Institute of Demographics and Socioeconomics - Research Seminar
May 7, 2015 – Geneva, Switzerland

The Rsocialdata.network extension: handling and analysing egocentric network survey data in R

beta release

Emmanuel Rousseaux

Institute of Information Systems
Institute of Demographics and Socioeconomics

Emmanuel.Rousseaux@unige.ch

Gilbert Ritschard

Institute of Demographics and Socioeconomics

Gilbert.Ritschard@unige.ch

University of Geneva
Switzerland

LIVES
Swiss National Centre of Competence in Research

Outline

The Rsocialdata project

Key points:

- Storing survey data in R

  (missing values, weighting, etc.)

- Documenting data:

  questions, values, variables, and surveys

- Saving time during analyses
- Facilitating the share of survey data

LIVES
Swiss National Centre of Competence in Research

Motivation for a network survey data extension

- Specific needs of survey data not fulfilled by other network packages

  data description, missing, values, etc.

- Storing network data in a tabular way implies a lot of work to prepare data

  VLV: $(5 \times 5) \times 4 + 5 \times 8 = 100 + 40 = 140$ network variables

Motivation for a network survey data extension

- Storing network data covariates makes preparing data even more tough

  and increase the risk of making mistakes

- This data preparation work limits the reproducibility of results

  a lot of way to proceed $\Rightarrow$ make R codes difficult to share

LIVES
Swiss National Centre of Competence in Research

Literature review

About 100 packages about 'network' available on the CRAN.

- `networks`:

  classes for representing relational data types and supports arbitrary

  vertex/edge/graph attributes

- `sna`:

  data structures, statistical tools, 2D/3D visualization

- `d3Network`:

  visualization: D3 JavaScript network, tree, dendrogram, and Sankey graphs

Literature review

- `diagram`:

  visualization of simple graphs (networks)

- `egonet`:

  small tool dealing with ego-centric network measures

- `igraph`:

  visuzalization and measures for large graphs

- `InteractiveIGraph`:

  allow to work with 'igraph' objects interactively

- ...

Literature review

Conclusion

- A lot of very specify tools

  in particular visualization and statistical analysis
- Nothing for (egocentric) network survey data
- No unified framework for storing and analysing

Outline

Overview of the package

Two new classes:

- `NetworkVariable`: designed to store network survey data
- `NetworkMetadata`: designed to store covariates describing network members

Main network measures directly available:

- Density
- In-out degree
- Centrality
- Betweeness
- etc.

Tools for facilitating network structure analyses:

- `net_extract`:
  Stem new networks by combining networks with metadata information
- `net_query`:
  Query a network variable on its structure

Outline

Loading the package

```
library(Rsocialdata)

library(Rsocialdata.network)
```

Loading data

For this example we use VLV data

```
load(file="data/vlv_geneva.rda")
```

The data contains 4 network variables and 1 metadata network variable

```
ls()
```

```
## [1] "net.conflit"   "net.influence" "net.leger"     "net.materiel"
## [5] "nmv"           "vlv"           "vlv.study"
```

Réseau de soutien émotionnel léger

```
description(net.leger)

## [1] "'X' (ligne) reçoit du soutien émotionnel de 'Y' (colonne)"
```

```
valids(net.leger)

## non selectionne             oui
##               0               1
```

```
missings(net.leger)

## PDR
##  -2
```

```
net.leger[1]
```

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

```
class(net.leger)[1]
```

```
## [1] "NetworkVariable"
```

Réseau de conflit

```
description(net.conflit)

## [1] "'X' (ligne) met en colère 'Y' (colonne)"
```

```
valids(net.conflit)

## non selectionne             oui
##               0               1
```

```
missings(net.conflit)

## PDR
##  -2
```

```
net.conflit[1]
```

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

```
class(net.conflit)[1]
```

```
## [1] "NetworkVariable"
```

Demographic data of network members

```
description(nmv)
```

```
## [1] "données socio-démographiques pour les 4 réseaux"
```

```
ncol(nmv)
```

```
## [1] 14
```

```
nrow(nmv) == nrow(vlv.study) * 5
```

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

```
nmv$id.ego[1:10]

## Description:  IDs for ego

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

```
nmv$id.alter[1:10]

## Description:  IDs for alter

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

```
nmv$sexe[1:10]

## Description:  Sexe des membres du réseau

##  [1] Masculin Feminin  Feminin  Feminin  Masculin Feminin  Masculin
##  [8] Masculin Masculin Masculin
## Levels: Masculin < Feminin
```

An example of network density analysis

Computing the density of all networks is easily done:

```
vlv.study$net.leger.density = net.density(net.leger, target = "oui")
```

For instance, we consider a network is *dense* when the density is over 0.5

```
vlv.study$net.leger.density.2 = cut(vlv.study$net.leger.density, breaks = 0.5)

frequencies("net.leger.density.2", vlv.study)
```

Coding	Missing	Label	N	N total	Percent	Percent (all)	Percent total
0		[0,0.5]	437		77.62	77.62	
1		(0.5,1]	126	563	22.38	22.38	100.00
				563			100

And we assess the impact of the family configuration on being a dense network or not:

```
leger.density.reglog <- reglog(
  formula = net.leger.density.2 ~ compclustr6hcousr,
  target= "(0.5,1]",
  data = vlv.study
)
```

Results:

	Model 1
compclustr6hcousrSon and co	1.166
compclustr6hcousrDaughter and co	0.410 *
compclustr6hcousrSiblings	0.868
compclustr6hcousrKinship	0.302 *
compclustr6hcousrSparse	0.257 ***
(Intercept)	0.414 ***

Table : *Estimated coefficients (odds ratios) , *** < 0.001, ** < 0.01, * < 0.05, + < 0.1*

Operations available for structure analysis

- action_rename
- action_remove
- condition_link
- condition_attribute

```
action_remove(x = "men", from = "sexe", direction = "all")

## Network operation 'action_remove' with parameter(s)
##    x: 'men'
##    from: 'sexe'
##    direction: 'all'
##    code: '0'
```

Manipulating the emotional support network

We start from:

```
net.leger[1]


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

```
net.leger_lien_parente <- net_extract(
  x = net.leger,
  action_rename(by = "lien.de.parente.code")
)
```

```
net.leger_lien_parente[1]
```

```
## [[1]]
##     ego S SH IU GU B
## ego   0 1  1  1  1 1
## S     1 0  1  0  0 0
## SH    0 1  0  0  0 0
## IU    0 0  0  0  1 0
## GU    0 0  0  1  0 0
## B     1 0  0  0  0 0
```

```
net.leger_without_son_all <- net_extract(
  x = net.leger,
  action_remove(x = "S", from = "lien.de.parente.code", direction = "all")
)
```

```
net.leger_without_son_all[1]
```

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

Now we combine operations:

```
net.leger_without_son_all <- net_extract(
  x = net.leger,
  action_rename(by = "lien.de.parente.code"),
  action_remove(x = "S", from = "lien.de.parente.code", direction = "all")
)
```

```
net.leger_without_son_all[1]
```

```
## [[1]]
##     ego S SH IU GU B
## ego   0 0  1  1  1 1
## S     0 0  0  0  0 0
## SH    0 0  0  0  0 0
## IU    0 0  0  0  1 0
## GU    0 0  0  1  0 0
## B     1 0  0  0  0 0
```

```
net.leger_without_son_in <- net_extract(
  x = net.leger,
  action_rename(by = "lien.de.parente.code"),
  action_remove(x = "S", from = "lien.de.parente.code", direction = "in")
)
```

```
net.leger_without_son_in[1]
```

```
## [[1]]
##     ego S SH IU GU B
## ego   0 0  1  1  1 1
## S     1 0  1  0  0 0
## SH    0 0  0  0  0 0
## IU    0 0  0  0  1 0
## GU    0 0  0  1  0 0
## B     1 0  0  0  0 0
```

```
net.leger_without_son_out <- net_extract(
  x = net.leger,
  action_rename(by = "lien.de.parente.code"),
  action_remove(x = "S", from = "lien.de.parente.code", direction = "out")
)
```

```
net.leger_without_son_out[1]
```

```
## [[1]]
##     ego S SH IU GU B
## ego   0 1  1  1  1 1
## S     0 0  0  0  0 0
## SH    0 1  0  0  0 0
## IU    0 0  0  0  1 0
## GU    0 0  0  1  0 0
## B     1 0  0  0  0 0
```

Querying the emotional support network

```
net.leger_link_son_daughter_xTOy = net_query(
  x = net.leger,
  condition_link(
    x = "S", x.from = "lien.de.parente.code",
    y = "D", y.from = "lien.de.parente.code",
    direction = "xTOy"
  )
)
```

```
net.leger_link_son_daughter_xTOy[1:10]

##  [1]    NA    NA FALSE    NA  TRUE    NA    NA    NA    NA    NA
```

We observe an NA when the son (S) or the daughter (D) is not cited in the network

```
net.leger_link_son_daughter_yTOx = net_query(
  x = net.leger,
  condition_link(
    x = "S", x.from = "lien.de.parente.code",
    y = "D", y.from = "lien.de.parente.code",
    direction = "yTOx"
  )
)
```

```
net.leger_link_son_daughter_yTOx[1:10]
```

```
##  [1]    NA    NA  TRUE    NA FALSE    NA    NA    NA    NA    NA
```

```
net.leger_link_son_daughter_both = net_query(
  x = net.leger,
  condition_link(
    x = "S", x.from = "lien.de.parente.code",
    y = "D", y.from = "lien.de.parente.code",
    direction = "both"
  )
)
```

```
net.leger_link_son_daughter_both[1:10]

##  [1]    NA    NA FALSE    NA FALSE    NA    NA    NA    NA    NA
```

Results:

```
summary(net.leger_link_son_daughter_xTOy)

##    Mode   FALSE    TRUE    NA's
## logical     100      58     405

summary(net.leger_link_son_daughter_yTOx)

##    Mode   FALSE    TRUE    NA's
## logical     102      56     405

summary(net.leger_link_son_daughter_both)

##    Mode   FALSE    TRUE    NA's
## logical     120      38     405
```

$\Rightarrow$ Son to daughter support seems to be as well as frequent than daughter to son support. In about 2/3 of cases there are a reciprocity of the emotional support.

Outline

What's next

Testing and documentating

- Testing by Myriam Girardin

  on VLV network data

- All functions are documented

  but a user guide is needed to have a quick start on the package

- Make a partnership with 2 data owners

  To have sample data embedded in the package

LIVES
Swiss National Centre of Competence in Research

What's next

Integration with the `spnet` package:

- Fixed position is needed for comparing network
- Usefull to plot several neworks on the same graphic
  emotional support, financial support, conflict

What's next

Adding network metrics developped in the NCCR LIVES

For instance IP208:

- Ambivalence ratio
- Ambivalence intensity

Thank you for your attention

Questions/remarks?