Mining Graph Data

Hazim Fitri

2025-01-12

Contents

Data Jaringan (Network Science)

```r
# Perlombongan data graf
library(igraph)
```

```
##
## Attaching package: 'igraph'

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

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

Types of graph

1) Directed graph

```r
g = graph_from_literal(1-2, 1-3, 1-7, 3-4, 2-3, 2-4, 3-5, 4-5,
                       4-6, 4-7, 5-6, 5-8, 6-7, 7-8)
g
```

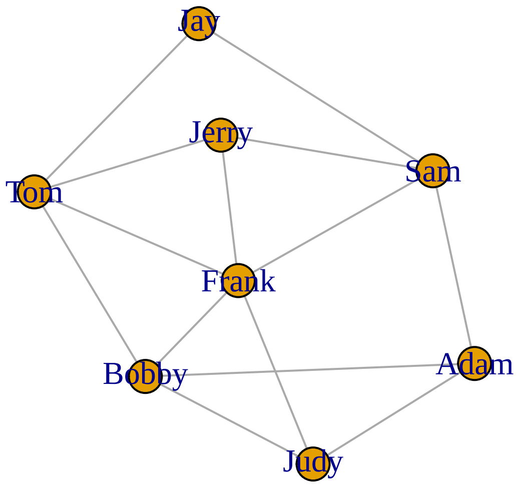
```
## IGRAPH 1b59e92 UN-- 8 14 --
## + attr: name (v/c)
## + edges from 1b59e92 (vertex names):
##  [1] 1--2 1--3 1--7 2--3 2--4 3--4 3--5 7--4 7--6 7--8 4--5 4--6 5--6 5--8
```

```r
# labelkan nod/verteks

V(g)$name = c('Adam', 'Judy', 'Bobby', 'Sam', 'Frank', 'Tom', 'Jerry',
              'Jay')
g
```

```
## IGRAPH 1b59e92 UN-- 8 14 --
## + attr: name (v/c)
## + edges from 1b59e92 (vertex names):
##  [1] Adam --Judy  Adam --Bobby Adam --Sam   Judy --Bobby Judy --Frank
##  [6] Bobby--Frank Bobby--Tom   Sam  --Frank Sam  --Jerry Sam  --Jay
## [11] Frank--Tom   Frank--Jerry Tom  --Jerry Tom  --Jay
```

```r
# Plot graf dengan hubungan tak terarah
set.seed(12)
plot(g)
```

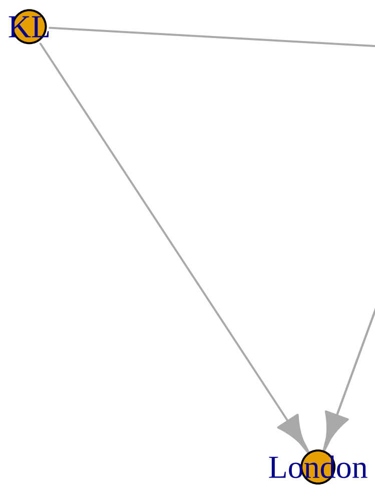


2) Undirected Graph

```r
dg = graph_from_literal(KL-+CHINA, KL-+London, CHINA++London)
dg
```

```
## IGRAPH 1b6de77 DN-- 3 4 --
## + attr: name (v/c)
## + edges from 1b6de77 (vertex names):
## [1] KL    ->CHINA  KL    ->London CHINA ->London London->CHINA
```

```r
plot(dg)
```



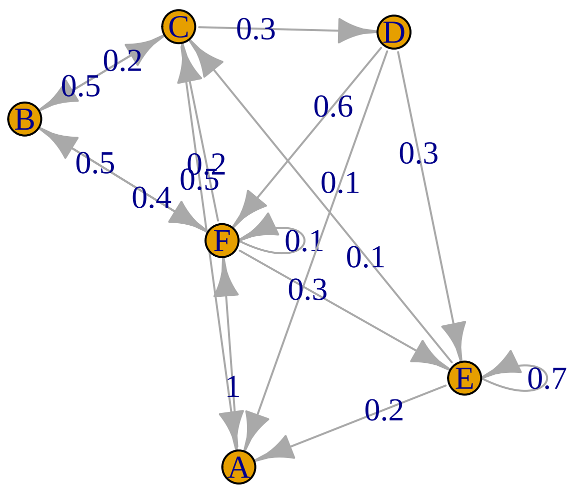
3) Weighted Graph

- berapa kuat hubungan antara nod/verteks

```r
m = read.table(row.names=1, header=T,
               text=' A   B   C   D   E   F
                    A 0   0   0   0   0   1
                    B 0   0   0.5 0   0   0.5
                    c 0.5 0.2 0   0.3 0   0
                    D 0.1 0   0   0   0.3 0.6
                    E 0.2 0   0.1 0   0.7 0
                    F 0   0.4 0.2 0   0.3 0.1')

m = as.matrix(m)

ig = graph_from_adjacency_matrix(m, weighted=T)
plot(ig, edge.label=E(ig)$weight)
```



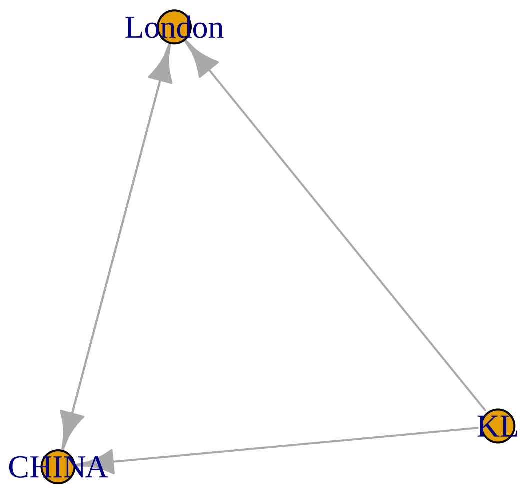
4) Labeled Graph

5) Cyclic Graph

At least ada 1 kitaran dah boleh dianggap sebagai graf berkitar

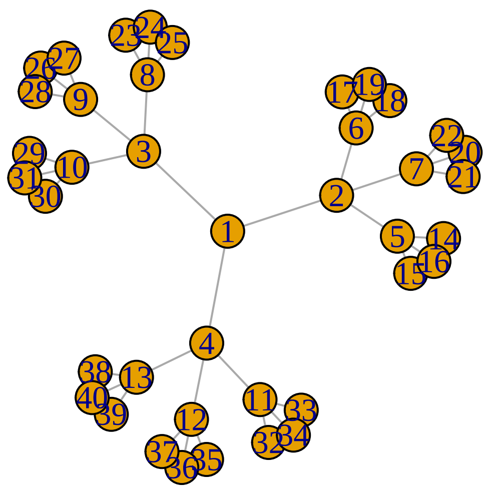
6) Acyclic Graph

```r
plot(dg)
```



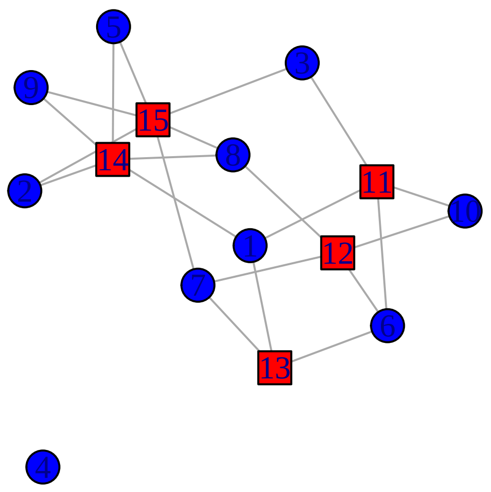
7) Trees Graph

```r
tr = make_tree(40, children=3, mode='undirected')
plot(tr)
```



8) Bipartite Graph

```
gb = sample_bipartite(10,5,p=0.4)
col = c('blue', 'red')
shape = c('circle', 'square')
plot(gb, vertex.color = col[as.numeric(V(gb)$type+1)],
     vertex.shape=shape[as.numeric(V(gb)$type+1)])
```



9) Hypergraph

```
library(HyperG)
```

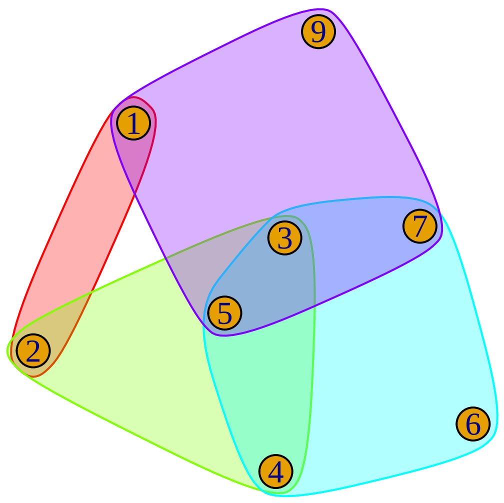
```
## Loading required package: mclust

## Package 'mclust' version 6.1.1
## Type 'citation("mclust")' for citing this R package in publications.

##
## Attaching package: 'HyperG'

## The following objects are masked from 'package:igraph':
##
##     is.simple, line.graph
```

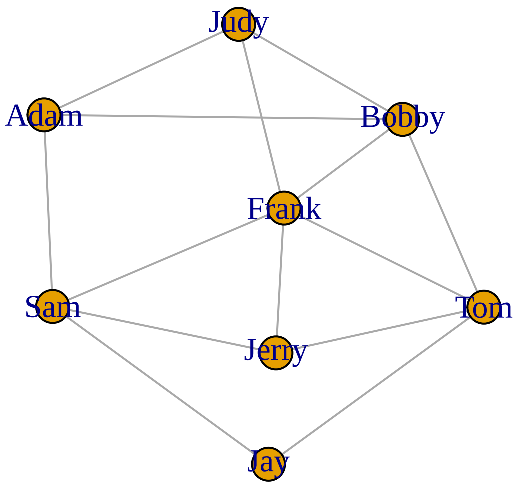
```
h = hypergraph_from_edgelist(list(1:2, 2:5, 3:7, c(1,3,5,7,9)))
plot(h)
```



Representations for Graphs

1) Adjacency list

```
plot(g)
```



```
Adj.list1 = as_adj_list(g)
Adj.list1
```

```
## $Adam
## + 3/8 vertices, named, from 1b59e92:
## [1] Judy  Bobby Sam
##
## $Judy
## + 3/8 vertices, named, from 1b59e92:
## [1] Adam  Bobby Frank
##
## $Bobby
## + 4/8 vertices, named, from 1b59e92:
## [1] Adam  Judy  Frank Tom
##
## $Sam
## + 4/8 vertices, named, from 1b59e92:
## [1] Adam  Frank Jerry Jay
##
## $Frank
## + 5/8 vertices, named, from 1b59e92:
## [1] Judy  Bobby Sam   Tom   Jerry
##
## $Tom
## + 4/8 vertices, named, from 1b59e92:
## [1] Bobby Frank Jerry Jay
##
## $Jerry
## + 3/8 vertices, named, from 1b59e92:
## [1] Sam   Frank Tom
##
## $Jay
## + 2/8 vertices, named, from 1b59e92:
## [1] Sam Tom
```

```
Adj.list2 = as_adj_list(dg)
Adj.list2
```

```
## $KL
## + 2/3 vertices, named, from 1b6de77:
## [1] CHINA  London
##
## $CHINA
## + 3/3 vertices, named, from 1b6de77:
## [1] KL     London London
##
## $London
## + 3/3 vertices, named, from 1b6de77:
## [1] KL    CHINA CHINA
```

```
Adj.list3 = as_adj_list(tr)
Adj.list3
```

```
## [[1]]
## + 3/40 vertices, from 1b8c5cb:
## [1] 2 3 4
##
## [[2]]
## + 4/40 vertices, from 1b8c5cb:
## [1] 1 5 6 7
##
```

```
## [[3]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  1  8  9 10
##
## [[4]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  1 11 12 13
##
## [[5]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  2 14 15 16
##
## [[6]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  2 17 18 19
##
## [[7]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  2 20 21 22
##
## [[8]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  3 23 24 25
##
## [[9]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  3 26 27 28
##
## [[10]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  3 29 30 31
##
## [[11]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  4 32 33 34
##
## [[12]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  4 35 36 37
##
## [[13]]
## + 4/40 vertices, from 1b8c5cb:
## [1]  4 38 39 40
##
## [[14]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 5
##
## [[15]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 5
##
## [[16]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 5
##
## [[17]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 6
##
## [[18]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 6
##
```

```
## [[19]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 6
##
## [[20]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 7
##
## [[21]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 7
##
## [[22]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 7
##
## [[23]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 8
##
## [[24]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 8
##
## [[25]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 8
##
## [[26]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 9
##
## [[27]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 9
##
## [[28]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 9
##
## [[29]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 10
##
## [[30]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 10
##
## [[31]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 10
##
## [[32]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 11
##
## [[33]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 11
##
## [[34]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 11
##
```

```
## [[35]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 12
##
## [[36]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 12
##
## [[37]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 12
##
## [[38]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 13
##
## [[39]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 13
##
## [[40]]
## + 1/40 vertex, from 1b8c5cb:
## [1] 13
```

2) Edge list

```
Ed.list1 = as.data.frame(as_edgelist(g))
Ed.list1
```

```
##       V1    V2
## 1   Adam  Judy
## 2   Adam Bobby
## 3   Adam   Sam
## 4   Judy Bobby
## 5   Judy Frank
## 6  Bobby Frank
## 7  Bobby   Tom
## 8    Sam Frank
## 9    Sam Jerry
## 10   Sam   Jay
## 11 Frank   Tom
## 12 Frank Jerry
## 13   Tom Jerry
## 14   Tom   Jay
```

```
Ed.list2 = as.data.frame(as_edgelist(dg))
Ed.list2
```

```
##       V1     V2
## 1     KL  CHINA
## 2     KL London
## 3  CHINA London
## 4 London  CHINA
```

```
Ed.list3 = as.data.frame(as_edgelist(gb))
Ed.list3
```

```
##    V1 V2
## 1   1 11
## 2   3 11
## 3   6 11
## 4  10 11
## 5   6 12
## 6   7 12
## 7   8 12
## 8  10 12
## 9   1 13
## 10  6 13
## 11  7 13
## 12  1 14
## 13  2 14
## 14  5 14
## 15  8 14
## 16  9 14
## 17  2 15
## 18  3 15
## 19  5 15
## 20  7 15
## 21  8 15
## 22  9 15
```

3) Adjacency Matrix

```
Adj.M1 = as_adjacency_matrix(g)
Adj.M1
```

```
## 8 x 8 sparse Matrix of class "dgCMatrix"
##       Adam Judy Bobby Sam Frank Tom Jerry Jay
## Adam     .    1     1   1     .   .     .   .
## Judy     1    .     1   .     1   .     .   .
## Bobby    1    1     .   .     1   1     .   .
## Sam      1    .     .   .     1   .     1   1
## Frank    .    1     1   1     .   1     1   .
## Tom      .    .     1   .     1   .     1   1
## Jerry    .    .     .   1     1   1     .   .
## Jay      .    .     .   1     .   1     .   .
```

```
Adj.M2 = as_adjacency_matrix(dg)
Adj.M2
```

```
## 3 x 3 sparse Matrix of class "dgCMatrix"
##        KL CHINA London
## KL      .     1      1
## CHINA   .     .      1
## London  .     1      .
```

Graph Manipulation

Among the important techniques of graph manipulation are:

1. remove specific nodes/vertices.
2. generate subgraph.
3. join graphs.
4. modify the nodes data.
5. modify the edge data.

```r
library(igraph)
g = graph_from_literal(1-2, 1-3, 1-7, 3-4, 2-3, 2-4, 3-5, 4-5,
                       4-6, 4-7, 5-6, 5-8, 6-7, 7-8)
V(g)$name = c('Adam', 'Judy', 'Bobby', 'Sam', 'Frank', 'Tom', 'Jerry',
              'Jay')
g
```

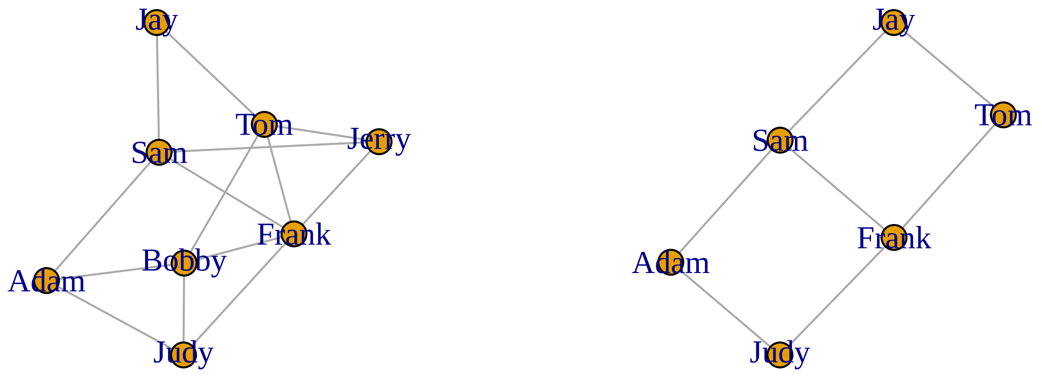
```
## IGRAPH 1c3f4ba UN-- 8 14 --
## + attr: name (v/c)
## + edges from 1c3f4ba (vertex names):
##  [1] Adam --Judy  Adam --Bobby Adam --Sam   Judy --Bobby Judy --Frank
##  [6] Bobby--Frank Bobby--Tom   Sam  --Frank Sam  --Jerry Sam  --Jay
## [11] Frank--Tom   Frank--Jerry Tom  --Jerry Tom  --Jay
```

Remove Specific Nodes/Vertices

```r
h = g- vertices(c('Jerry', 'Bobby'))
h
```

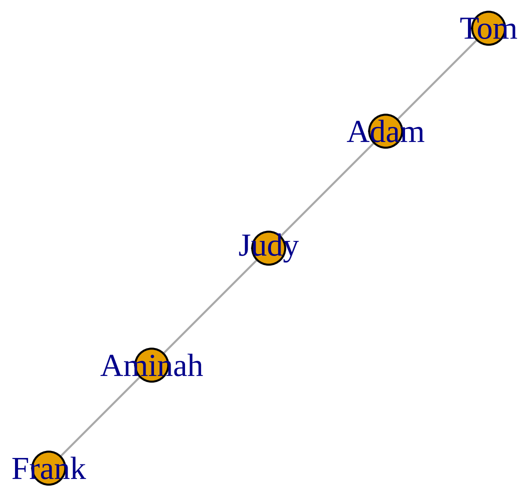
```
## IGRAPH 1c41577 UN-- 6 7 --
## + attr: name (v/c)
## + edges from 1c41577 (vertex names):
## [1] Adam --Judy  Adam --Sam   Judy --Frank Sam  --Frank Sam  --Jay
## [6] Frank--Tom   Tom  --Jay
```

```r
par(mfrow=c(1,2))
plot(g)
plot(h)
```



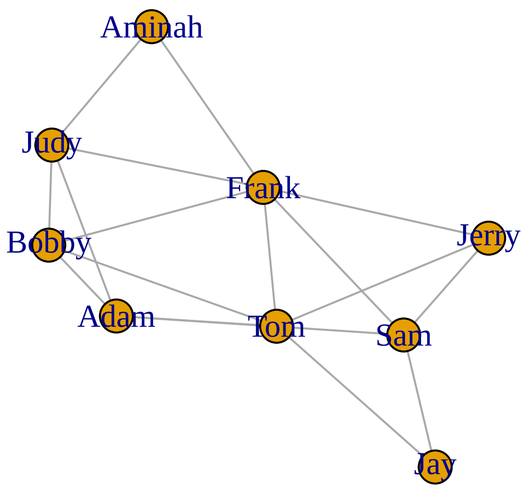
Generate Subgraph

```r
h2 = graph_from_literal('Adam'-'Judy', 'Adam'-'Tom', 'Judy'-'Aminah',
                        'Aminah'-'Frank')
plot(h2)
```



Join Graph

```r
h3 = union(h2, g)
plot(h3)
```



Modify the Nodes Data

```
V(h3)
```

```
## + 9/9 vertices, named, from 1c49663:
## [1] Adam   Judy   Tom    Aminah Frank  Bobby  Sam    Jerry  Jay
```

```
V(h3)$gender = c('male', 'female', 'male', 'female', 'male', 'male', 'male',
                 'male', 'female')
h3
```

```
## IGRAPH 1c49663 UN-- 9 17 --
## + attr: name (v/c), gender (v/c)
## + edges from 1c49663 (vertex names):
##  [1] Sam   --Jay    Sam   --Jerry  Frank --Jerry  Frank --Sam    Frank --Bobby
##  [6] Aminah--Frank  Tom   --Jay    Tom   --Jerry  Tom   --Bobby  Tom   --Frank
## [11] Judy  --Bobby  Judy  --Frank  Judy  --Aminah Adam  --Sam    Adam  --Bobby
## [16] Adam  --Tom    Adam  --Judy
```

```
vertex_attr(h3)
```

```
## $name
## [1] "Adam"   "Judy"   "Tom"    "Aminah" "Frank"  "Bobby"  "Sam"    "Jerry" 
## [9] "Jay"   
## 
## $gender
## [1] "male"   "female" "male"   "female" "male"   "male"   "male"   "male"  
## [9] "female"
```

Modify the Edge Data

```r
E(h3)
```

```
## + 17/17 edges from 1c49663 (vertex names):
##  [1] Sam   --Jay    Sam   --Jerry  Frank --Jerry  Frank --Sam    Frank --Bobby
##  [6] Aminah--Frank  Tom   --Jay    Tom   --Jerry  Tom   --Bobby  Tom   --Frank
## [11] Judy  --Bobby  Judy  --Frank  Judy  --Aminah Adam  --Sam    Adam  --Bobby
## [16] Adam  --Tom    Adam  --Judy
```

```r
E(h3)$type = c('email', 'phone', 'FB', 'email', 'class', 'Twitter', 'neighbor',
               'phone', 'FB', 'email', 'class', 'neighbor', 'phone', 'email',
               'email', 'FB', 'neighbor')
```

```r
edge_attr(h3)
```

```
## $type
##  [1] "email"    "phone"    "FB"       "email"    "class"    "Twitter"
##  [7] "neighbor" "phone"    "FB"       "email"    "class"    "neighbor"
## [13] "phone"    "email"    "email"    "FB"       "neighbor"
```

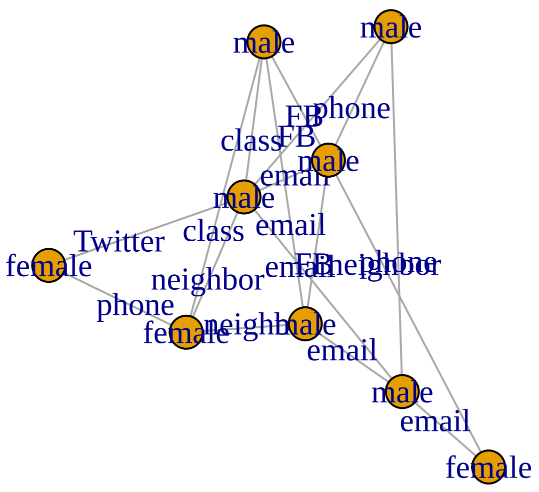
```r
E(h3)$weight = c(10,1,3,2,2,2,1,5,9,8,1,6,2,9,3,10,7)
```

```r
edge_attr(h3)
```

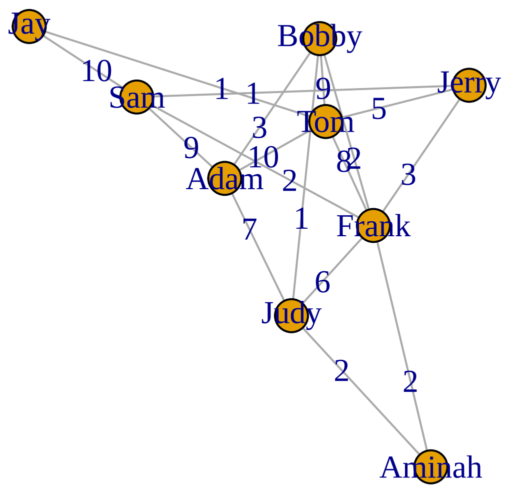
```
## $type
##  [1] "email"    "phone"    "FB"       "email"    "class"    "Twitter"
##  [7] "neighbor" "phone"    "FB"       "email"    "class"    "neighbor"
## [13] "phone"    "email"    "email"    "FB"       "neighbor"
##
## $weight
##  [1] 10  1  3  2  2  2  1  5  9  8  1  6  2  9  3 10  7
```

Graph Visualization

```r
plot(h3, vertex.label=V(h3)$gender, edge.label = E(h3)$type)
```



```r
plot(h3, vertex.label=V(h3)$name, edge.label = E(h3)$weight)
```



Node Prominence Analysis

```r
library(statnet)
```

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

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

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

```
##
## 'network' 1.19.0 (2024-12-08), 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 object is masked from 'package:HyperG':
##
##     has.loops
```

```
## 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
```

```
##
## 'ergm' 4.7.5 (2024-11-06), part of the Statnet Project
## * 'news(package="ergm")' for changes since last version
## * 'citation("ergm")' for citation information
## * 'https://statnet.org' for help, support, and other information
```

```
## 'ergm' 4 is a major update that introduces some backwards-incompatible
## changes. Please type 'news(package="ergm")' for a list of major
## changes.
```

```
## Loading required package: networkDynamic
```

```
##
## 'networkDynamic' 0.11.5 (2024-11-21), part of the Statnet Project
## * 'news(package="networkDynamic")' for changes since last version
## * 'citation("networkDynamic")' for citation information
## * 'https://statnet.org' for help, support, and other information
```

```
## Registered S3 method overwritten by 'tergm':
##   method                   from
##   simulate_formula.network ergm
```

```
##
## 'tergm' 4.2.1 (2024-10-08), part of the Statnet Project
## * 'news(package="tergm")' for changes since last version
## * 'citation("tergm")' for citation information
## * 'https://statnet.org' for help, support, and other information
```

```
##
## Attaching package: 'tergm'

## The following object is masked from 'package:ergm':
##
##     snctrl

## Loading required package: ergm.count

##
## 'ergm.count' 4.1.2 (2024-06-15), part of the Statnet Project
## * 'news(package="ergm.count")' for changes since last version
## * 'citation("ergm.count")' for citation information
## * 'https://statnet.org' for help, support, and other information

## Loading required package: sna

## Loading required package: statnet.common

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

## The following object is masked from 'package:ergm':
##
##     snctrl

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

## sna: Tools for Social Network Analysis
## Version 2.8 created on 2024-09-07.
## 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

## Loading required package: tsna

##
## 'statnet' 2019.6 (2019-06-13), part of the Statnet Project
## * 'news(package="statnet")' for changes since last version
## * 'citation("statnet")' for citation information
## * 'https://statnet.org' for help, support, and other information

## unable to reach CRAN
```

install UserNetR from github

```r
library(devtools)
```

```
## Loading required package: usethis
```

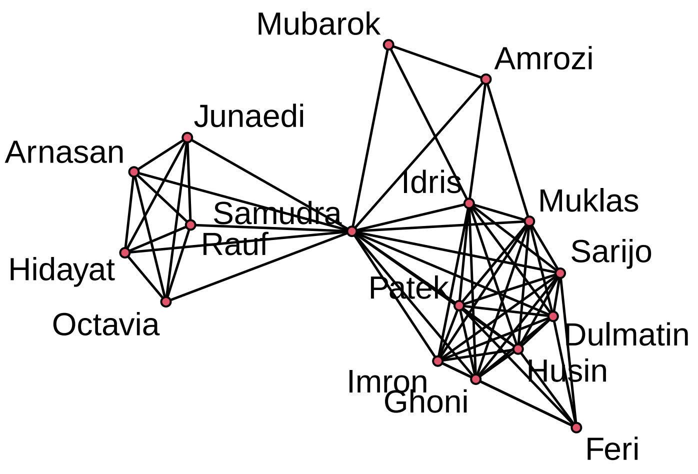
```r
install_github('DougLuke/UserNetR')
```

```
## Using GitHub PAT from the git credential store.

## Skipping install of 'UserNetR' from a github remote, the SHA1 (0888dd2b) has not changed since last install.
##   Use `force = TRUE` to force installation
```

```r
library(UserNetR)
```

```r
data(Bali)
#par(mar=c(1,1,1,1))
plot(Bali, displaylabels = T)
```



```r
Bali
```

```
##  Network attributes:
##   vertices = 17
##   directed = FALSE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 63
##     missing edges= 0
##     non-missing edges= 63
##
##  Vertex attribute names:
##     role vertex.names
##
##  Edge attribute names:
##     IC
```

Node names

```
name = Bali%v%'vertex.names'
name
```

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

Node roles

```
Role = Bali%v%'role'
Role
```

```
##  [1] "CT" "OA" "OA" "CT" "BM" "CT" "OA" "BM" "BM" "SB" "TL" "TL" "TL" "TL" "BM"
## [16] "SB" "BM"
```

Edge attribute

```
Attr = Bali%e%'IC'
Attr
```

```
##  [1] 2 2 1 1 5 1 1 1 1 2 4 5 3 3 3 3 3 3 1 3 2 5 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 1 3
## [39] 2 2 2 2 2 3 2 1 2 3 1 3 2 2 2 2 2 2 2 2 2 2 1 3 1
```

Prominence Node Measurement:

Degree Centrality

```
deg = degree(Bali)
deg
```

```
##  [1] 18  8 18 30 18 20  6 18 18 10 10 10 10 10 18 12 18
```

Closeness Centrality

```
cls = closeness(Bali)
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
```

Betweenness Centrality

```
btw = betweenness(Bali)
btw
```

```
##  [1]   4.6666667   0.6666667   3.3333333 122.3333333   3.3333333  12.3333333
##  [7]   0.0000000   3.3333333   3.3333333   0.0000000   0.0000000   0.0000000
## [13]   0.0000000   0.0000000   3.3333333   0.0000000   3.3333333
```

Eigenvector Centrality Scores

Information Centrality Scores

Flow Betweenness Scores

Centralization

Cutpoints