Effective Network Graphic Design

Nick Lauerman

Contents

Libraries and data used

Libraries

```r
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
```

```r
library(RColorBrewer)
```

Data

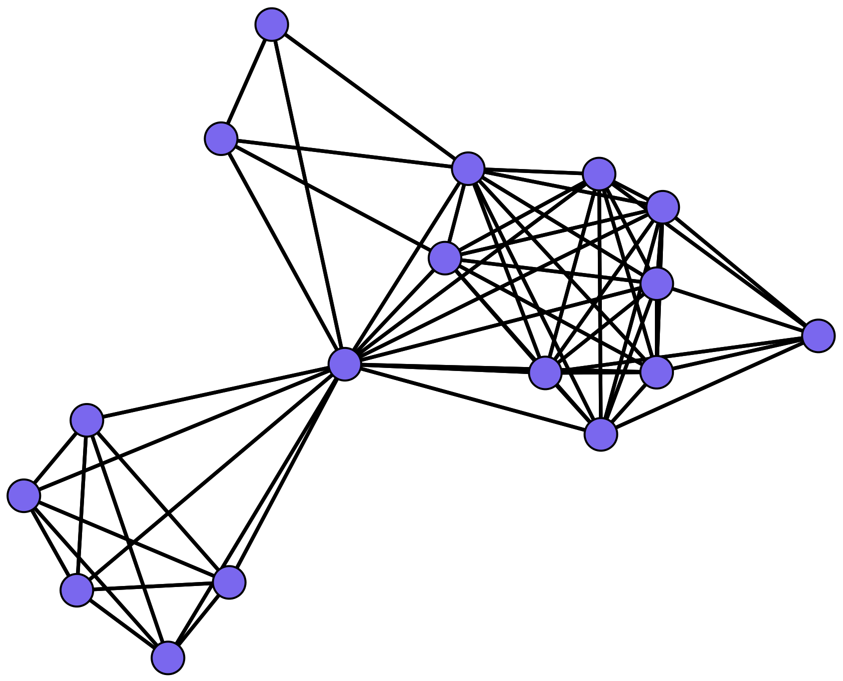
```r
data("Bali")
```

basic principles

design Elements

node color

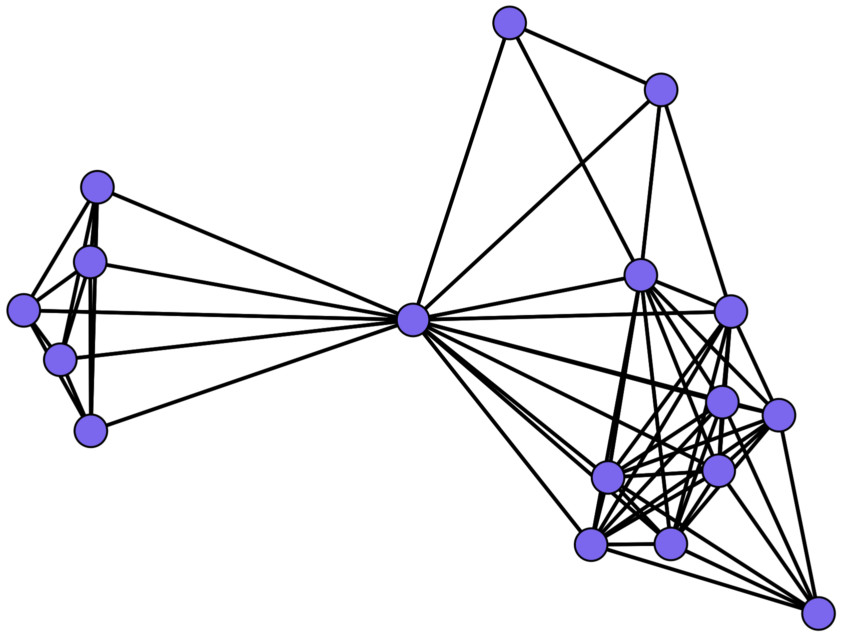
```
gplot(Bali,
      vertex.col = "slateblue2",
      gmode = "graph")
```



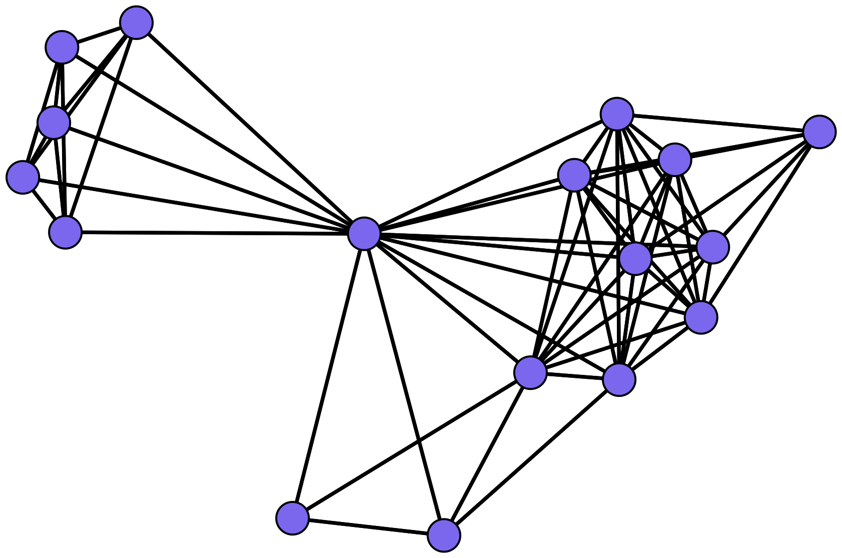
```
col2rgb('slateblue2')
```

```
##       [,1]
## red    122
## green  103
## blue   238
```

```
gplot(Bali,
      vertex.col = rgb(122, 103, 238,
                       maxColorValue = 255),
      gmode = "graph")
```

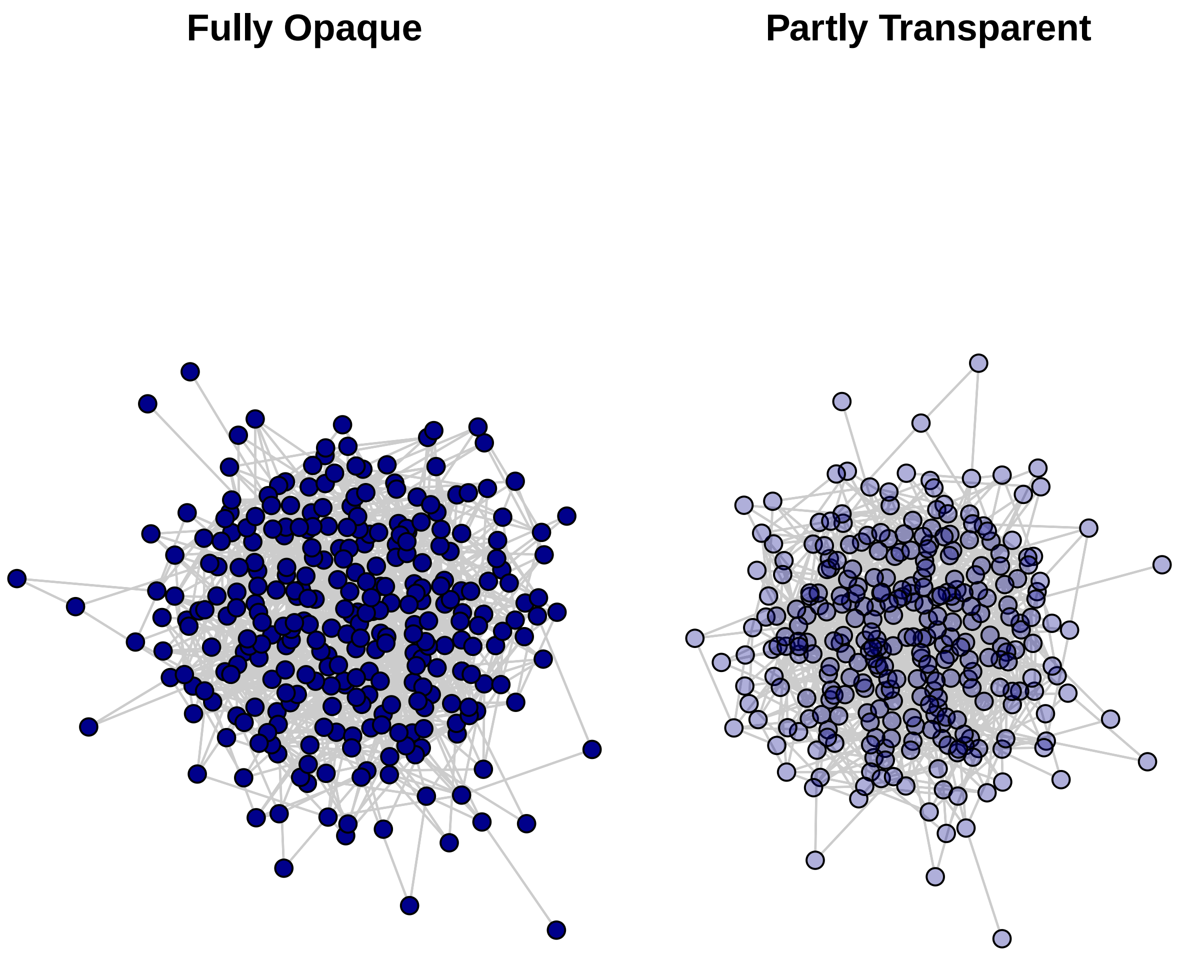


```
gplot(Bali,
      vertex.col = "#7A67EE",
      gmode = "graph")
```



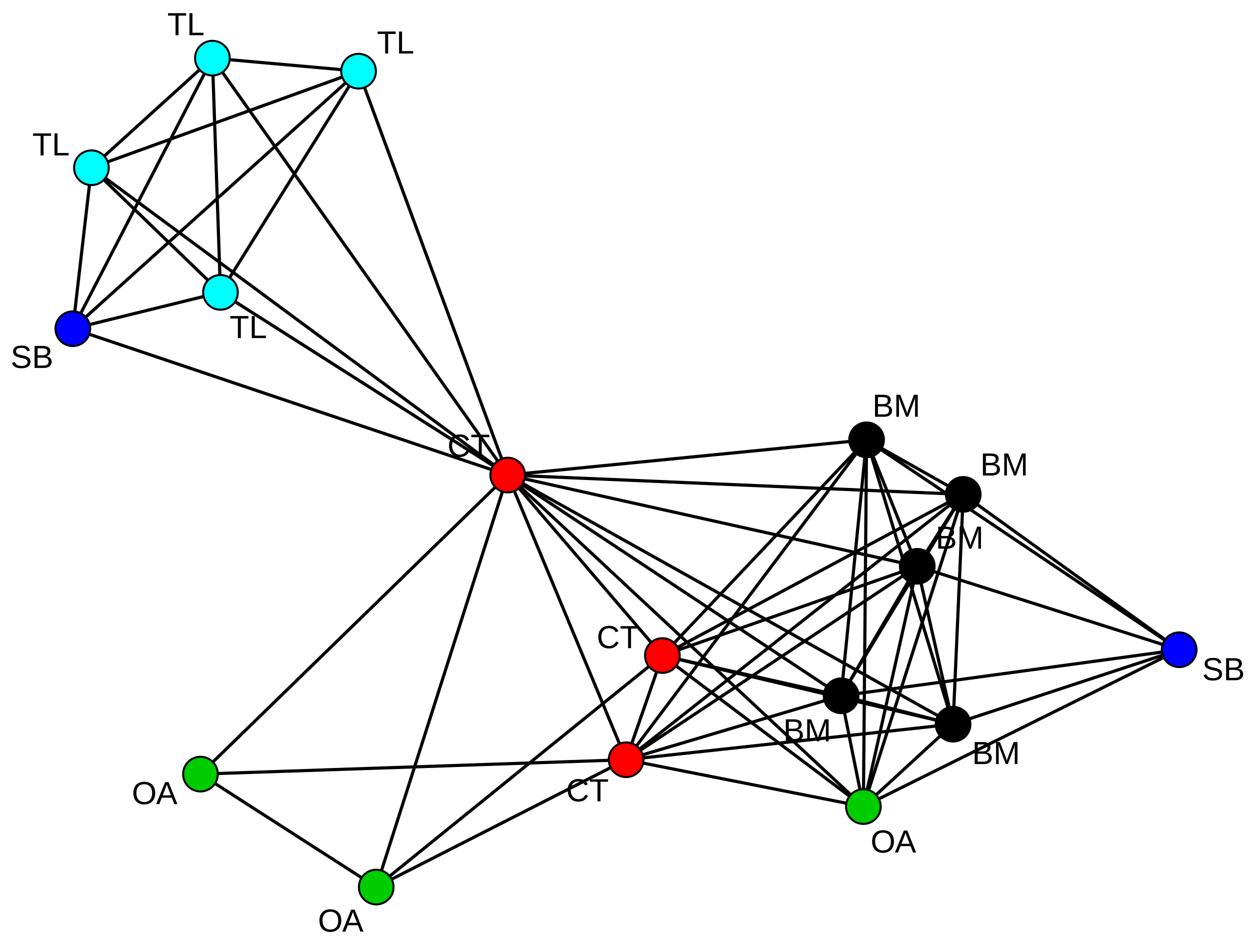
```
ndum <- rgraph(300,
               tprob = 0.025,
               mode = "graph")
op <- par(mar = c(0, 0, 2, 0),
          mfrow = c(1, 2))
gplot(ndum,
      gmode = "graph",
      vertex.cex = 2,
      vertex.col = rgb(0, 0, 139,
                       maxColorValue = 255),
      edge.col = "grey80",
      edge.lwd = 0.5,
      main = "Fully Opaque")
```

```r
gplot(ndum,
      gmode = "graph",
      vertex.cex = 2,
      vertex.col = rgb(0, 0, 139,
                       maxColorValue = 255,
                       alpha = 80),
      edge.col = "grey80",
      edge.lwd = 0.5,
      main = "Partly Transparent")
```



```r
par(op)

rolelab <- get.vertex.attribute(Bali,
                                "role")
```

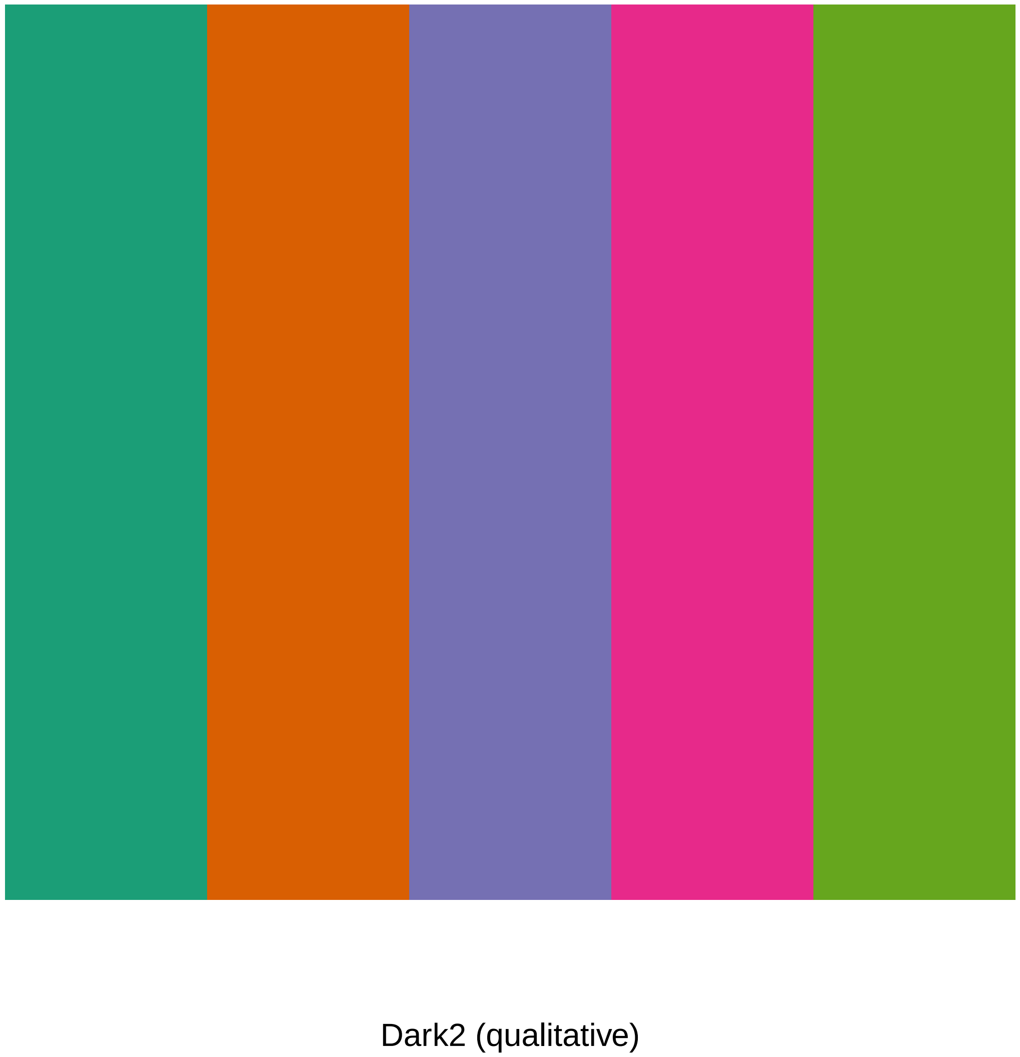
```r
op <- par(mar = c(0, 0, 0, 0))
plot(Bali,
     usearrows = FALSE,
     vertex.cex = 1.5,
     label = rolelab,
     displaylabels = TRUE,
     vertex.col = "role")
```



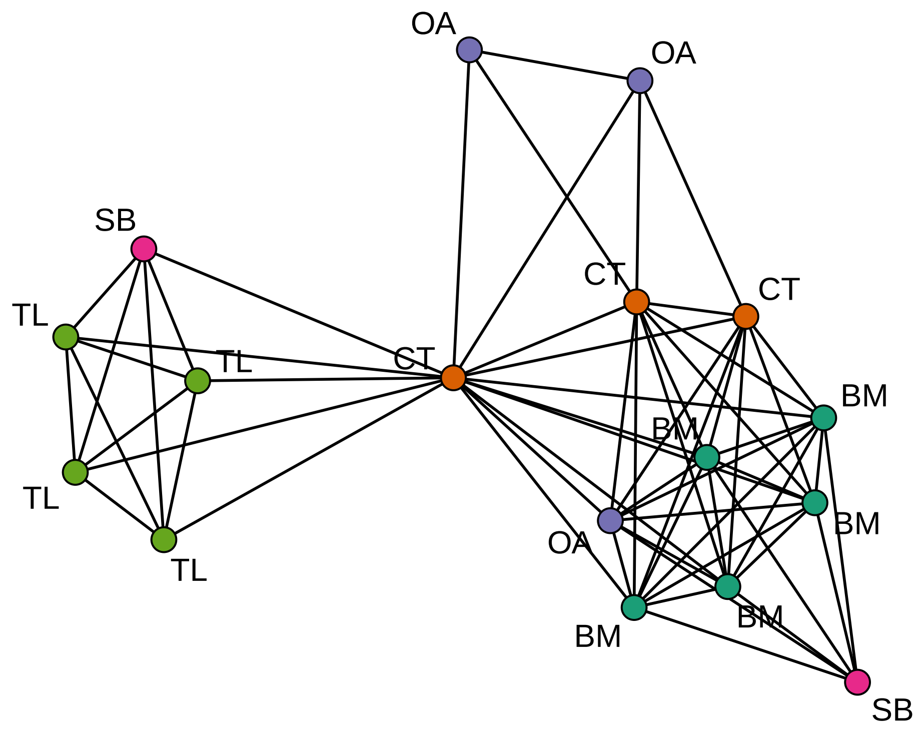
```r
par(op)

palette()
```

```
## [1] "black"   "red"     "green3"  "blue"    "cyan"    "magenta" "yellow" 
## [8] "gray"
```

```
display.brewer.pal(5, "Dark2")
```

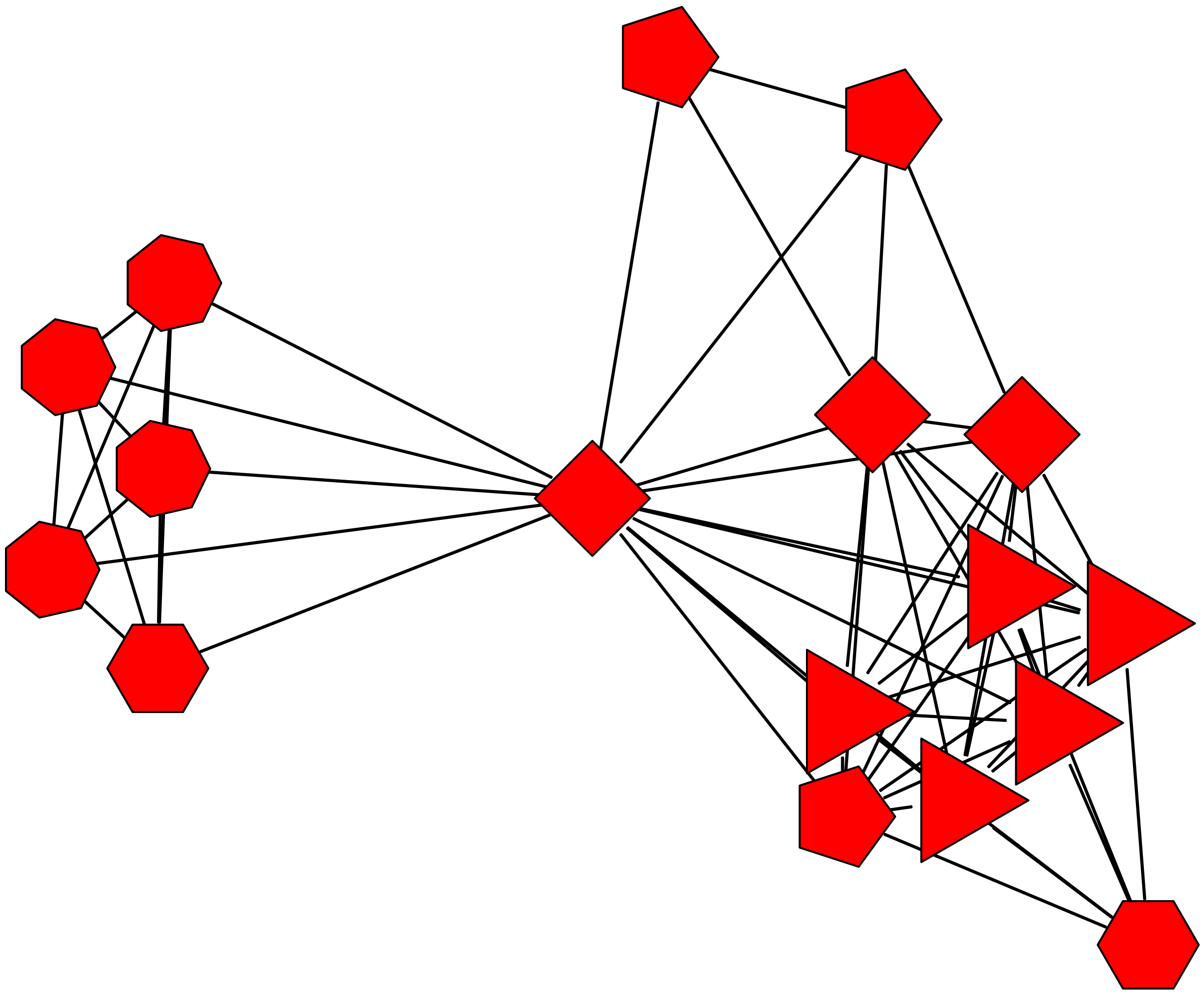


```
my_pal <- brewer.pal(5, "Dark2")
rolecat <- as.factor(get.vertex.attribute(Bali, "role"))
plot(Bali,
     vertex.cex = 1.5,
     label = rolelab,
     displaylabels = TRUE,
     vertex.col = my_pal[rolecat])
```



Node Shape

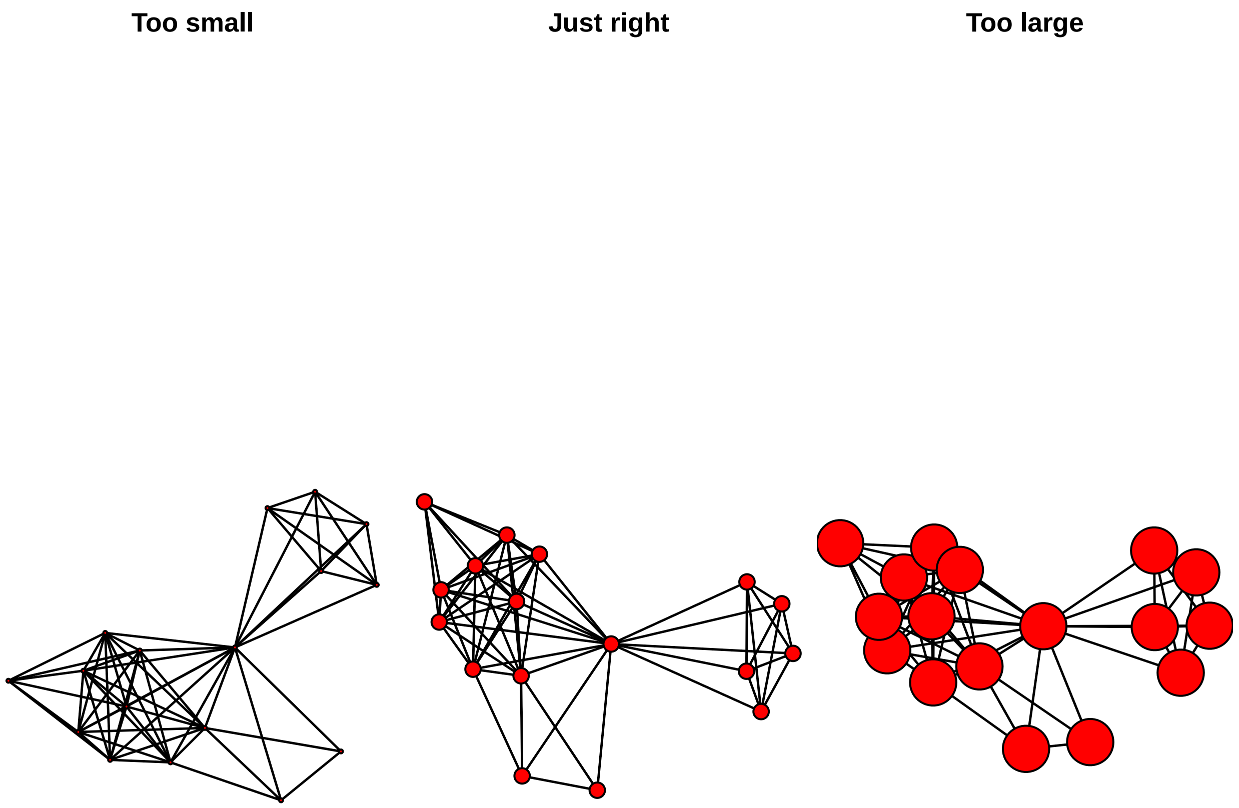
```
op <- par(mar = c(0, 0, 0, 0))
sidenum <- 3:7
plot(Bali,
     usearrows = FALSE,
     vertex.cex = 4,
     displaylabels = FALSE,
     vertex.sides = sidenum[rolecat])
```



```r
par(op)
```

Node Size

```r
op <- par(mar = c(0, 0, 1, 0),
          mfrow = c(1,3))
plot(Bali,
     vertex.cex = 0.5,
     main = "Too small")
plot(Bali,
     vertex.cex = 2,
     main = "Just right")
plot(Bali,
     vertex.cex = 6,
     main = "Too large")
```



```
par(op)

deg <- degree(Bali,
              gmode = "graph")
deg
```

```
##  [1]  9  4  9 15  9 10  3  9  9  5  5  5  5  5  9  6  9
```

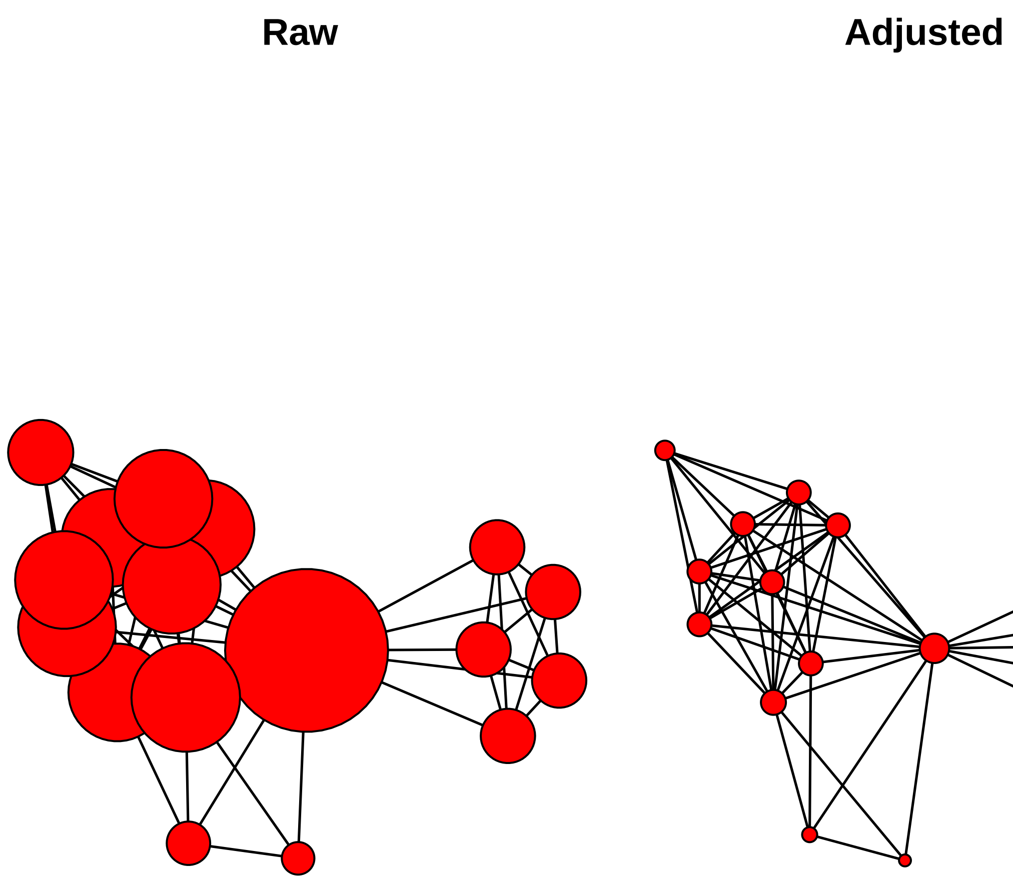
```
cls <- closeness(Bali,
                 gmode = "graph")
cls
```

```
##  [1] 0.6956522 0.5517241 0.6956522 0.9411765 0.6956522 0.7272727 0.5333333
##  [8] 0.6956522 0.6956522 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286
## [15] 0.6956522 0.4848485 0.6956522
```

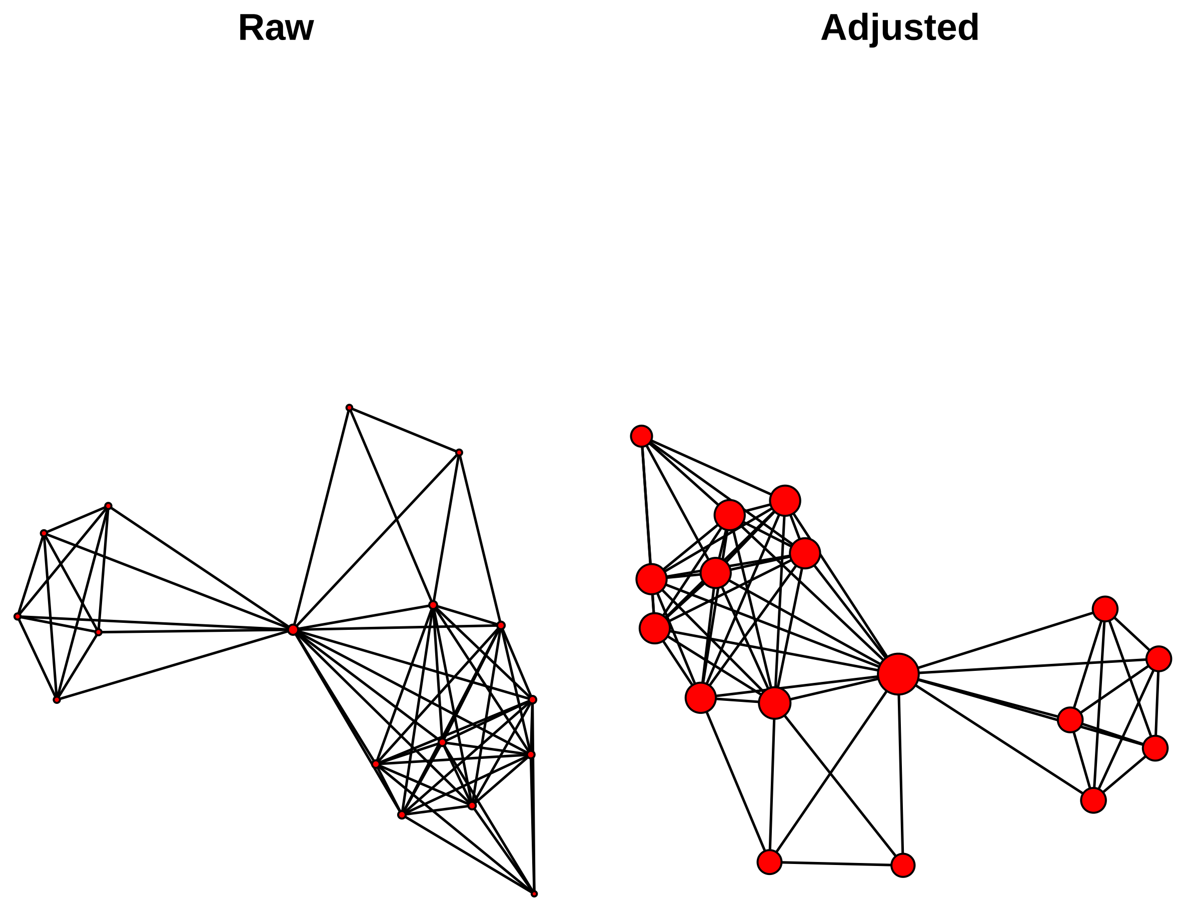
```
bet <- betweenness(Bali,
                   gmode = "graph")
bet
```

```
##  [1]  2.3333333  0.3333333  1.6666667 61.1666667  1.6666667  6.1666667
##  [7]  0.0000000  1.6666667  1.6666667  0.0000000  0.0000000  0.0000000
## [13]  0.0000000  0.0000000  1.6666667  0.0000000  1.6666667
```

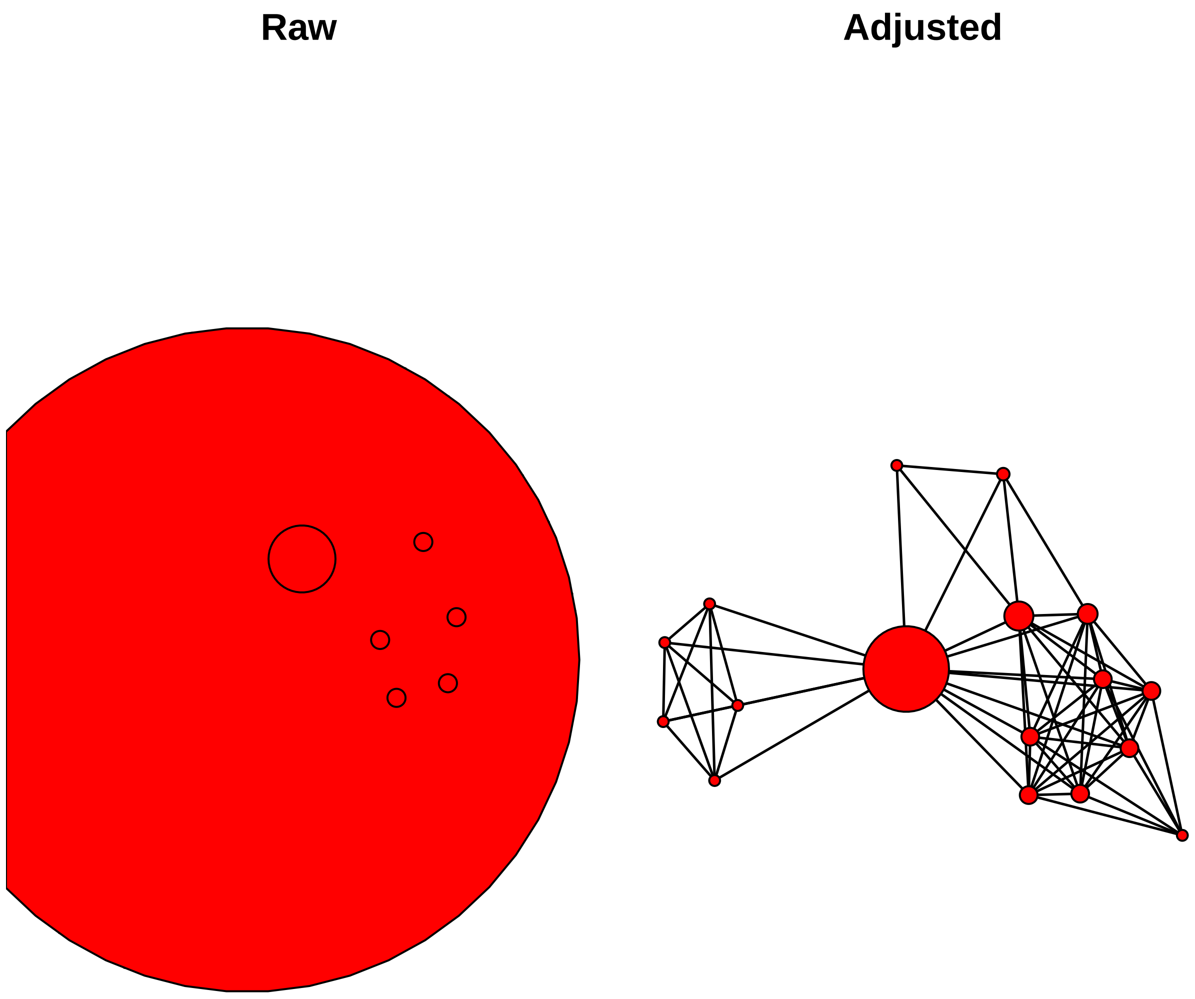
```
op <- par(mar = c(0, 0, 2, 1),
          mfrow = c(1,2))
plot(Bali,
     usearrows = T,
     vertex.cex = deg,
     main = "Raw")
plot(Bali,
     usearrows = F,
     vertex.cex = log(deg),
     main = "Adjusted")
```



```
par(op)

op <- par(mar = c(0, 0, 2, 1),
          mfrow = c(1,2))
plot(Bali,
     usearrows = T,
     vertex.cex = cls,
     main = "Raw")
plot(Bali,
     usearrows = F,
     vertex.cex = 4*cls,
     main = "Adjusted")
```

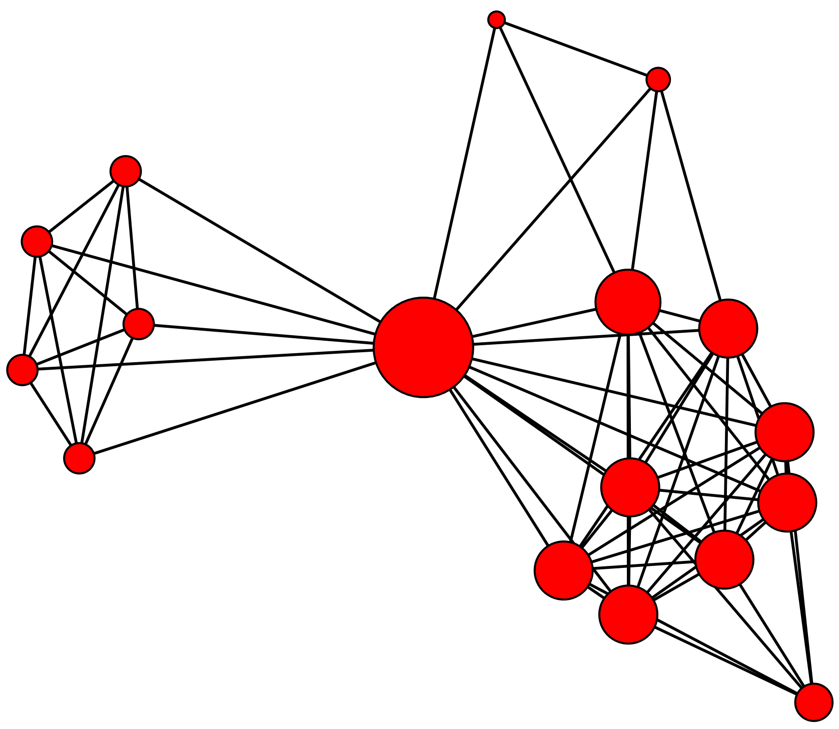


```r
par(op)

op <- par(mar = c(0, 0, 2, 1),
          mfrow = c(1,2))
plot(Bali,
     usearrows = T,
     vertex.cex = bet,
     main = "Raw")
plot(Bali,
     usearrows = F,
     vertex.cex = sqrt(bet + 1),
     main = "Adjusted")
```



```r
par(op)

rescale <- function(nchar, low, high){
  min_d <- min(nchar)
  max_d <- max(nchar)
  rscl <- ((high - low) * (nchar - min_d)) /
    (max_d - min_d) + low
  return(rscl)
}
plot(Bali,
     vertex.cex = rescale(deg, 1, 6),
     main = "Adjusted node size wth rescale function")
```

Adjusted node size wth rescale function

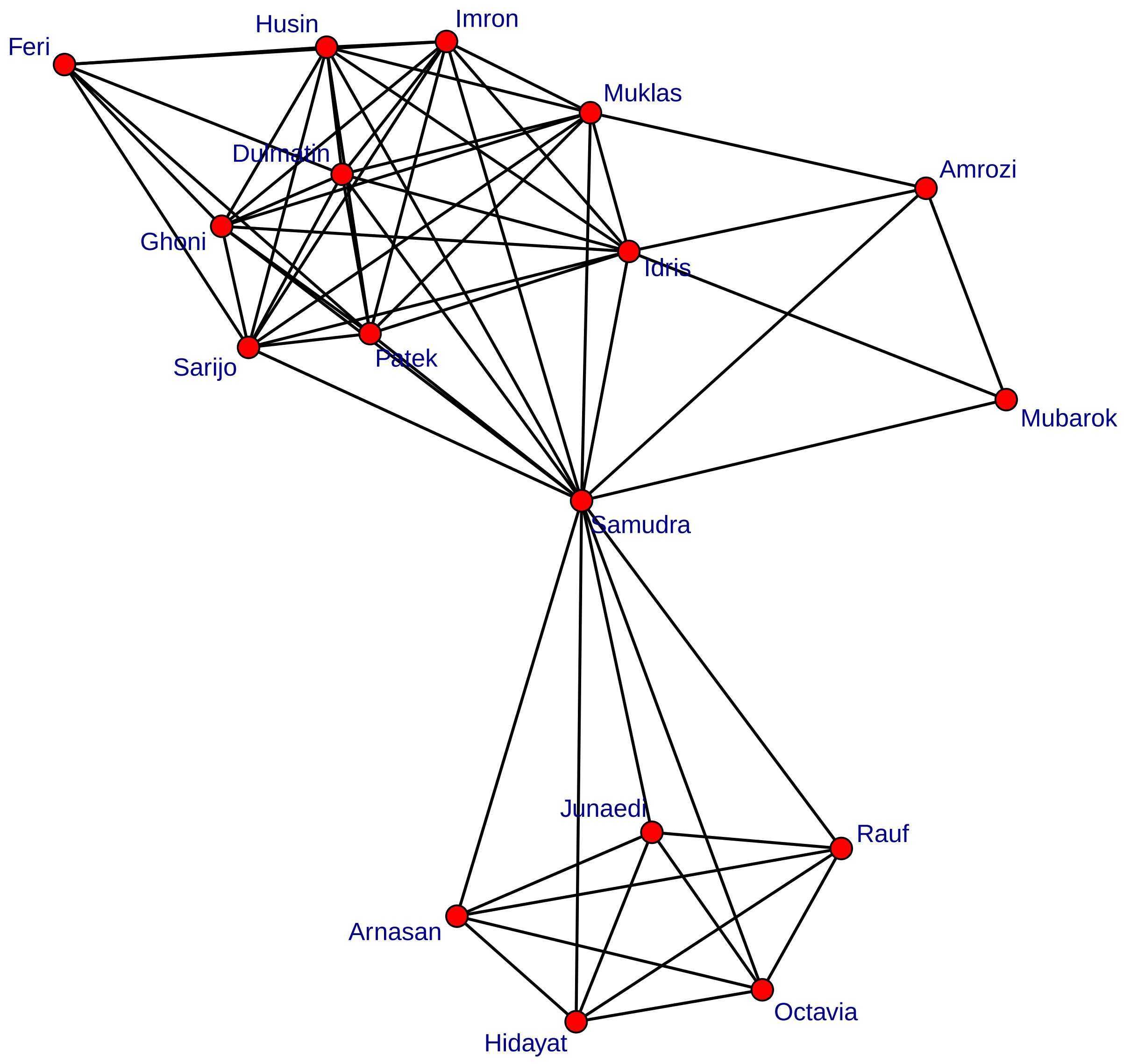


node label

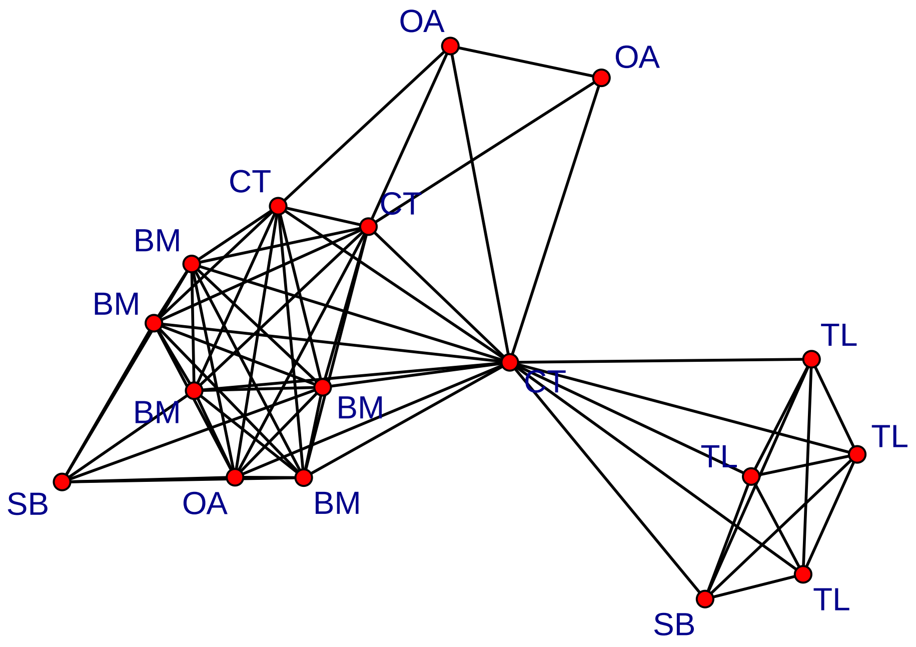
```
get.vertex.attribute(Bali,
                     "vertex.names")
```

```
##  [1] "Muklas"   "Amrozi"   "Imron"    "Samudra"  "Dulmatin" "Idris"
##  [7] "Mubarok"  "Husin"    "Ghoni"    "Arnasan"  "Rauf"     "Octavia"
## [13] "Hidayat"  "Junaedi"  "Patek"    "Feri"     "Sarijo"
```

```
op <- par(mar = c(0, 0, 0, 0))
plot(Bali,
     displaylabels = TRUE,
     label.cex = 0.8,
     pad = 0.4,
      label.col = "darkblue")
```

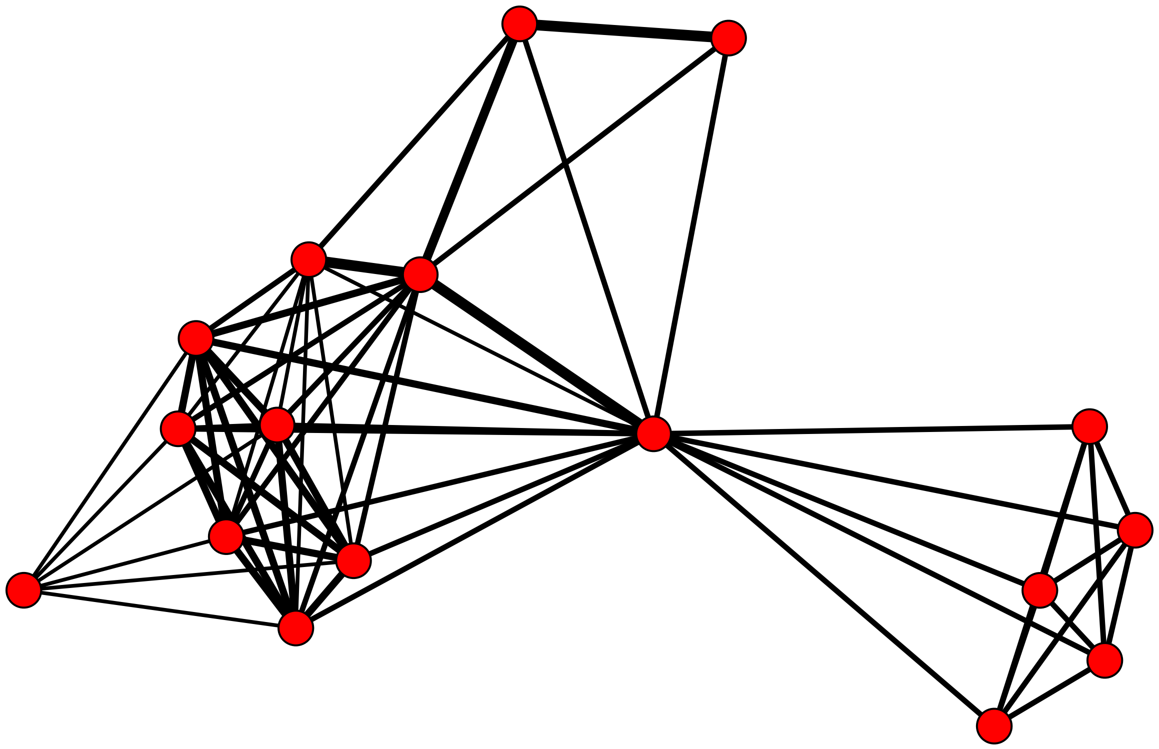


```
par(op)

rolelab <- get.vertex.attribute(Bali, "role")
plot(Bali,
      usearrows = FALSE,
      label = rolelab,
      displaylabels = TRUE,
      label.col = "darkblue")
```



Edge width

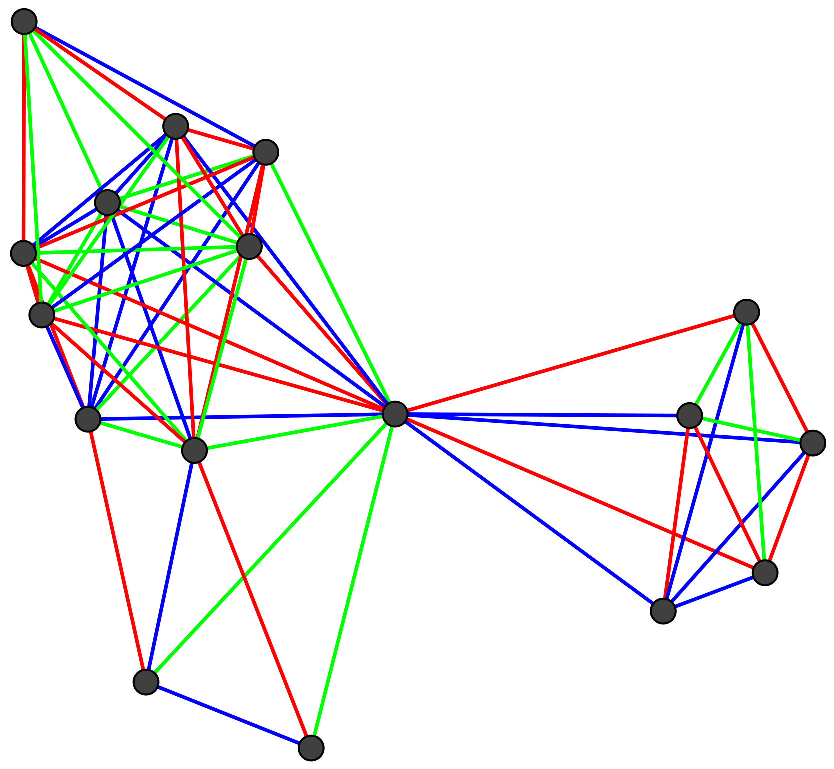
```r
op <- par(mar = c(0, 0, 0, 0))
IClevel <- Bali %e% "IC"
plot(Bali,
     vertex.cex = 1.5,
     edge.lwd  = 1.5 * IClevel)
```



```
par(op)
```

edge color

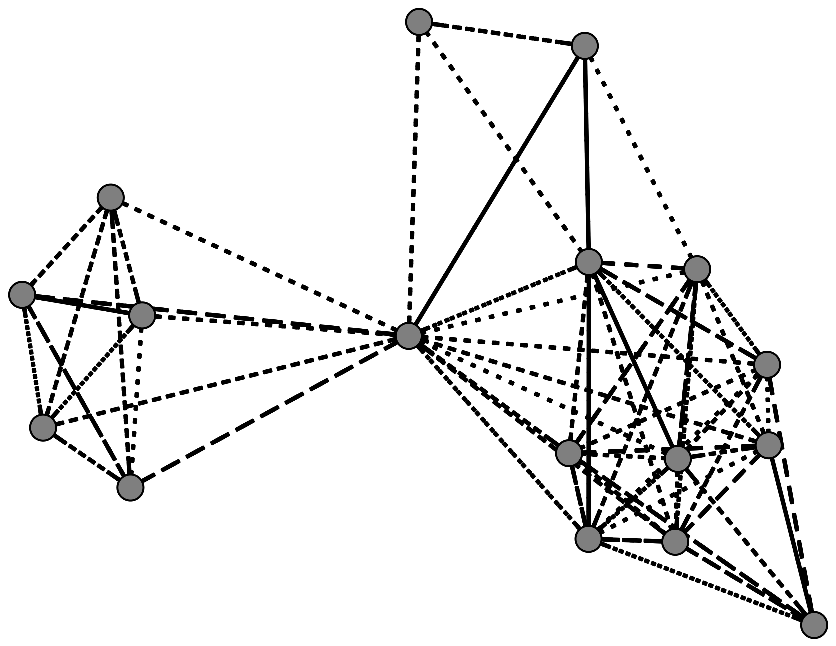
```
n_edge <- network.edgecount(Bali)
edge_cat <- sample(1:3,
                   n_edge,
                   replace = T)
linecol_pal <- c("blue",
                 "red",
                 "green")
plot(Bali,
     vertex.cex = 1.5,
     vertex.col = "grey25",
     edge.col = linecol_pal[edge_cat],
     edge.lwd = 2)
```



edge type

```
n_edge <- network.edgecount(Bali)
edge_cat <- sample(1:3,
                   n_edge,
                   replace = T)
line_pal <- c(2,
              3,
              4)
```

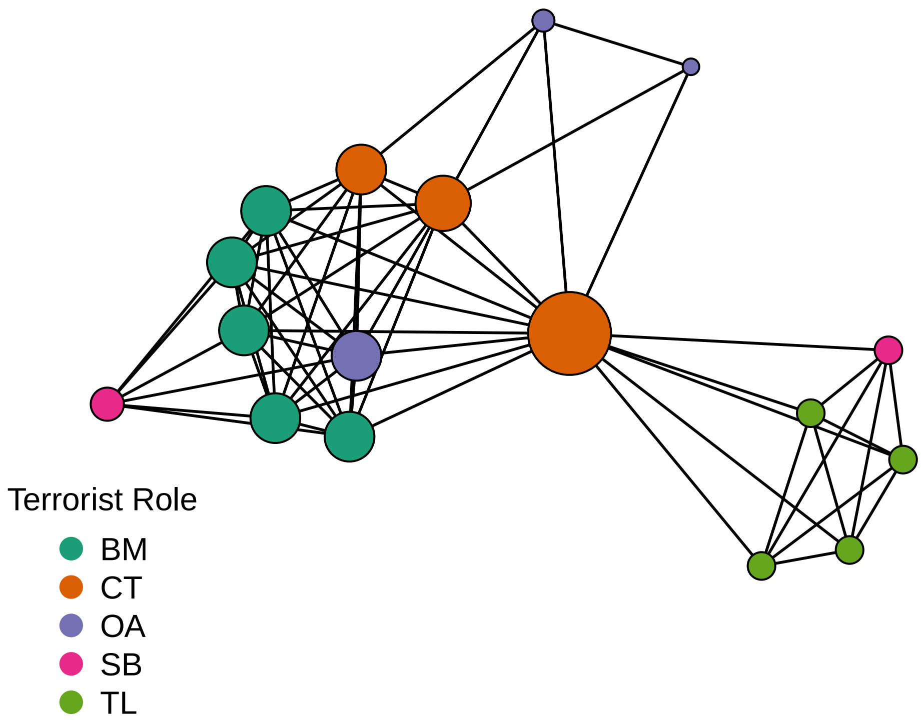
```
gplot(Bali,
      vertex.cex = 0.8,
      gmode = "graph",
      vertex.col = "grey50",
      edge.lwd = 1.5,
      edge.lty = line_pal[edge_cat])
```



Legends

```
my_pal <- brewer.pal(5, "Dark2")
rolecat <- as.factor(get.vertex.attribute(Bali,
                                          "role"))
plot(Bali,
```

```r
     vertex.cex = rescale(deg, 1, 5),
     vertex.col = my_pal[rolecat])
legend("bottomleft",
       legend = c("BM",
                  "CT",
                  "OA",
                  "SB",
                  "TL"),
      col = my_pal,
      pch = 19,
      pt.cex = 1.5,
      bty = "n",
      title = "Terrorist Role"
      )
```


Terrorist Role
BM
CT
OA
SB
TL