Actor Prominence Chapter 7

Nick Lauerman

Started 26-Feb-2020

Contents

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()
## 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
```

```
library(RColorBrewer)
```

Data

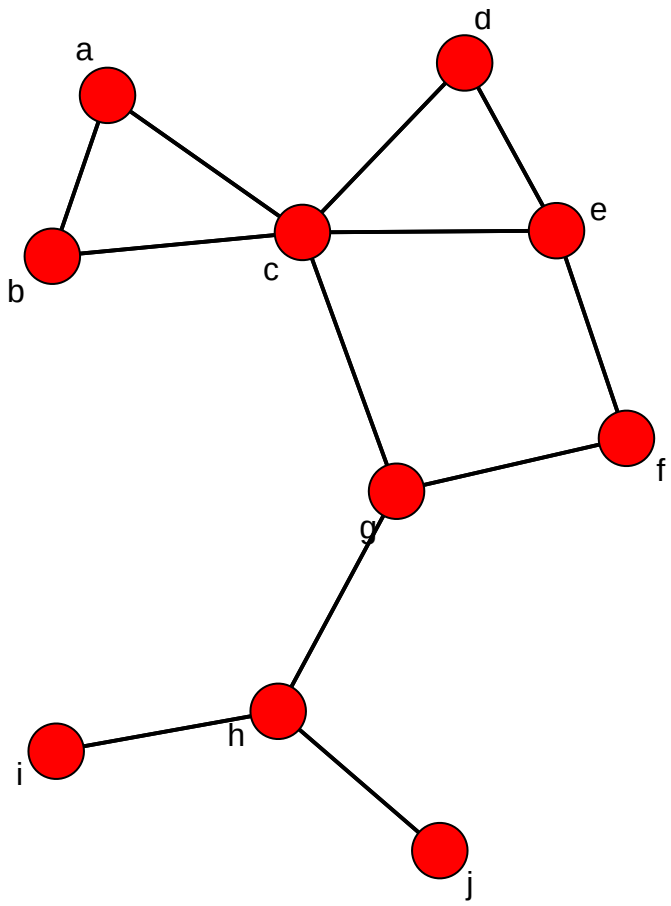
```
data(DHHS)
data("Bali")
```

Centerality: Prominence for undirected nets

```
net_mat <- rbind(c(0,1,1,0,0,0,0,0,0,0),
                 c(1,0,1,0,0,0,0,0,0,0),
                 c(1,1,0,1,1,0,1,0,0,0),
                 c(0,0,1,0,1,0,0,0,0,0),
                 c(0,0,1,1,0,1,0,0,0,0),
                 c(0,0,0,0,1,0,1,0,0,0),
                 c(0,0,1,0,0,1,0,1,0,0),
                 c(0,0,0,0,0,0,1,0,1,1),
                 c(0,0,0,0,0,0,0,1,0,0),
                 c(0,0,0,0,0,0,0,0,1,0,0))
rownames(net_mat) <- colnames(net_mat) <- letters[1:10]
net_mat
```

```
##   a b c d e f g h i j
## a 0 1 1 0 0 0 0 0 0 0
## b 1 0 1 0 0 0 0 0 0 0
## c 1 1 0 1 1 0 1 0 0 0
## d 0 0 1 0 1 0 0 0 0 0
## e 0 0 1 1 0 1 0 0 0 0
## f 0 0 0 0 1 0 1 0 0 0
## g 0 0 1 0 0 1 0 1 0 0
## h 0 0 0 0 0 0 1 0 1 1
## i 0 0 0 0 0 0 0 1 0 0
## j 0 0 0 0 0 0 0 1 0 0
```

```
net <- network(net_mat) # from following section
gplot(net,
      gmode = "graph",
      mode = "fruchtermanreingold",
      vertex.cex = 1.5,
      displaylabels = TRUE)
```



Three Common measure of Centerality

Degree Centrality

```
# net <- network(net_mat) # executed above
net %v% "vertex.names"
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
```

```
degree(net,
       gmode = "graph")
```

```
##  [1] 2 2 5 2 3 2 3 3 1 1
```

Closeness Centrality

```
closeness(net,
          gmode = "graph")
```

```
##  [1] 0.4090909 0.4090909 0.6000000 0.4285714 0.4500000 0.4500000 0.6000000
##  [8] 0.4736842 0.3333333 0.3333333
```

Betweenness Centrality

```
betweenness(net,
            gmode = "graph")
```

```
##  [1]  0.0  0.0 20.0  0.0  2.5  2.0 19.5 15.0  0.0  0.0
```

Centrality measures in *R*

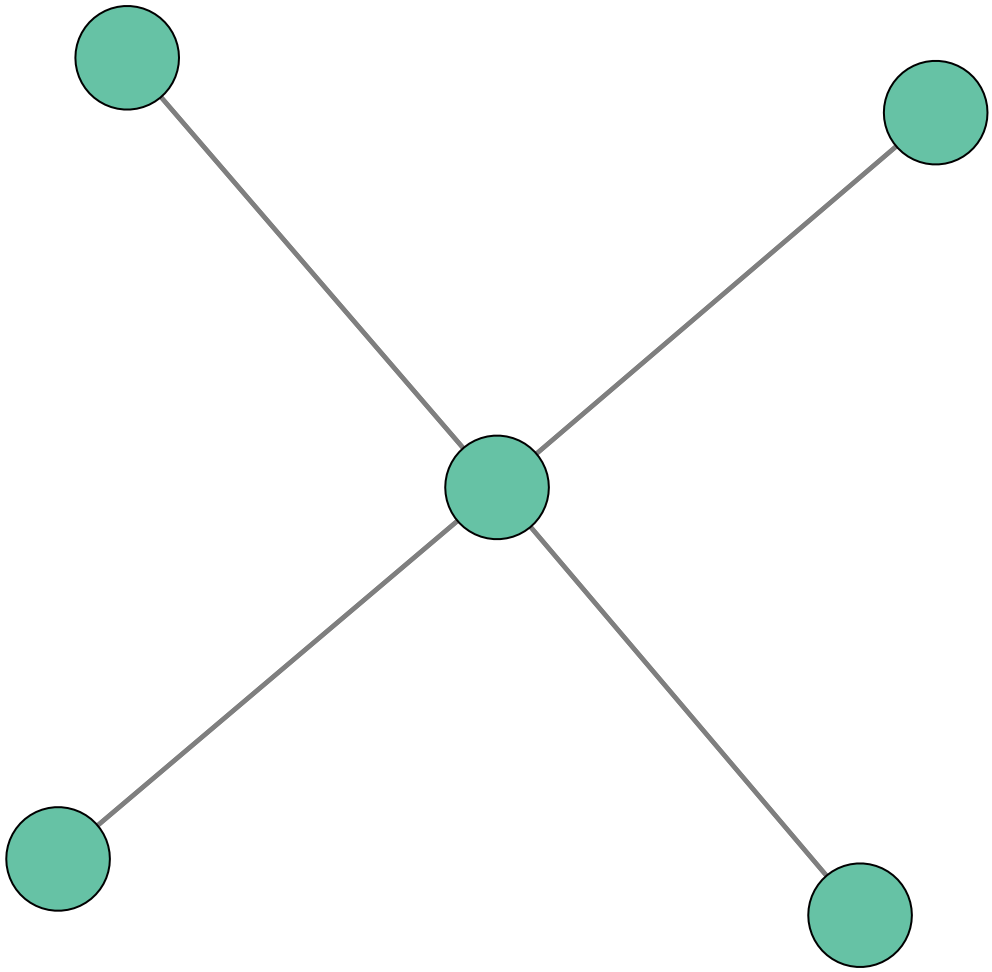
```
df.prom <- data.frame(
  deg = degree(DHHS),
  cls = closeness(DHHS),
  btw = betweenness(DHHS),
  evc = evcent(DHHS),
  inf = infocent(DHHS),
  flb = flowbet(DHHS)
)
cor(df.prom)
```

```
##           deg       cls       btw       evc       inf       flb
## deg 1.0000000 0.9728955 0.7495416 0.9724735 0.9017211 0.9442772
## cls 0.9728955 1.0000000 0.7867419 0.9338638 0.8897166 0.9409675
## btw 0.7495416 0.7867419 1.0000000 0.6002209 0.4854913 0.8844251
## evc 0.9724735 0.9338638 0.6002209 1.0000000 0.9402564 0.8428375
## inf 0.9017211 0.8897166 0.4854913 0.9402564 1.0000000 0.7733465
## flb 0.9442772 0.9409675 0.8844251 0.8428375 0.7733465 1.0000000
```

Centeralization: Network Level Indices of Centerality

```
dum1 <- rbind(c(1,2),
              c(1,3),
              c(1,4),
              c(1,5))
star_net <- network(dum1,
                    directed = FALSE)
dum2 <- rbind(c(1,2),
              c(2,3),
              c(3,4),
              c(4,5),
              c(5,1))
circle_net <- network(dum2,
                      directed = FALSE)
par(mar = c(4, 4, 0.1, 0.1))
my_pal <- brewer.pal(5, "Set2")
gplot(star_net,
      usearrows = FALSE,
```

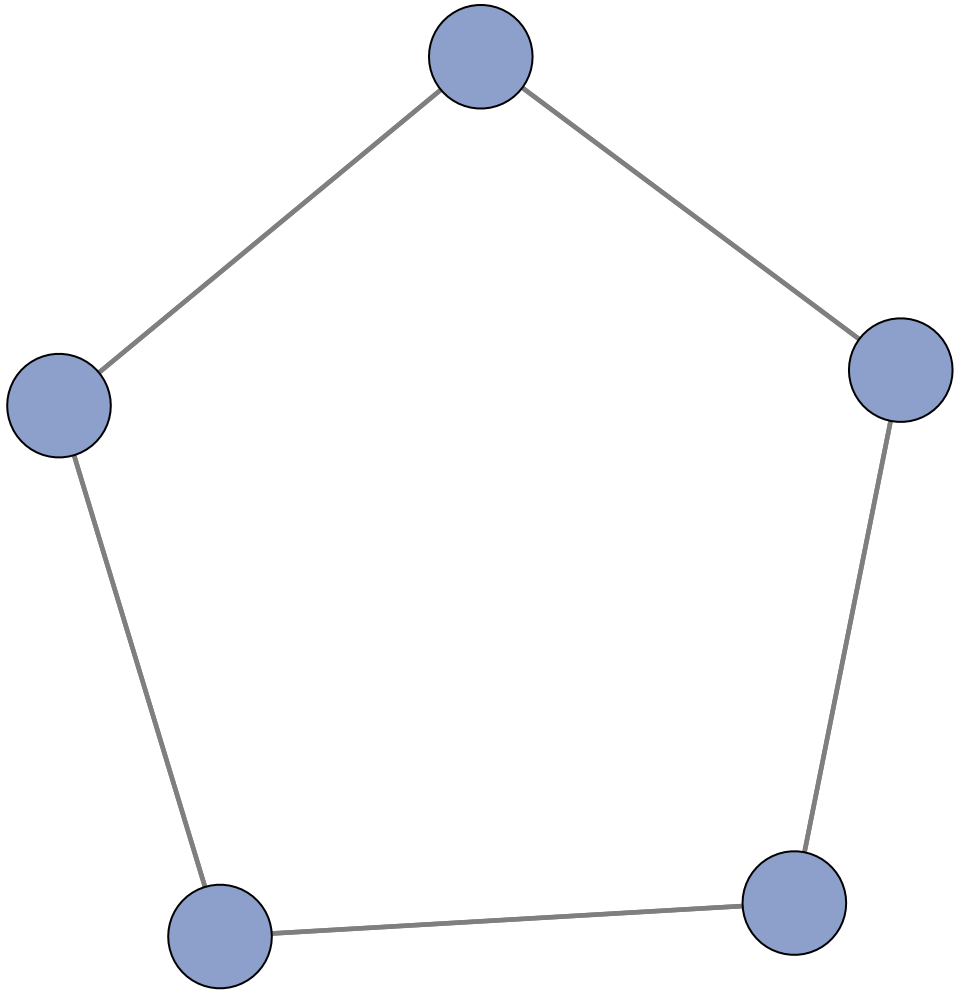
```
      displaylabels = FALSE,
      vertex.cex = 2,
      vertex.col = my_pal[1],
      edge.lwd = 0,
      edge.col = "grey50",
      xlab = "Star Graph")
```



Star Graph

```
gplot(circle_net,
      usearrows = FALSE,
      displaylabels = FALSE,
      vertex.cex = 2,
      vertex.col = my_pal[3],
      edge.lwd = 0,
      edge.col = "grey50",
```

```
      xlab = "Circle Graph")
```



Circle Graph

```
closeness(circle_net)
```

```
## [1] 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667
```

```
centralization(circle_net,
               closeness)
```

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

```
closeness(star_net)
```

```
## [1] 1.0000000 0.5714286 0.5714286 0.5714286 0.5714286
```

```
centralization(star_net,
                closeness)
```

```
## [1] 0.5357143
```

Reporting Centrality

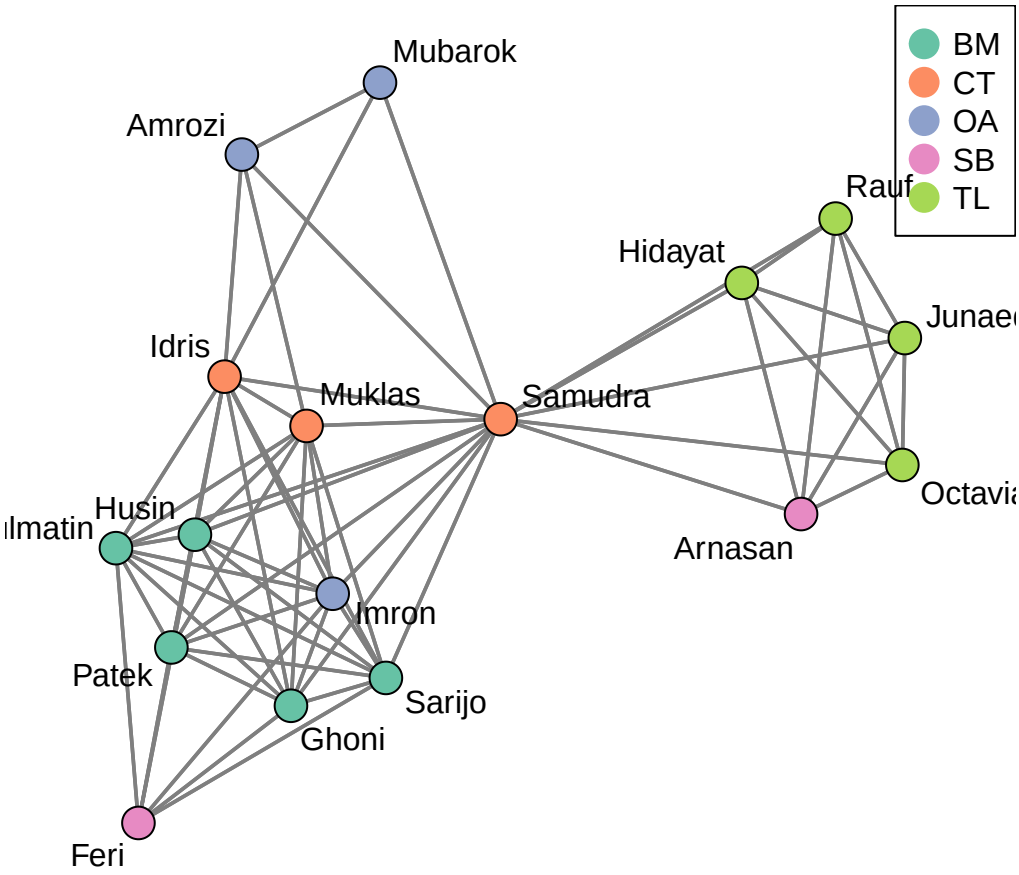
```
str(degree(Bali))
```

```
##  num [1:17] 18 8 18 30 18 20 6 18 18 10 ...
```

```
summary(degree(Bali))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6.00   10.00   18.00   14.82   18.00   30.00
```

```
rolecat <- Bali %v% "role"
gplot(Bali,
      usearrows = FALSE,
      displaylabels = TRUE,
      vertex.col = my_pal[as.factor(rolecat)],
      edge.lwd = 0,
      edge.col = "grey50")
legend("topright",
       legend = c("BM",
                  "CT",
                  "OA",
                  "SB",
                  "TL"),
       col = my_pal,
       pch = 19,
       pt.cex = 2.0)
```

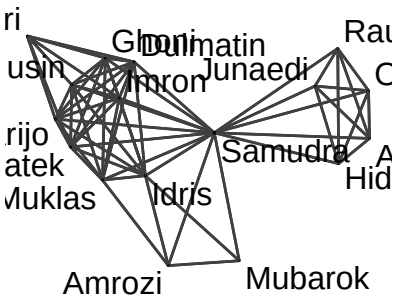



```
df.prom2 <- data.frame(
  degree = degree(Bali),
  closeness = closeness(Bali),
  betweenness = betweenness(Bali)
)
row.names(df.prom2) <- Bali %v% "vertex.names"
df.promsort <- df.prom2[order(-df.prom2$degree), ]
cd <- centralization(Bali, degree)
cc <- centralization(Bali, closeness)
cb <- centralization(Bali, betweenness)
df.promsort <- rbind(df.promsort, c(cd, cc, cb))
#row.names(df.promsort)[18] <- "\\empf[Centraluzation}" suspected for mark down for book
row.names(df.promsort)[18] <- "Centraluzation"
df.promsort
```

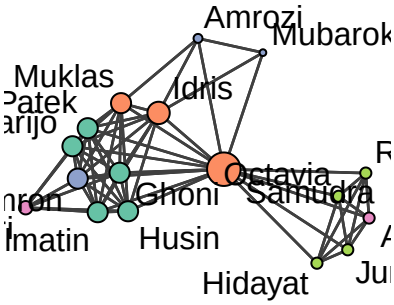
```
##                 degree closeness betweenness
## Samudra        30.0000 0.9411765 122.3333333
## Idris          20.0000 0.7272727  12.3333333
## Muklas         18.0000 0.6956522   4.6666667
## Imron          18.0000 0.6956522   3.3333333
## Dulmatin       18.0000 0.6956522   3.3333333
## Husin          18.0000 0.6956522   3.3333333
## Ghoni          18.0000 0.6956522   3.3333333
## Patek          18.0000 0.6956522   3.3333333
## Sarijo         18.0000 0.6956522   3.3333333
## Feri           12.0000 0.4848485   0.0000000
## Arnasan        10.0000 0.5714286   0.0000000
## Rauf           10.0000 0.5714286   0.0000000
## Octavia        10.0000 0.5714286   0.0000000
## Hidayat        10.0000 0.5714286   0.0000000
## Junaedi        10.0000 0.5714286   0.0000000
## Amrozi          8.0000 0.5517241   0.6666667
## Mubarok         6.0000 0.5333333   0.0000000
## Centraluzation  0.5375 0.3343513   0.4999132
```

```
deg <- degree(Bali, rescale = TRUE)
op <- par(mfrow = c(1,2))
gplot(Bali,
      usearrows = F,
      displaylabels = T,
      vertex.cex = deg,
      vertex.col = my_pal[as.factor(rolecat)],
      edge.lwd = 0,
      edge.col = "grey25",
      main = "Too small")
gplot(Bali,
      usearrows = F,
      displaylabels = T,
      vertex.cex = deg * 20,
      vertex.col = my_pal[as.factor(rolecat)],
      edge.lwd = 0,
      edge.col = "grey25",
      main = "a little better")
```

Too small



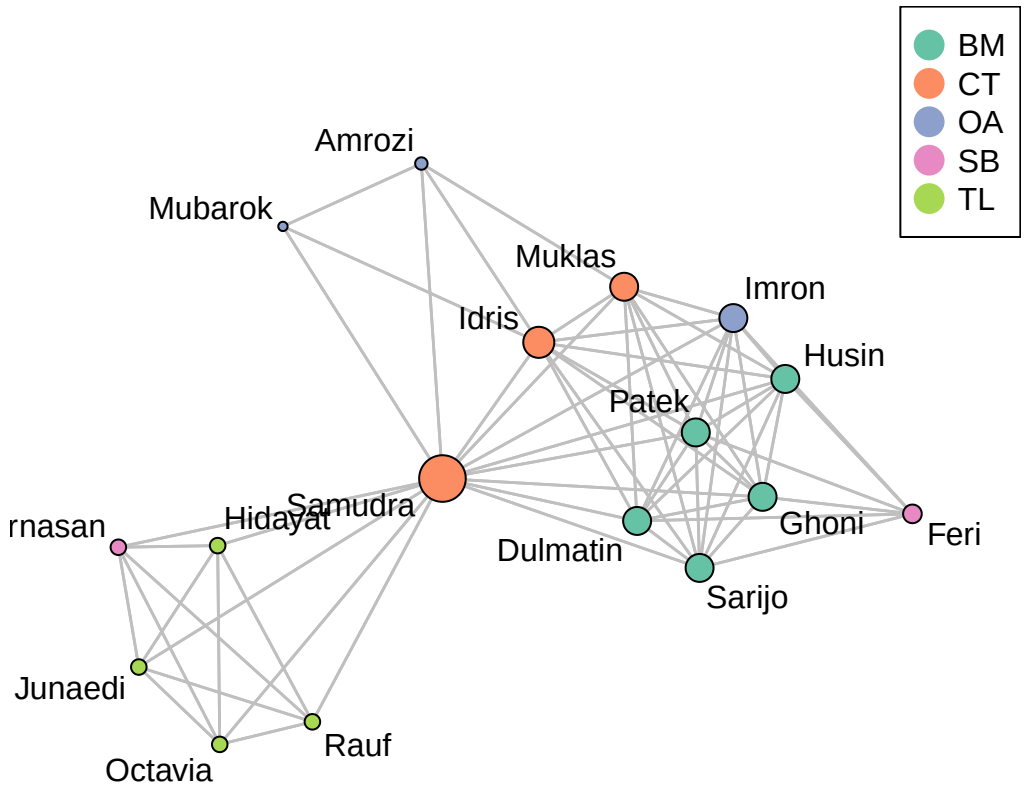
a little better




```
par(op)

gplot(Bali,
      usearrows = F,
      displaylabels = T,
      vertex.cex = deg * 12,
      vertex.col = my_pal[as.factor(rolecat)],
      edge.lwd = 0.5,
      edge.col = "grey75")
legend("topright",
       legend = c("BM",
                  "CT",
                  "OA",
                  "SB",
```

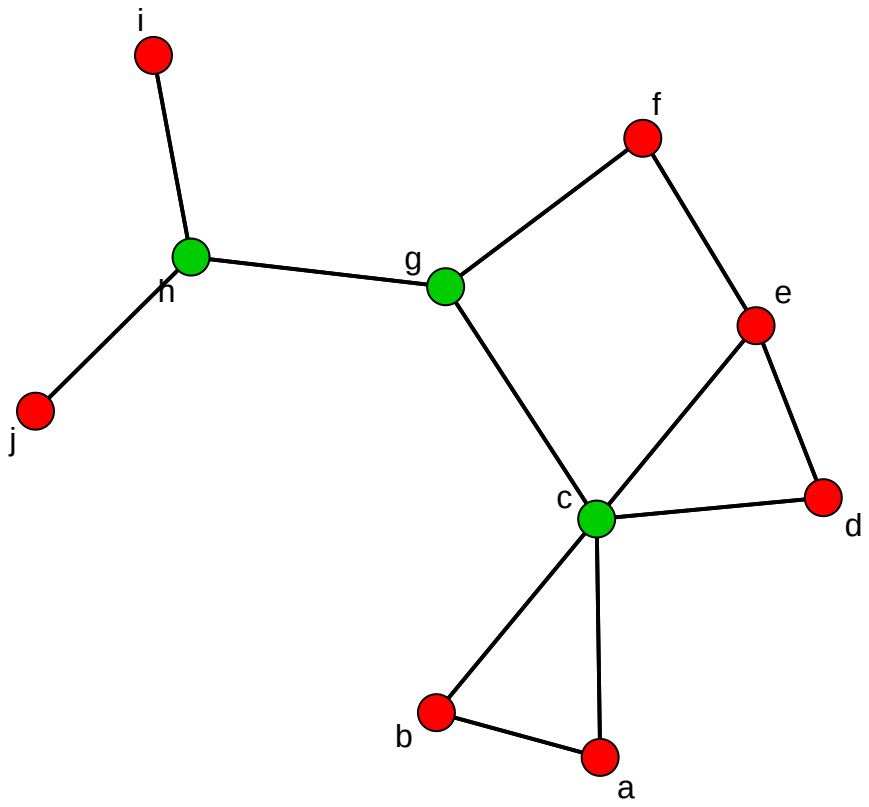
```
                  "TL"),
       col = my_pal,
       pch = 19,
       pt.cex = 2.0)
```



Cutpoints and Bridges

```
cpnet <- cutpoints(net,
                   mode = "graph",
                   return.indicator = TRUE)
gplot(net,
      gmode = "graph",
```

```
      vertex.col = cpnet + 2,
      #coord = coords, not surewhat coords does not exest
      jitter = FALSE,
      displaylabels = TRUE)
```



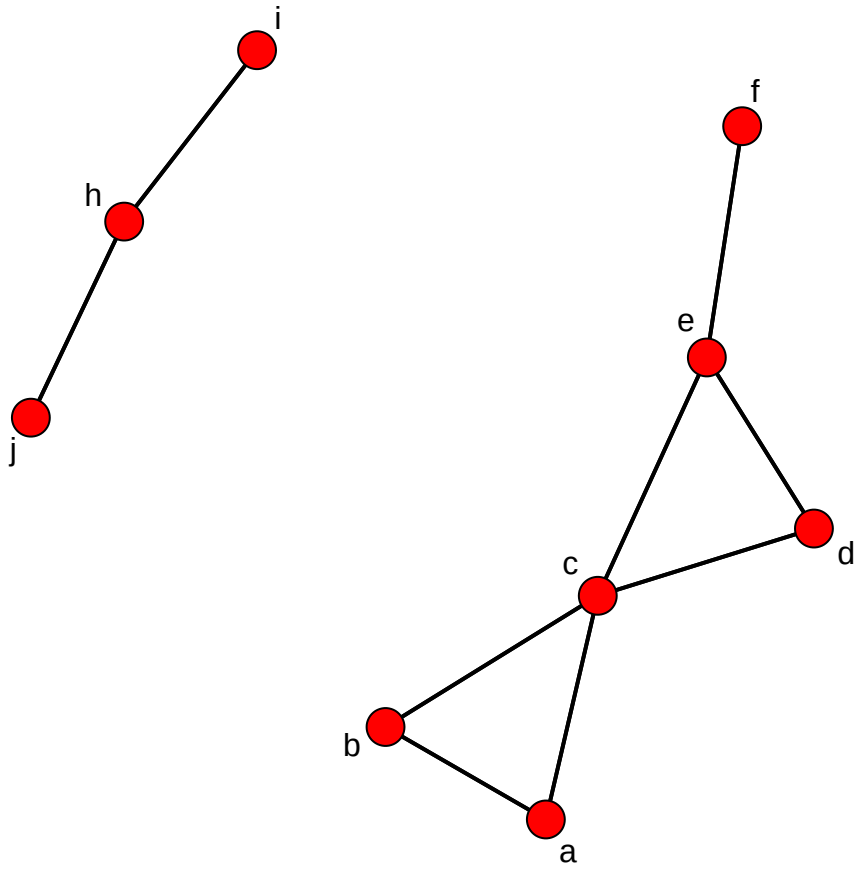

```
net2 <- net
components(net2)
```

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

```
delete.vertices(net2, 7)
components(net2)
```

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

```
gplot(net2,
      gmode = "graph",
      vertex.col = 2,
      jitter = FALSE,
      displaylabels = TRUE)
```



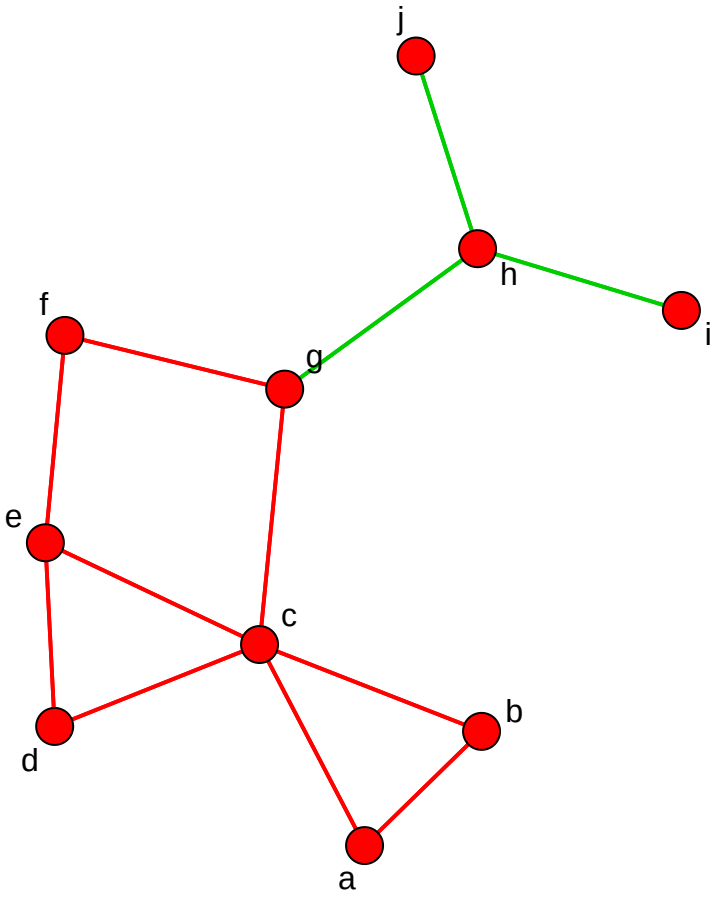

```
bridges <- function(dat,
                    mode = "graph",
                    connected = c("strong",
                                  "weak")){
  e_cnt <- network.edgecount(dat)
  if(mode == "graph"){
    cmp_cnt <- components(dat)
    b_vec <- rep(FALSE, e_cnt)
```

```
    for(i in 1:e_cnt){
      dat2 <- dat
      delete.edges(dat2, i)
      b_vec[i] <- (components(dat2) != cmp_cnt)
    }
  }
  else{
    cmp_cnt <- components(dat, connected = connected)
    b_vec <- rep(FALSE, e_cnt)
    for(i in 1:e_cnt){
      dat2 <- dat
      delete.edges(dat2, i)
      b_vec[i] <- (components(dat2, connected = connected) != cmp_cnt)
    }
  }
  return(b_vec)
}
```

```
bridges(net)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
brnet <- bridges(net)
gplot(net,
      gmode = "graph",
      vertex.col = "red",
      edge.col = brnet + 2,
      jitter = FALSE,
      displaylabels = TRUE)
```


j
h
i
f
g
e
c
b
d
a