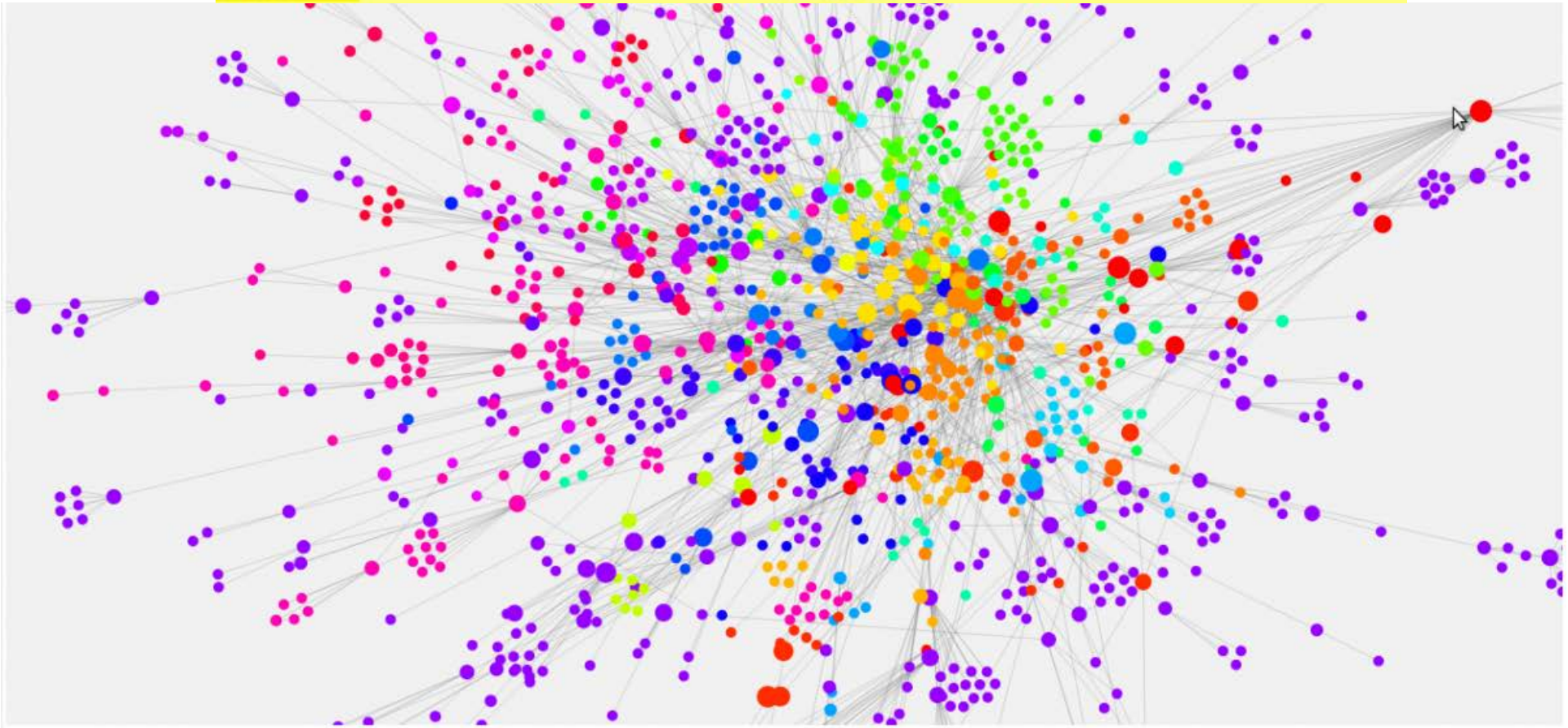
# Social Network Analysis

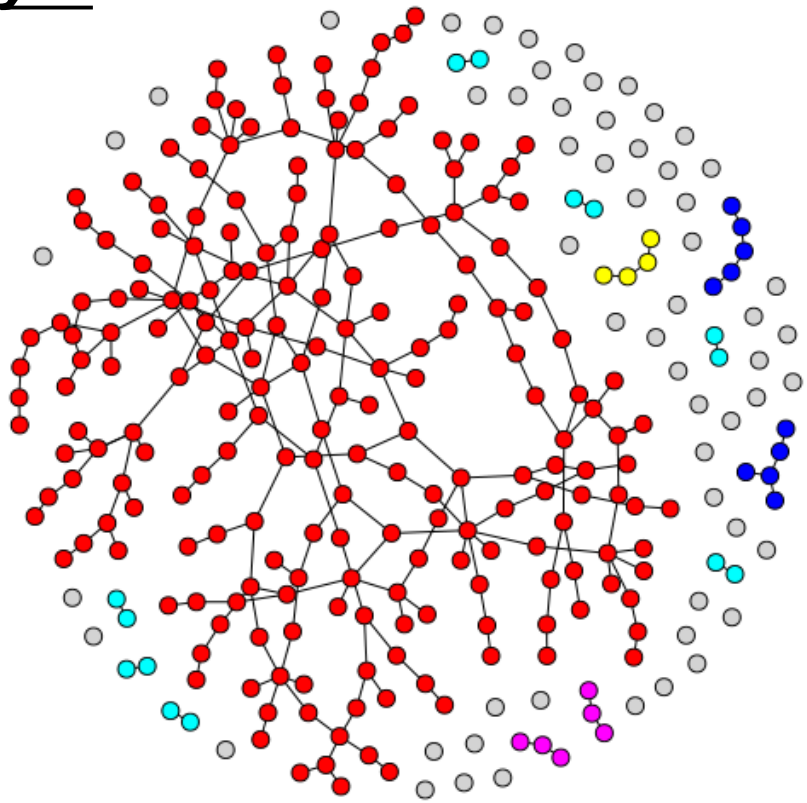
## Part 1



1. Manipulating Network with igraph
2. Manipulating Network with ggnetwork
3. Interactive Graph Drawing
4. Network Visualizations of Complex Relationships Among Variables

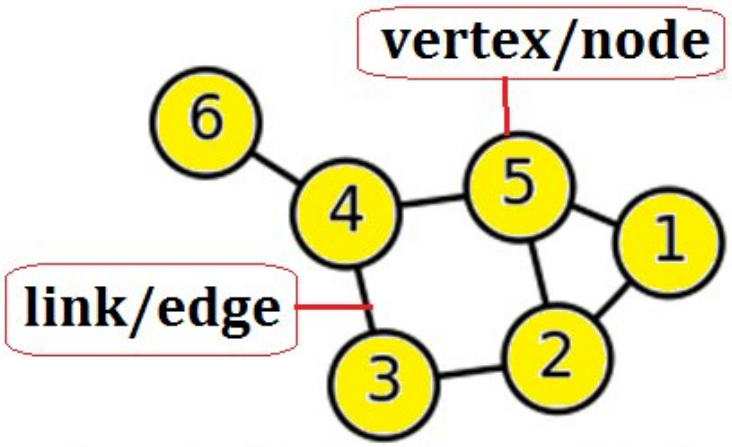
# R packages for social network analysis

- igraph
- network
- sna
- ggnetwork
- qgraph



## Graph

A graph G=(V,E) is a binary relation between elements of a set.




A graph with 6 vertices and 7 edges

# 1. Manipulating Network with igraph

For creating, decorating, and assessing basic properties of network graphs, **igraph** is particularly useful.

## (1) Vertex and Edge Attributes in igraph

### Vertex Attributes

| | |
|---|---|
| vertex.color | Node color |
| vertex.frame.color | Node border color |
| vertex.shape | none, circle, square, rectangle, pie, raster, or sphere |
| vertex.label.color | color of the name labels |
| vertex.label.font | font of the name labels |
| vertex.label.cex | size of the font of the labels |

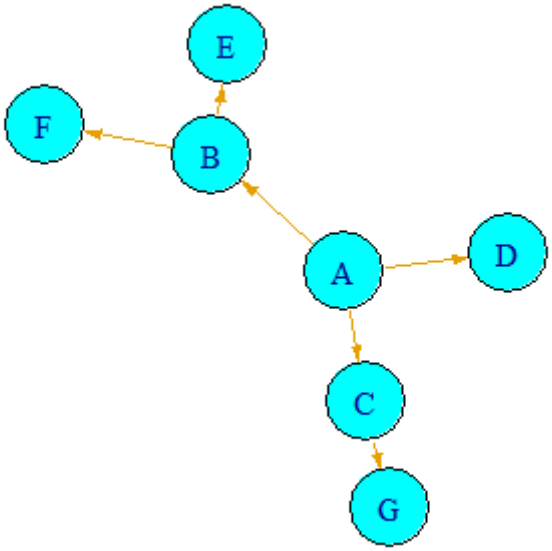
### Edge Attributes

| | |
|---|---|
| edge.color | color of the edges |
| edge.width | width of the edges |
| edge.arrow.size | size of the edge arrow |

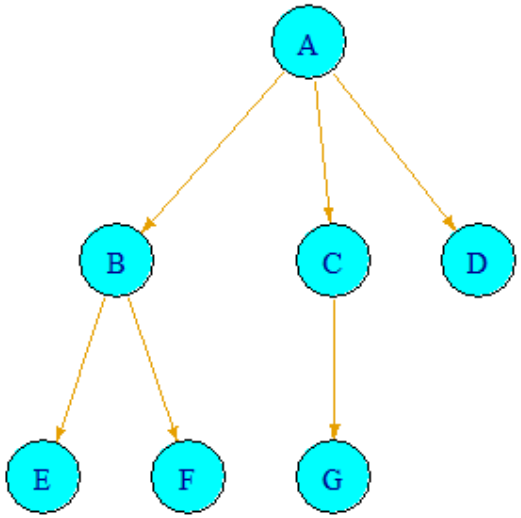
**graph_from_literal**(..., simplify = TRUE) {igraph}
Creating (small) graphs via a simple interface
It is useful if you want to create a small (named) graph quickly,
it works for both directed and undirected graphs.

```
library(igraph)
gt <- graph_from_literal(A-+B,A-+C,A-+D,B-+E,B-+F,C-+G)
igraph.options(vertex.size=35,edge.arrow.size=0.4,edge.color=1)
plot(gt, layout=layout.auto, vertex.color='cyan')
```

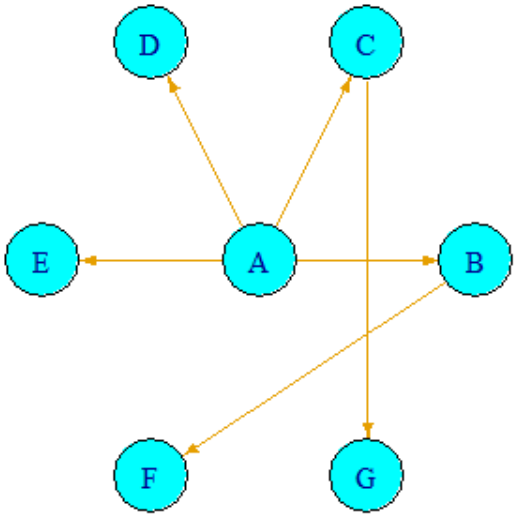
*layout.auto*




*layout_as_tree(gt)*




*layout_as_star(gt)*



# (2) Creating Graph from Data Frame Containing Edge List

**graph.data.frame**(d, ... ) {igraph}

This function creates an igraph graph from one or two data frames containing the edge list and edge/vertex attributes.

http://www.r-bloggers.com/network-visualization-in-r-with-the-igraph-package/

```
> web = "http://www.dimiter.eu/Data_files/edgesdata3.txt"
> dat <-read.table(web, header=TRUE)
> head(dat)
  first second grade spec
1    AA     DD     6    Y
2    AB     DD     6    R
3    AF     BA     6    Q
4    DD     DA     6    Q
5    CD     EC     6    X
6    DD     CE     6    Y
> gdf <- graph.data.frame(dat)
> vcount(gdf) #vertex
[1] 52
> ecount(gdf) #edge
[1] 290
```

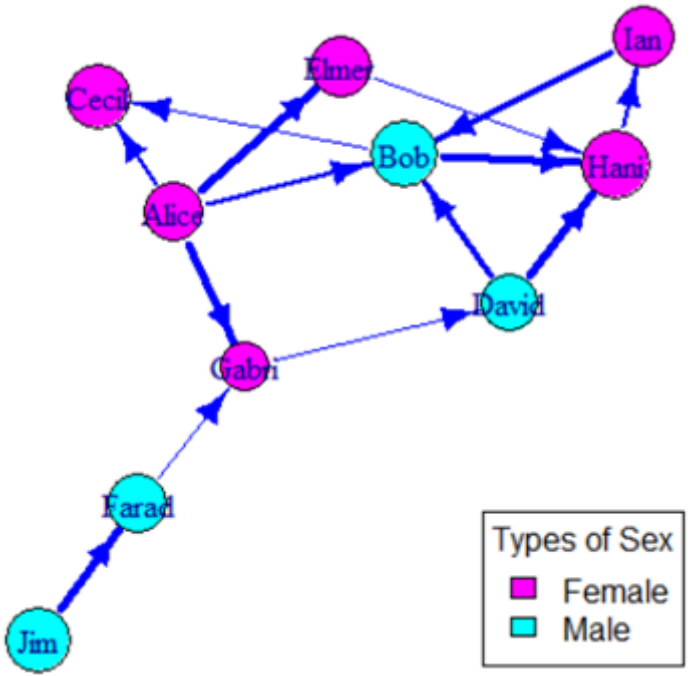

```
plot(gdf,edge.arrow.size=0.2,vertex.label.cex=0.7,
 vertex.size=15, vertex.color='ivory',
 edge.arrow.size=0.5,edge.color="deepskyblue")
```

# (3) Creating Graph from Two Datasets with Edge and Vertex Lists



**graph_from_data_frame**(d, directed, vertices) {igraph}
This function creates an igraph graph from one or two data frames containing the edge list and edge/vertex attributes.

```
library(igraph)
actor <- read.table("actor.txt",header=TRUE)
relation <- read.table("relation.txt",header=TRUE)
net <- graph_from_data_frame(relation,directed=TRUE,vertices=actor)
V(net)[sex=='M']$color <- 'cyan'
V(net)[sex=='F']$color <- 'magenta'
V(net)$size <- 0.8*V(net)$age
E(net)$width <- 0.8*E(net)$friendship
kam <- layout.kamada.kawai(net)
plot(net,layout=kam,vertex.label=V(net)$name,edge.color='blue')
legend('bottomright',1,0,title="Types of Sex",
       legend=c("Female","Male"),fill=c("magenta","cyan"))
```




"actor.txt"

| name | age | sex |
|---|---|---|
| Alice | 25 | F |
| Bob | 28 | M |
| Cecil | 27 | F |
| David | 24 | M |
| Elmer | 26 | F |
| Farad | 25 | M |
| Gabri | 21 | F |
| Hani | 29 | F |
| Ian | 26 | F |
| Jim | 27 | M |

"relation.txt"

| from | to | friendship |
|---|---|---|
| Alice | Bob | 3 |
| Alice | Elmer | 5 |
| Alice | Cecil | 3 |
| Alice | Gabri | 5 |
| Bob | Cecil | 2 |
| Bob | Hani | 5 |
| David | Hani | 5 |
| David | Bob | 4 |
| Elmer | Hani | 2 |
| Farad | Gabri | 2 |
| Gabri | David | 2 |
| Hani | Ian | 3 |
| Ian | Bob | 4 |
| Jim | Farad | 5 |

# (4) Message Channels of the Network

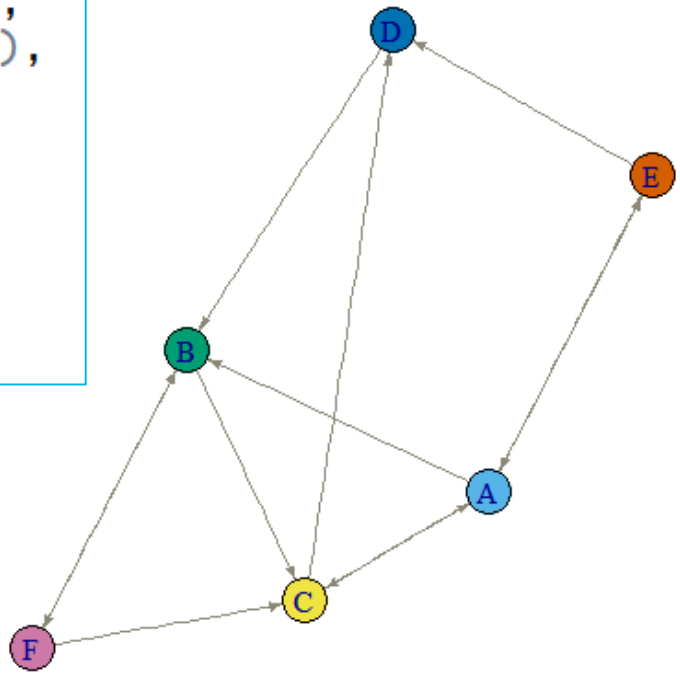
`graph.adjacency {igraph}` creates a graph from an adjacency matrix.

$A \rightarrow B,\ A \rightarrow C,\ A \rightarrow E$
$B \rightarrow C,\ B \rightarrow F$
$C \rightarrow A,\ C \rightarrow D$
$D \rightarrow B$
$E \rightarrow A,\ E \rightarrow D$
$F \rightarrow B,\ F \rightarrow C$

**Indicator matrix:**

$$N = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

```
N = matrix(c(0,1,1,0,1,0, 0,0,1,0,0,1, 1,0,0,1,0,0,
             0,1,0,0,0,0, 1,0,0,1,0,0, 0,1,1,0,0,0),
           nrow=6, byrow=TRUE)
lab = LETTERS[1:6]
dimnames(N) <- list(lab, lab)
gn <- graph.adjacency(N)
plot(gn,vertex.color=2:7,vertex.size=15,
     edge.color="cornsilk4",edge.arrow.size=0.3)
```



# 2. Manipulating Network with ggnetwork

The **ggnetwork** package provides a way to build network plots with ggplot2.

**ggnetwork**(x, ... ) {ggnetwork} Fortify network objects.
A wrapper for the fortify.network and fortify.igraph functions that will also try to coerce matrices and data frames to network objects.

## (1) Sample Data 1 : flo

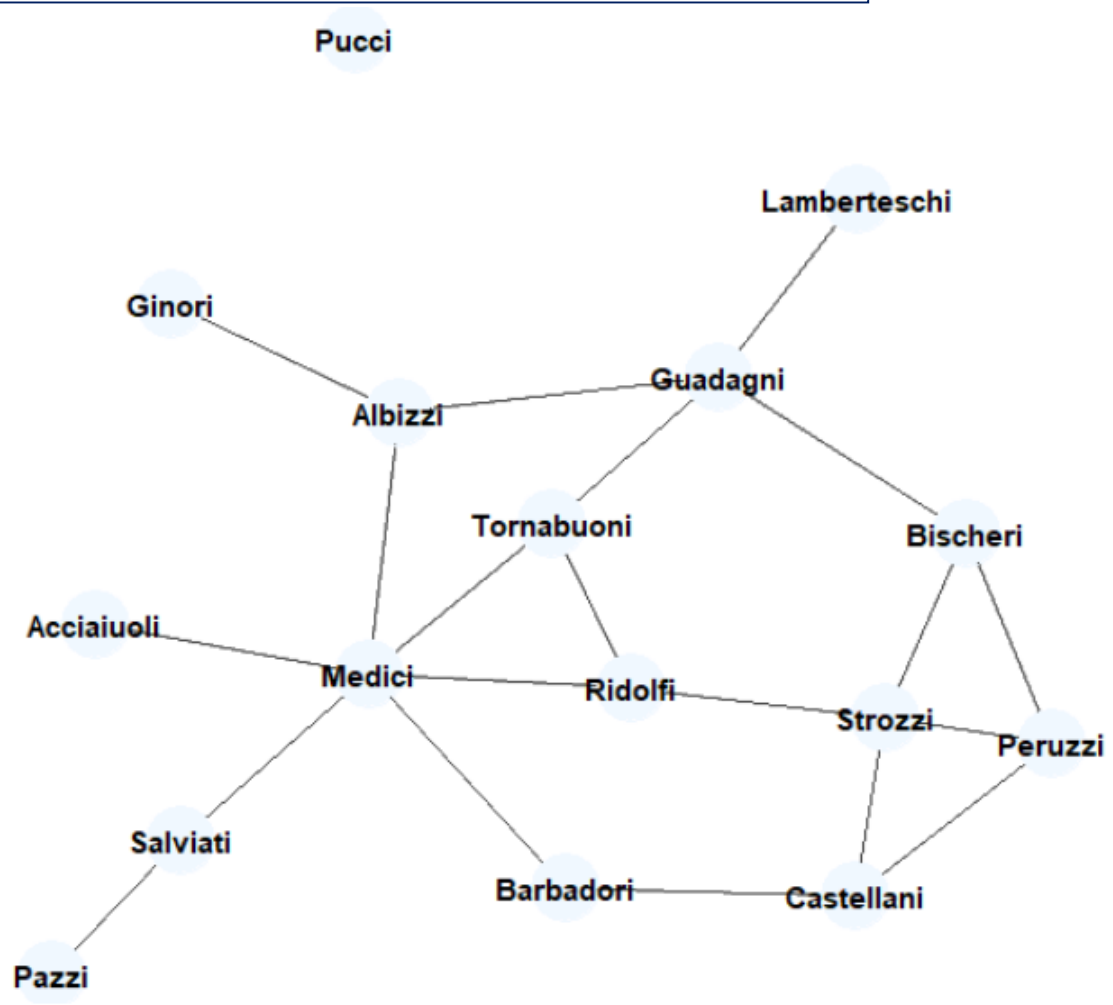
data(**flo**) {network}
This is a data set of Padgett (1994), consisting of weddings among leading Florentine families. This data is stored in symmetric adjacency matrix form.
[**Source**] J. F. Padgett, *Marriage and Elite Structure in Renaissance Florence* (1994)1282-1500.

```
> data(flo, package='network')
> head(flo,2)
           Acciaiuoli Albizzi Barbadori Bischeri Castellani Ginori Guadagni
Acciaiuoli          0       0         0        0          0      0        0
Albizzi             0       0         0        0          0      1        1
           Lamberteschi Medici Pazzi Peruzzi Pucci Ridolfi Salviati Strozzi Tornabuoni
Acciaiuoli            0      1     0       0     0       0        0       0          0
Albizzi               0      1     0       0     0       0        0       0          0
```

# Network plot with ggnetwork

```
library(ggnetwork)
gnet <- ggnetwork(flo)
library(ggplot2)
ggplot(gnet, aes(x, y, xend=xend, yend=yend)) +
    geom_edges(alpha=0.5) +
    geom_nodes(size=12, color="aliceblue") +
    geom_nodetext(aes(label=vertex.names), fontface="bold") +
    theme_blank()
```

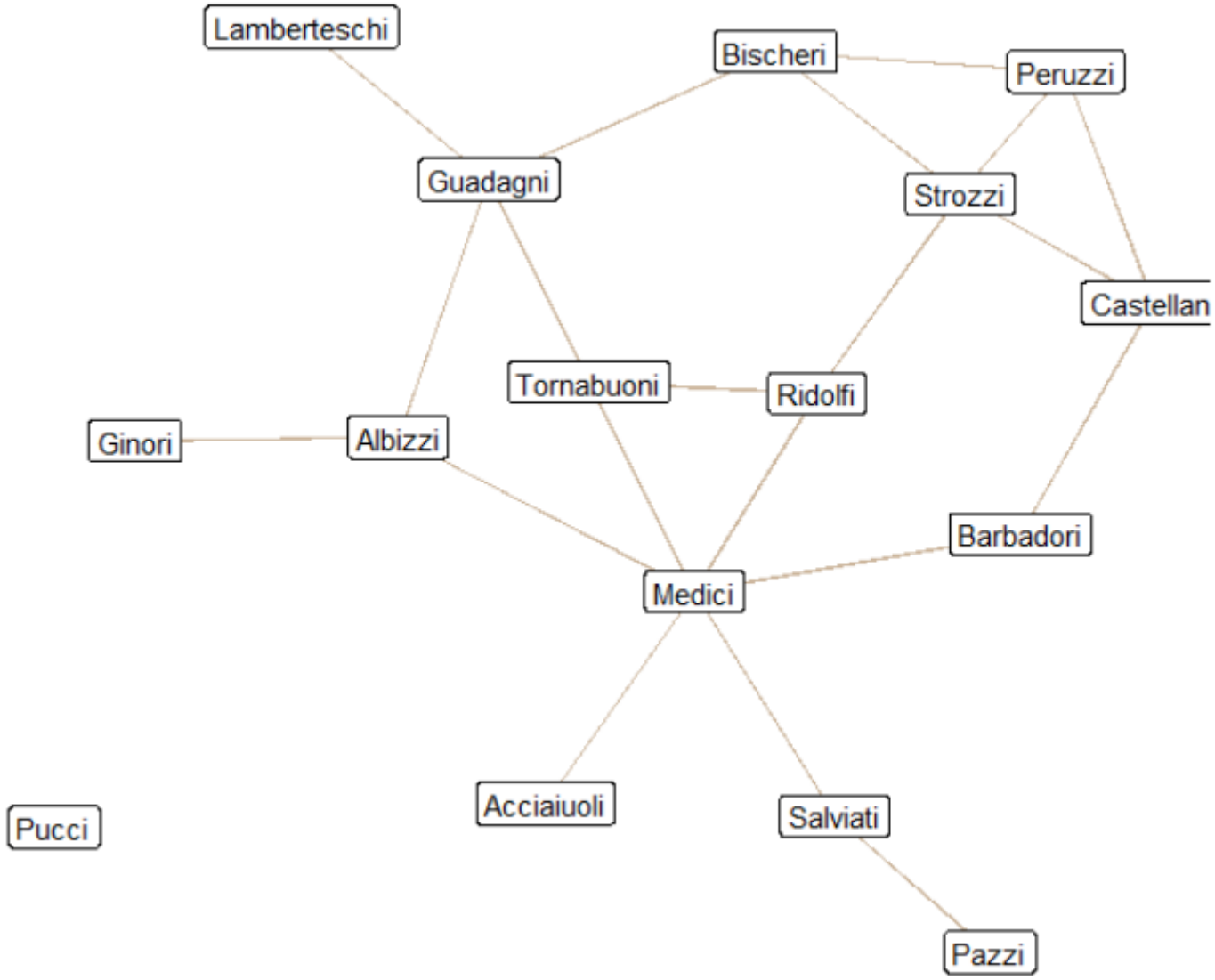
*x: starting point of x*
*xend: end point of x*



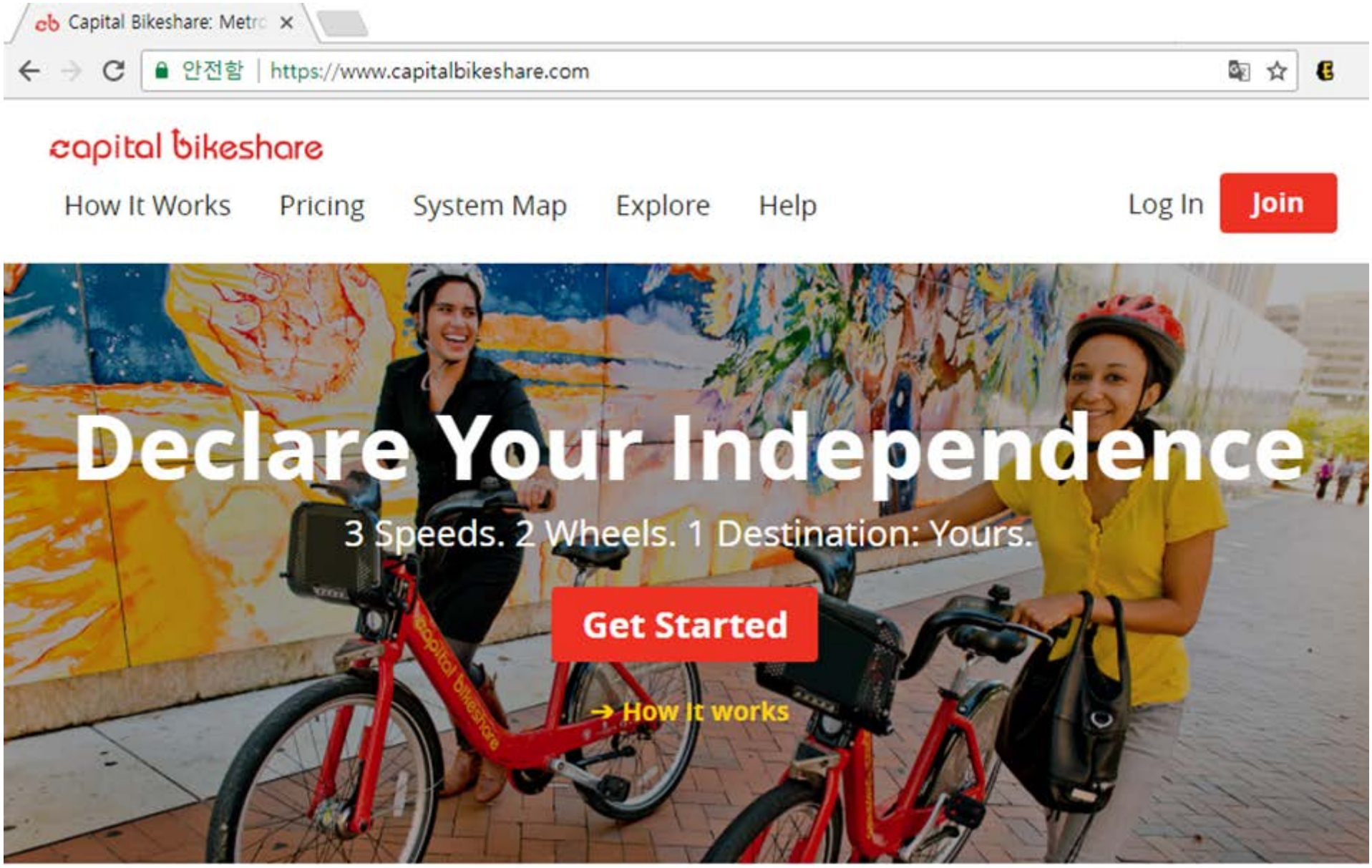
# Network plot with ggnetwork



```
ggplot(gnet, aes(x, y, xend=xend, yend=yend)) +
    geom_edges(color="bisque3") + theme_blank() +
    geom_nodelabel(aes(label=vertex.names))
```



## (2) Sample Data 2 : bikes



## **bikes** {geomnet} *Bike sharing network (directed)*

This network is a summary of the bike trips taken by customers of the bike sharing company Capital Bikeshare (https://secure.capitalbikeshare.com/) during the second quarter of 2015.

A list of two data frames:

- the trips data set consists of four variables of length 53:
  - Start.station: Station where bike trip starts
  - End.station: Station where bike trip ends
  - n: Number of trips between the two stations
  - minlength: Duration of shortest trip between the two stations (in seconds).
- the vertices data set consists of five variables with information on 21 stations:
  - id: Station ID number
  - name: Station name
  - lat: Latitude of station location
  - long: Longitude of station location
  - nbDocks: Number of bike docks at the station

```
> data(bikes, package = 'geomnet')
> head(bikes$trips,2)
                      Start.station                      End.station  n minlength
24228 Crabbs Branch Way & Calhoun Pl Crabbs Branch Way & Redland Rd 11   231.760
24232 Crabbs Branch Way & Calhoun Pl   Needwood Rd & Eagles Head Ct 14   464.435
> head(bikes$station,2)
     id                              name      lat      long nbDocks
253 261   Fallsgrove Blvd & Fallsgrove Dr 39.09631 -77.19267      19
254 262 Traville Gateway Dr & Gudelsky Dr 39.09378 -77.20250      19
```

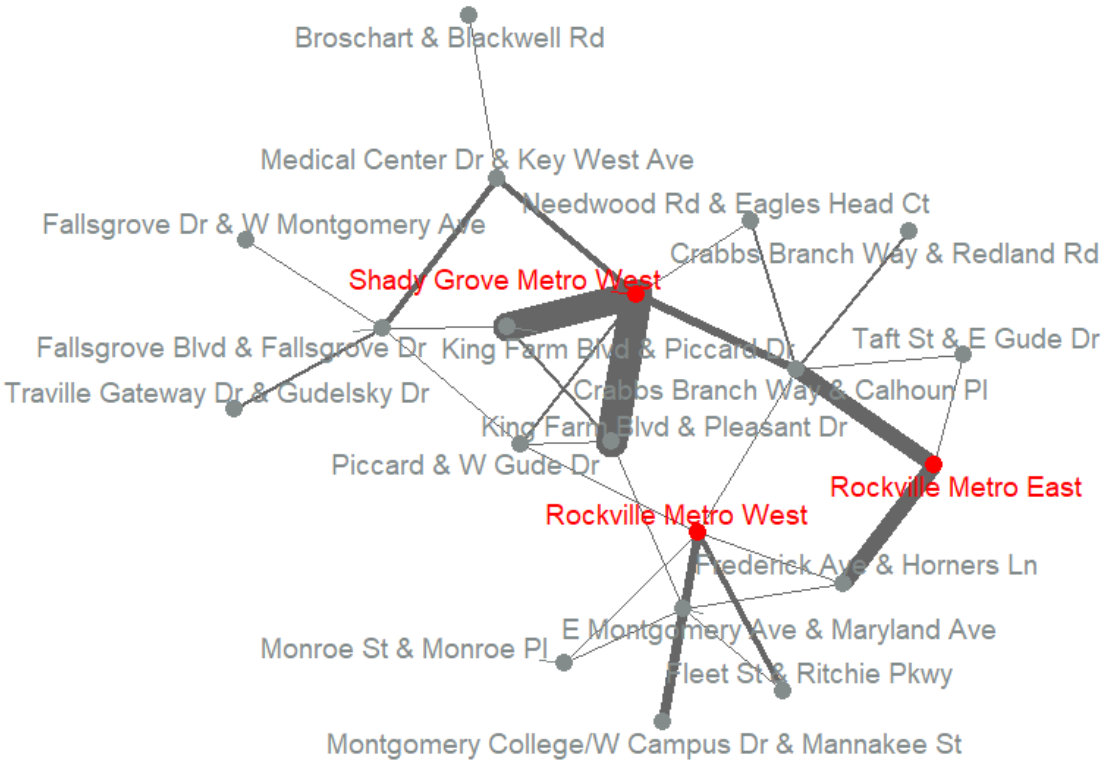
**fortify**(model, data, ...) {ggplot2}
Fortify a model with data.
model : model or other R object to convert to data frame
data : original dataset, if needed

**as.edgedf**(dat) {geomnet} Cast a data frame to an edgedf
Create and assign a new class for use of geomnet's fortify.edgedf function.

**grep**(pattern, x, ... ) {base}
Pattern Matching and Replacement

```
> library(ggnetwork)
> tripnet <- fortify(as.edgedf(bikes$trips), bikes$stations[,c(2,1,3:5)])
Using Start.station as the from node column and End.station as the to node column.
Joining edge and node information by from_id and name respectively.
> # create variable to identify Metro Stations
> tripnet$Metro = FALSE
> idx <- grep("Metro", tripnet$from_id)
> tripnet$Metro[idx] <- TRUE
> head(tripnet,3)
                          from_id                                to_id  n minlength  id
1       Broschart & Blackwell Rd                                 <NA> NA         NA 291
2 Crabbs Branch Way & Calhoun Pl Crabbs Branch Way & Redland Rd 11   231.760 281
3 Crabbs Branch Way & Calhoun Pl   Needwood Rd & Eagles Head Ct 14   464.435 281
       lat      long nbDocks Metro
1 39.10210 -77.20032      15 FALSE
2 39.10771 -77.15207      15 FALSE
3 39.10771 -77.15207      15 FALSE
```

**geom_net**(mapping, data, labelon, repel, ... ) {geomnet}
The net geom is used visualize networks within the ggplot2 framework.

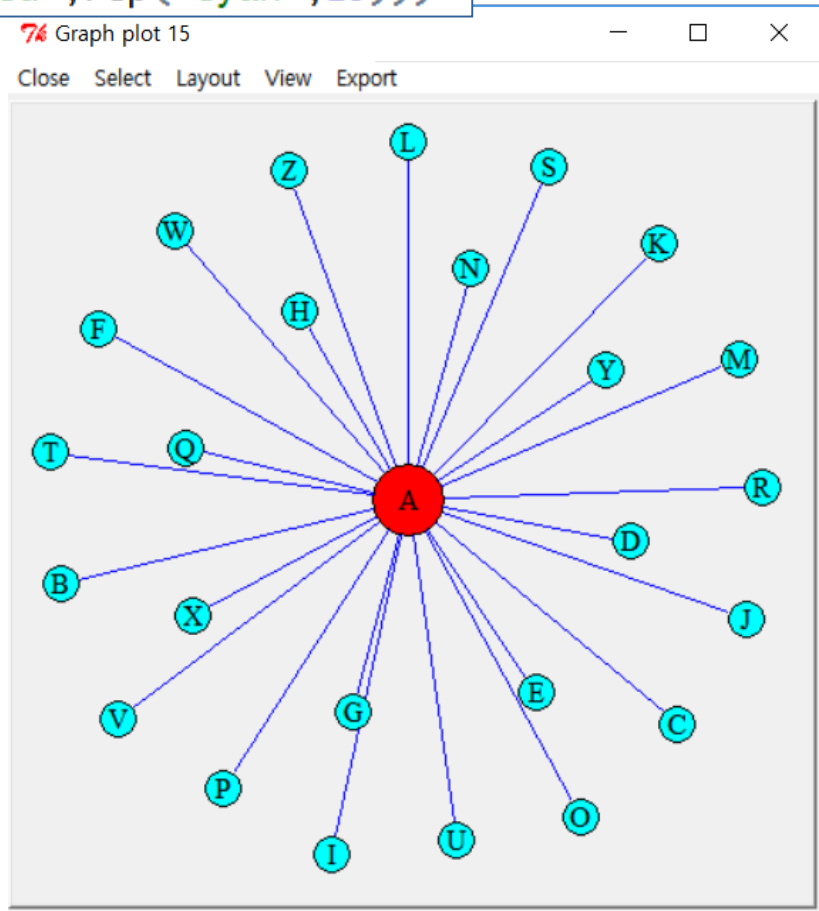



```
# plot the bike sharing network
ggplot(aes(from_id=from_id, to_id=to_id), data=tripnet) +
  geom_net(aes(linewidth=n/15,colour=Metro),labelon=TRUE,repel=TRUE) +
  theme_net() + xlim(c(-0.1, 1.1)) +
  scale_colour_manual("Metro Station", values=c("azure4","red")) +
  theme(legend.position="bottom")
```


# 3. Interactive Graph Drawing

**tkplot**(graph, canvas.width=450, canvas.height=450, ...) {igraph}
tkplot and its companion functions serve as an interactive graph drawing facility.

```
library(igraph)
net <- graph.star(26,mode="undirected")
V(net)$name <- LETTERS[1:26]
V(net)$size <- ifelse(V(net)$name=="A",20,10)
tkplot(net,edge.color="blue",vertex.label.color="black",
       vertex.label=V(net)$name,vertex.color=c("red",rep("cyan",25)))
```



# 4. Network Visualizations of Complex Relationships Among Variables

The **qgraph** package provides an interface to visualize data through network modeling techniques.

**qgraph**(input, ... ) {qgraph}
This function automatically creates an appropriate network and sends it to the plotting method.

## [Example 1] USairpollution

data(**USairpollution**) {HSAUR3}
Air pollution data of 41 US cities.
SO2 : SO2 content of air in micrograms per cubic metre.
temp : average annual temperature in Fahrenheit.
manu : number of manufacturing enterprises employing 20 or more workers.
popul : population size (1970 census); in thousands.
wind : average annual wind speed in miles per hour.
precip : average annual precipitation in inches.
predays : average number of days with precipitation per year.
[**Source**] R. R. Sokal and F. J. Rohlf, Biometry 2/e (W. H. Freeman, San Francisco, 1981)

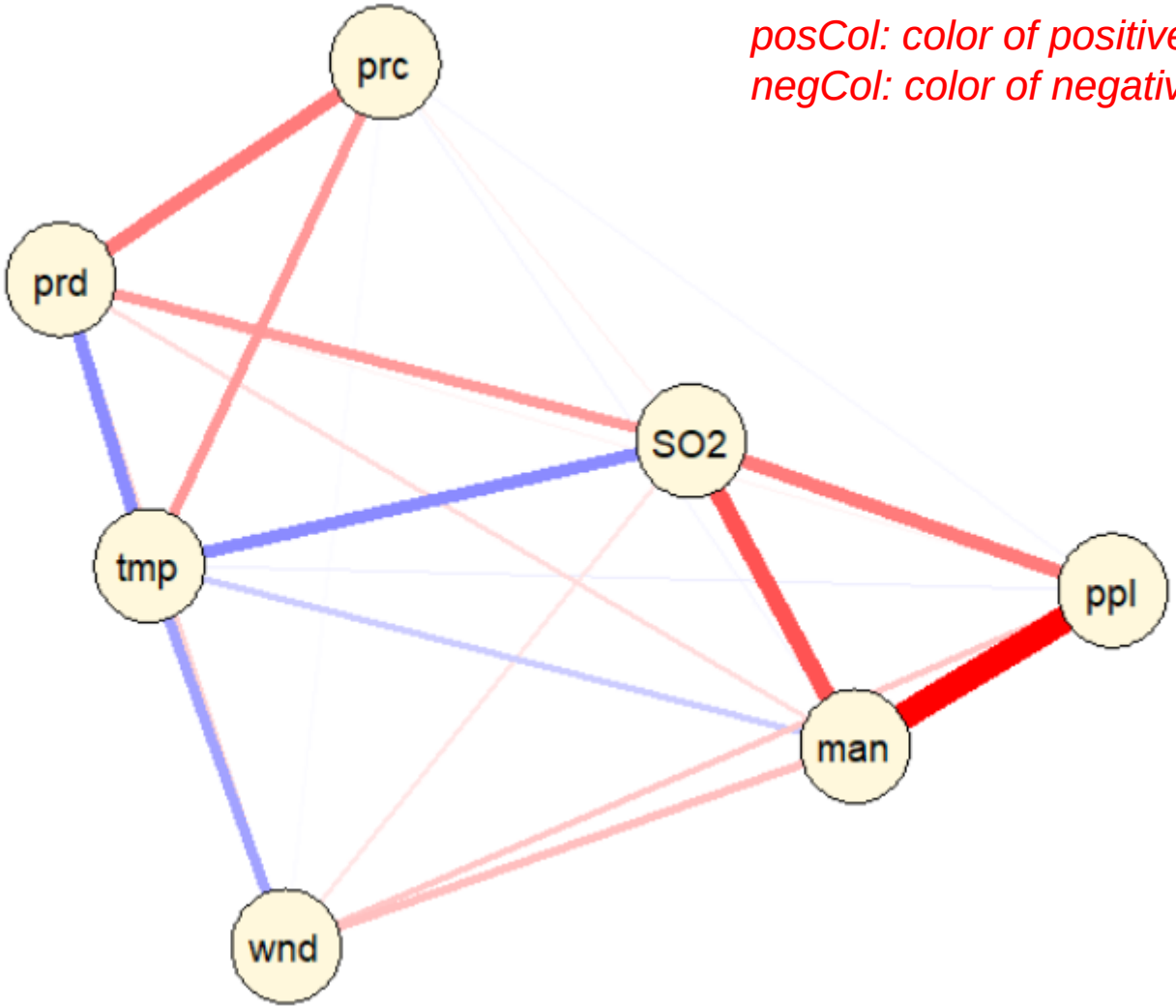
```
> data(USairpollution, package='HSAUR3')
> head(USairpollution,2)
            SO2 temp manu popul wind precip predays
Albany       46 47.6   44   116  8.8  33.36     135
Albuquerque  11 56.8   46   244  8.9   7.77      58
```

**cor_auto**(data, select, ... ) {qgraph}

Automatically compute an apppropriate correlation matrix based on pychoric, polyserial and/or Pearson correlations.

```
library(qgraph)
cor_usa <- cor_auto(USairpollution)
qgraph(cor_usa,layout='spring',color='cornsilk',posCol='red',negCol='blue')
```

*posCol: color of positive edges*
*negCol: color of negative edges*



# [Example 2] bfi

data(**bfi**) {psych}
25 personality self report items taken from the International Personality Item Pool.
Three additional demographic variables (sex, education, and age) are also included.

A1: Am indifferent to the feelings of others.
A2: Inquire about others' well-being.
A3: Know how to comfort others.
A4: Love children.
A5: Make people feel at ease.
C1: Am exacting in my work.
C2: Continue until everything is perfect.
C3: Do things according to a plan.
C4: Do things in a half-way manner.
C5: Waste my time.
E1: Don't talk a lot.
E2: Find it difficult to approach others.
E3: Know how to captivate people.
E4: Make friends easily.
E5: Take charge.

N1: Get angry easily.
N2: Get irritated easily.
N3: Have frequent mood swings.
N4: Often feel blue.
N5: Panic easily.
O1: Am full of ideas.
O2: Avoid difficult reading material.
O3: Carry the conversation to a higher level.
O4: Spend time reflecting on things.
O5: Will not probe deeply into a subject.

gender: Males = 1, Females =2
education: 1=HS, 2=finished HS, 3=some college, 4=college graduate, 5=graduate degree
age: age in years

```
> library(psych)
> data(bfi); bfin=bfi[,1:25]
> dim(bfin); head(bfin,2)
[1] 2800   25
      A1 A2 A3 A4 A5 C1 C2 C3 C4 C5 E1 E2 E3 E4 E5 N1 N2 N3 N4 N5 O1 O2 O3 O4 O5
61617  2  4  3  4  4  2  3  3  4  4  3  3  3  4  4  3  4  2  2  3  3  6  3  4  3
61618  2  4  5  2  5  5  4  4  3  4  1  1  6  4  3  3  3  3  5  5  4  2  4  3  3
```

```
library(qgraph)
cor_bfi <- cor_auto(bfin)
cols <- c(rep("aquamarine",5),rep("magenta",5),
          rep("yellow",5),rep("ivory",5),rep("cyan",5))
qgraph(cor_bfi,layout="spring",posCol='red', negCol='blue',
       minimum=0.25,color=cols)
```

*minimum: only those edges of absolute strength at least 0.25 will be visualized*