## A 0 1 0
## B 0 0 1
## D 0 0 0
gplot(n1F, displaylabels = TRUE)

```



```
deg <- net1 %v% "alldeg"  
n2 <- get.inducedSubgraph(net1, which(deg > 1))  
  
gplot(n2, displaylabels = TRUE)
```



remove isolates

```
data("ICTS_G10")
summary(ICTS_G10, print.adj = FALSE )
```

```
## Network attributes:
##   vertices = 493
##   directed = FALSE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   title = G10_Pajek
##   total edges = 1359
```

```
## missing edges = 0
## non-missing edges = 1359
## density = 0.01120566
##
## Vertex attributes:
##
## DiscCol:
## numeric valued attribute
## attribute summary:
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 1100 3600 6500 5231 6500 8888
##
## DiscExp:
## numeric valued attribute
## attribute summary:
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 1100 2800 6540 5067 6910 8888
## vertex.names:
## character valued attribute
## 493 valid vertex names
##
## year:
## numeric valued attribute
## attribute summary:
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 2008 2008 2009 2009 2010 2010
##
## No edge attributes
```

```
gden(ICTS_G10)
```

```
## [1] 0.01120566
```

```
length(isolate(ICTS_G10))
```

```
## [1] 96
```

```
n3 <- ICTS_G10
```

```
delete.vertices(n3, isolate(n3))
```

```
gden(n3)
```

```
## [1] 0.01728876
```

```
length(isolate(n3))
```

```
## [1] 0
```

filtering based on edge values

```
data("DHHS")
```

```
summary(DHHS, print.adj = FALSE )
```

```
## Network attributes:
```

```
## vertices = 54
```

```
## directed = FALSE
```

```
## hyper = FALSE
```

```
## loops = FALSE
```

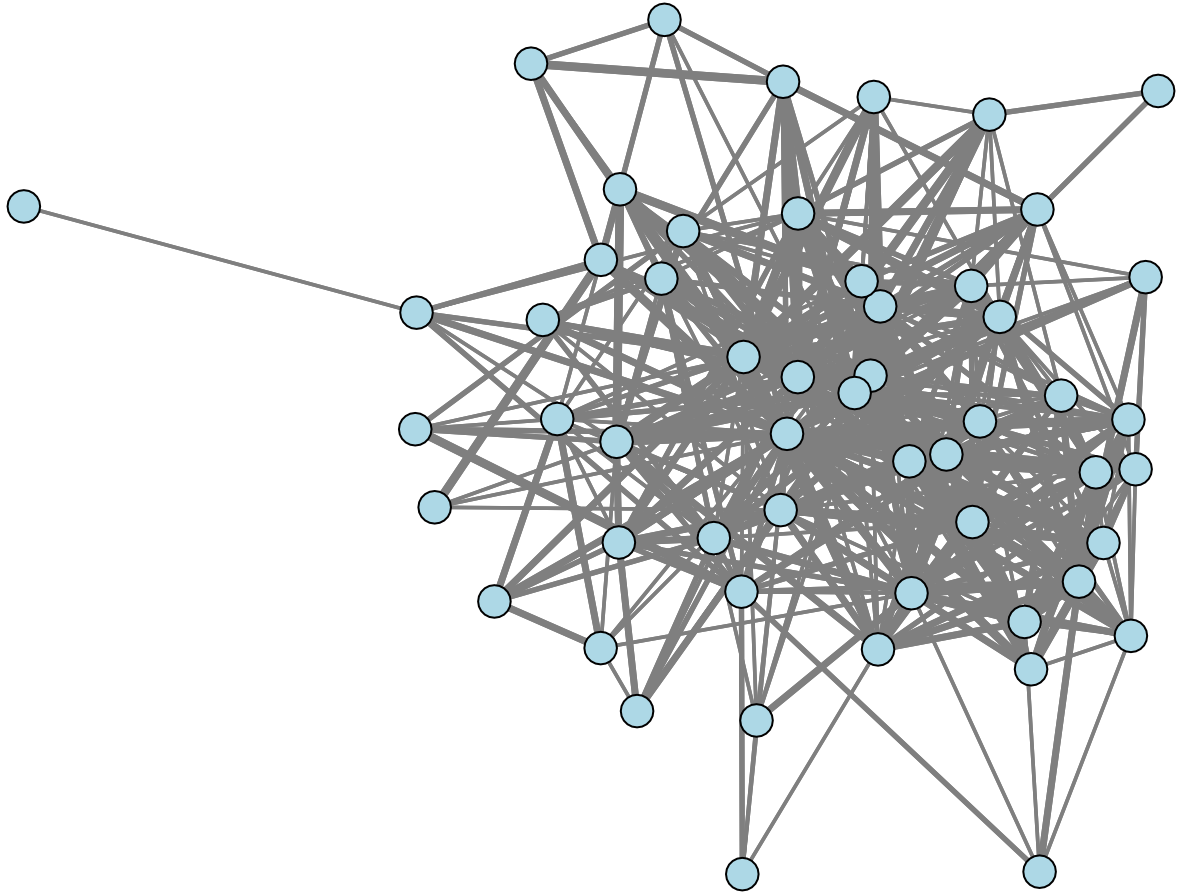


```
## multiple = FALSE
## bipartite = FALSE
## title = DHHS_Collab
## total edges = 447
## missing edges = 0
## non-missing edges = 447
## density = 0.312369
##
## Vertex attributes:
##
## agency:
## numeric valued attribute
## attribute summary:
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.000 2.000 6.500 5.185 7.000 10.000
## vertex.names:
## character valued attribute
## 54 valid vertex names
##
## Edge attributes:
##
## collab:
## numeric valued attribute
## attribute summary:
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 1.000 1.000 2.000 2.199 3.000 4.000
```

```
d <- DHHS
gden(d)
```

```
## [1] 0.312369
```

```
op <- par(mar = rep(0,4))
gplot(d,
  gmode = "graph",
  edge.lwd = d %e% 'collab',
  edge.col = "grey50",
  vertex.col = "lightblue",
  vertex.cex = 1.0,
  vertex.sides = 20)
```



```
par(op)
as.sociomatrix(d)[1:6,1:6]
```

```
##          ACF-1 ACF-2 AHRQ-1 AHRQ-2 AHRQ-3 AHRQ-4
## ACF-1      0     1     0     0     0     0
## ACF-2      1     0     0     0     0     0
## AHRQ-1      0     0     0     1     1     1
## AHRQ-2      0     0     1     0     1     1
## AHRQ-3      0     0     1     1     0     1
## AHRQ-4      0     0     1     1     1     0
```

```
table(d %e% "collab")
```

```
##
##  1  2  3  4
```

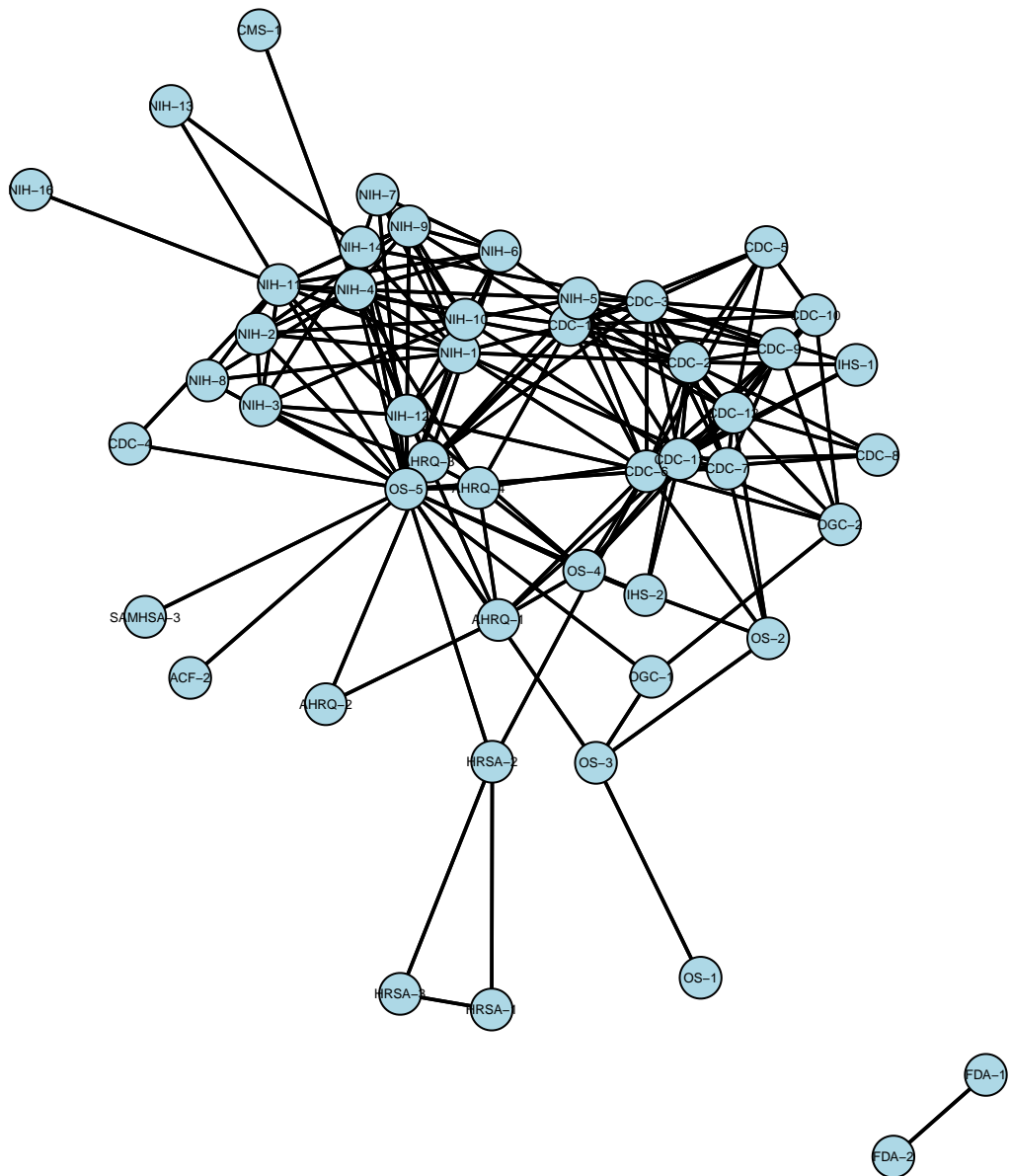
```
## 163 111 94 79
d.val <- as.sociomatrix(d, attrname = "collab")
d.val[d.val < 3] <- 0
d.filt <- as.network(d.val,
                     directed = FALSE,
                     matrix.type = "a",
                     ignore.eval = FALSE,
                     names.eval = "collab")
summary(d.filt, print.adj = FALSE)

## Network attributes:
##   vertices = 54
##   directed = FALSE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
## total edges = 173
##   missing edges = 0
##   non-missing edges = 173
## density = 0.1208945
##
## Vertex attributes:
##   vertex.names:
##     character valued attribute
##     54 valid vertex names
##
## Edge attributes:
##
##   collab:
##     numeric valued attribute
##     attribute summary:
##       Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       3.000   3.000   3.000   3.457   4.000   4.000

gden(d.filt)

## [1] 0.1208945

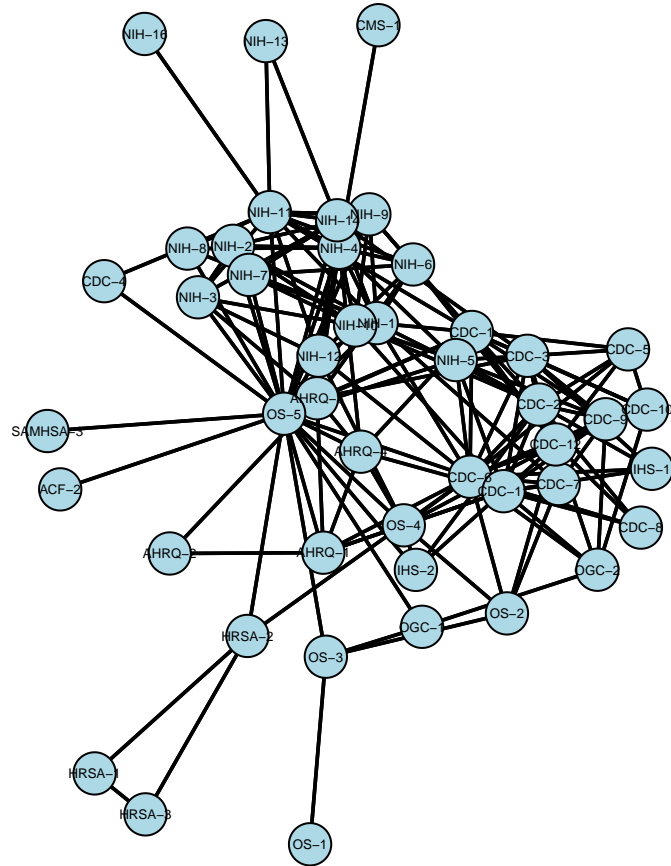
op <- par(mar = rep(0,4))
gplot(d.filt,
      gmode = "graph",
      displaylabels = TRUE,
      vertex.col = "lightblue",
      vertex.cex = 1.3,
      label.cex = 0.4,
      label.pos = 5,
      displayisolates = FALSE)
```



```

par(op)
op <- par(mar = rep(0,4))
d.val <- as.sociomatrix(d, attrname = "collab")
gplot(d.val,
      gmode = "graph",
      thresh = 2,
      displaylabels = TRUE,
      vertex.col = "lightblue",
      vertex.cex = 1.3,
      label.cex = 0.4,
      label.pos = 5,
      displayisolates = FALSE)

```



transforming directed to non-directed network

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    1    1    0    0
## [2,]    1    0    1    1    0
## [3,]    1    1    0    0    1
## [4,]    0    1    0    0    0
## [5,]    0    0    1    0    0
```

```
net1symm <- network(net1mat,
                    matrix.type = "adjacency")
network.vertex.names(net1symm) <- c("A","B","C","D","E")
summary(net1symm)
```

```
## Network attributes:
##   vertices = 5
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
## total edges = 10
##   missing edges = 0
##   non-missing edges = 10
## density = 0.5
##
## Vertex attributes:
##   vertex.names:
##     character valued attribute
##     5 valid vertex names
##
## No edge attributes
##
## Network adjacency matrix:
##   A B C D E
## A 0 1 1 0 0
## B 1 0 1 1 0
## C 1 1 0 0 1
## D 0 1 0 0 0
## E 0 0 1 0 0
```