REDES Revista Hispana para el Análisis de Redes Sociales

Vol.32,#1, (2021), 96-101

http://revistes.uab.es/redes https://doi.org/10.5565/rev/redes.911

Análisis de redes egocéntricas con R (IV). Análisis multinivel

Raffaele Vacca[1]

University of Florida

RESUMEN

Este texto es el cuarto y último de la serie que constituye un taller sobre análisis de ego-redes (y/o redes personales) con R. El texto está acompañado por ejemplos de datos y los scripts de lenguaje R necesarios para realizar las actividades propuestas.

Palabras clave: *Ego-redes - Redes personales – R.*

ABSTRACT

This text is the last one of a series of four that together constitute a workshop on analysis of ego-networks (and / or personal networks) using R. The text is accompanied by data samples and the R scripts necessary to carry out the suggested activities.

Key words: *Ego-networks – Personal networks – R.*

[1] *Contacto el autor: Raffaele Vacca (r.vacca@ufl.edu). Traducción de José Luis Molina*

INTRODUCCIÓN

En esta última entrega abordamos el análisis multinivel de redes egocéntricas. Para ello es necesario consultar el artículo siguiente:

- Vacca, R. (2018). Multilevel models for personal networks: Methods and applications. *Statistica Applicata - Italian Journal of Applied Statistics*, 1, 59–97. https://doi.org/10.26398/IJAS.0030-003.

El código R se presenta a continuación.

```
############################################################ #
###       Setup                                           ====
############################################################ #

# Load packages.
library(tidyverse)
library(lme4)
library(summarytools)
library(car)

# Clear the workspace
rm(list=ls())
```

```
# Load the data
load("./Data/data.rda")

# Create data frame object for models (level-1 join)
(model.data <- left_join(alter.attr.all, ego.df, by= "ego_ID"))

## Create variables to be used in multilevel models                         ====
# ============================================================= =

# Ego-alter age homophily variable
# - - - - - - - - - - - - - - - - -

# (TRUE if alter and ego are in the same age bracket)
model.data <- model.data %>%
  mutate(alter.same.age = (alter.age.cat==ego.age.cat))

# See result
freq(model.data$alter.same.age)

# Recode: TRUE = Yes, FALSE = No
model.data <- model.data %>% mutate(alter.same.age = as.character(alter.same.age),
                                     alter.same.age = fct_recode(alter.same.age,
                                                                 Yes = "TRUE", No = "FALSE"))

# See result
freq(model.data$alter.same.age)

# Centered/rescaled versions of ego and alter age
# - - - - - - - - - - - - - - - - - - - - - - - - -

# This is done for easier interpretation of model coefficients
model.data <- model.data %>%
  # Ego age centered around its means and scaled by 5 (1 unit = 5 years)
  mutate(ego.age.cen = scale(ego.age, scale= 5),
         # Alter age category centered around its mean
         alter.age.cat.cen = scale(alter.age.cat, scale= FALSE))

# Count of family members in ego-network
# - - - - - - - - - - - - - - - - - - - - - - - - -

model.data <- model.data %>%
  group_by(ego_ID) %>%
  mutate(net.count.fam= sum(alter.fam=="Yes", na.rm=TRUE)) %>%
  ungroup(model.data)

# Center and rescale by 5 (+1 unit = 5 more family members in ego-network)
model.data <- model.data %>%
  mutate(net.count.fam.cen = scale(net.count.fam, scale=5))
```

```
############################################################ #
###       Random intercept models                                ====
############################################################ #

## m1: Variance components models                                  ====
# ============================================================ =

# Variance components model: level 1 is ties, level 2 is egos, random intercept,
# no predictor
m1 <- glmer(alter.loan ~ # Dependent variable
            (1 | ego_ID), # Intercept (1) varies in level-2 units (ego_ID)
          family = binomial("logit"), # Model class (logistic)
          data= model.data) # Data object

# View results
car::S(m1)

## m2: Add tie characteristics as predictors (level 1)              ====
# ============================================================ =

# Add alter.fam and alter.same.age as predictors

# See descriptives for the new predictors
freq(model.data$alter.fam)
freq(model.data$alter.same.age)

# Estimate the model and view results
m2 <- glmer(alter.loan ~ # Dependent variable
            alter.fam + alter.same.age + # Tie characteristics
            (1 | ego_ID), # Intercept (1) varies in level-2 units (ego_ID)
          family = binomial("logit"), # Model class (logistic)
          data= model.data) # Data object
car::S(m2)

## m3: Add ego characteristics as predictors (level 2)              ====
# ============================================================ =

# Add ego age (ego.age.cen), employment status (ego.empl.bin),
# educational level (ego.edu)

# See descriptives for the new predictors
descr(model.data$ego.age.cen)
freq(model.data$ego.empl.bin)
freq(model.data$ego.edu)

# Estimate the model and view results
m3 <- glmer(alter.loan ~ alter.fam + alter.same.age + # Tie characteristics
            ego.age.cen + ego.empl.bin + ego.edu + # Ego characteristics
```

```
          (1 | ego_ID), # Intercept (1) varies in level-2 units
         family = binomial("logit"), # Model class (logistic)
         data= model.data) # Data
car::S(m3)

## m4: Add alter characteristics as predictors (level 1)                    ====
# =============================================================== =

# Add alter sex and alter age (centered)

# See descriptives for the new predictors
freq(model.data$alter.sex)
descr(model.data$alter.age.cat.cen)

# Estimate the model and view results
m4 <- glmer(alter.loan ~ alter.fam + alter.same.age + # Tie characteristics
          ego.age.cen + ego.empl.bin + ego.edu + # Ego characteristics
          alter.sex + alter.age.cat.cen + # Alter characteristics
          (1 | ego_ID), # Intercept (1) varies in level-2 units
         family = binomial("logit"), # Model class (logistic)
         data= model.data) # Data
car::S(m4)

## m5: Add network characteristics as predictors (level 2)                  ====
# =============================================================== =

# Add count of family members in network (centered)

# See descriptives for the new predictor
descr(model.data$net.count.fam.cen)

# Estimate model and see results
m5 <- glmer(alter.loan ~ alter.fam + alter.same.age + # Tie characteristics
          ego.age.cen + ego.empl.bin + ego.edu + # Ego characteristics
          alter.sex + alter.age.cat.cen +  # Alter characteristics
          net.count.fam.cen + # Ego-network characteristics
          (1 | ego_ID), # Intercept (1) varies in level-2 units
         family = binomial("logit"), # Model class (logistic)
         data= model.data) # Data
car::S(m5)

## Plot predictor effects                                                    ====
# =============================================================== =

library(effects)

# Probability of financial support as a function of alter.fam
predictorEffects(m5, "alter.fam") %>%
  plot(ylab= "Prob(alter.loan = Yes)")
```

```
# Probability of financial support as a function of ego.edu
predictorEffects(m5, "ego.edu") %>%
  plot(ylab= "Prob(alter.loan = Yes)")

############################################################ #
###       Random slope models                                ====
############################################################ #

## m6: Random slope for alter.fam                             ====
# ==========================================================  =

# Fit model
set.seed(2707)
m6 <- glmer(alter.loan ~ alter.fam + alter.same.age + # Tie characteristics
              ego.age.cen + ego.empl.bin + ego.edu + # Ego characteristics
              alter.sex + alter.age.cat.cen +  # Alter characteristics
              net.count.fam.cen + # Ego-network characteristics
              (1 + alter.fam | ego_ID), # Both intercept (1) and alter.fam
            # slppe vary in level-2 units (ego_ID)
            family = binomial("logit"), # Model class (logistic)
            data= model.data) # Data

# Re-fit with starting values from previous fit to address convergence warnings

# Get estimate values from previous fit
ss <- getME(m6, c("theta", "fixef"))

# Refit by setting ss as starting values
m6 <- glmer(alter.loan ~ alter.fam + alter.same.age + # Tie characteristics
              ego.age.cen + ego.empl.bin + ego.edu + # Ego characteristics
              alter.sex + alter.age.cat.cen +  # Alter characteristics
              net.count.fam.cen + # Ego-network characteristics
              (1 + alter.fam | ego_ID), # Both intercept (1) and alter.fam
            start= ss, #
            # slppe vary in level-2 units (ego_ID)
            family = binomial("logit"), # Model class (logistic)
            data= model.data) # Data

# View results
car::S(m6)


############################################################ #
###       Test significance of random effects                ====
############################################################ #

## Test significance of ego-level random intercept         ====
# ========================================================== =
```

```
# Test that there is significant clustering by egos, i.e. ego-level variance
# of random intercepts is significantly higher than 0. This means comparing
# the random-intercept null model (i.e. "variance components" model) to the
# single-level null model.

# First estimate the simpler, single-level null model: m0, which is nested in m1
m0 <- glm(alter.loan ~ 1, family = binomial("logit"), data= model.data)

# Then conduct a LRT comparing deviance of m0 to deviance of m1.

# Difference between deviances.
(val <- -2*logLik(m0)) - (-2*logLik(m1))

# Compare this difference to chi-squared distribution with 1 degree of freedom.
pchisq(val, df= 1, lower.tail = FALSE)

# The same result is obtained using the anova() function
anova(m1, m0, refit=FALSE)


## Test significance of ego-level random slope for alter.fam                    ====
# ============================================================================== =

# This is done with a LRT comparing the same model with (m6) and without (m5)
# random slope for alter.fam. Note that m5 is nested in m6, that's why we can
# use LRT.

# Difference between deviances.
(val <- (-2*logLik(m5)) - (-2*logLik(m6)))

# Compare this difference to chi-squared distribution with 2 degrees of freedom.
pchisq(val, df= 2, lower.tail = FALSE)

# Same results with anova() function
anova(m6, m5, refit=FALSE)
```

REFERENCIAS

McCarty, C., Lubbers, M. J., Vacca, R., & Molina, J. L. (2019). *Conducting Personal Network Research: A Practical Guide*. New York: Guilford Publishers.

Vacca, R. (2018). "Multilevel models for personal networks: Methods and applications", *Statistica Applicata - Italian Journal of Applied Statistics*, 1, 59–97. https://doi.org/10.26398/IJAS.0030

Remitido: 08-01-2021

Aceptado: 09-01-2021