

## Modelos estocásticos orientados en el actor utilizando RSiena (III). Análisis Descriptivo.

Alejandro Espinosa-Rada<sup>1</sup>

Alvaro Uzaheta<sup>1</sup>

<sup>1</sup>ETH Zürich, Social Networks Lab

### RESUMEN

Este texto es el tercer guión de una serie de seis guiones escritos en el lenguaje y ambiente de programación R que conjuntamente constituyen la introducción al software RSiena para estimar modelos estocásticos orientados en el actor para redes dinámicas. Este tercer guión es un ejemplo de cómo realizar un análisis descriptivo antes de estimar un modelo SAOM. El texto está acompañado por el código en R para realizar los análisis.

**Palabras clave:** *Análisis de redes Sociales - Modelos estocásticos orientados en el actor - Análisis estadísticos de redes sociales - RSiena - R.*

### ABSTRACT

This article is the third document of six scripts written in the language and environment R, introducing the RSiena software to estimate the stochastic actor-oriented models for dynamic networks. This third script is an example of how to conduct a descriptive analysis before estimating a SAOM model. The text incorporated the R code into the document to analyse the data.

**Key words:** *Social Network Analysis - Stochastic actor-oriented model - Statistical analysis for social networks - RSiena - R.*

<sup>1</sup> Contacto con los autores: Alejandro Espinosa-Rada ([alejandro.espinosa@gess.ethz.ch](mailto:alejandro.espinosa@gess.ethz.ch))<sup>1</sup>

<sup>1</sup> El siguiente guión se encuentra basada en el guión oficial de la página web de SIENA (<https://www.stats.ox.ac.uk/~snijders/siena/RSienaSNADescriptives.R>).

## INTRODUCCIÓN

Los materiales para esta serie incluyen 6 guiones (*scripts*) de **RSiena** escritos en el entorno y lenguaje de programación **R** (2022) enfocado a la estadística computacional y gráficos. Los guiones se encuentran disponibles en la [página oficial](#) del programa para el análisis estadístico de datos de redes SIENA (acrónimo de *Simulation Investigation for Empirical Network Analysis*), con énfasis en redes sociales, implementado en el paquete **RSiena** y que cuenta con un manual de usuarios (Ripley et al., 2022). Para una revisión sobre aspectos teóricos y los usos contemporáneos de los modelos estocásticos orientados en el actor (SAOMs) para redes dinámicas en la comunidad hispana e internacional ver Espinosa-Rada (2022a).

Este guión es la tercera contribución de una serie de seis guiones que introducen el software **RSiena**. Este guión pretende presentar un ejemplo de análisis descriptivo antes de estimar un modelo SAOM (Snijders, 2001, 2017; Snijders, van de Bunt y Steglich, 2010). El primer guión de esta serie entrega un panorama general de cómo se estima un modelo SAOM (Espinosa-Rada, 2022b). El segundo guión, en cambio, introduce el ambiente **R** y los distintos tipos de estructuras de datos que pueden ser utilizados para importar o exportar los datos para los análisis (Espinosa-Rada, 2022c).

A continuación, se preseta en el texto los códigos a ejecutar en el programa **R** precedidos por una breve explicación. En color “( )” se encuentran las líneas de códigos o comandos a ejecutar y/o que pertenecen a funciones propias del programa. De la misma forma, líneas de códigos precedidos por **#** corresponden a comentarios y no comandos.

### Análisis descriptivos de redes en R

Actualmente existen varios paquetes para realizar análisis descriptivo de redes sociales utilizando **R**. Entre los más populares se encuentran **igraph** (Csárdi y Nepusz, 2006), **network** (Butts, 2008a) y **sna** (Butts, 2008b), todos ellos disponibles en los repositorios de CRAN. De la misma forma, el siguiente tutorial supone cierta familiaridad con análisis descriptivo de redes (como ejemplo: Wasserman y Faust, 1994; Newman, 2018) y para quienes quisieran disponer de mayor detalle sobre análisis descriptivos de redes utilizando R ver Borgatti et al. (2022).

En el caso de **igraph**, este paquete es descrito como un conjunto de rutinas para redes

simples y para analizar redes. Una de sus particularidades es que puede analizar redes relativamente grandes y provee de funciones para generar diferentes tipos de redes, visualizar redes, dispone de métodos para identificar centralidades, entre otros (Csárdi y Nepusz, 2006). Por otro lado, tanto **network** como **sna** son parte de [statnet](#) el cual es un conjunto de paquetes que permiten gestionar, explorar, analizar estadísticamente, simular y visualizar datos de redes. De forma más específica, **network** contiene un conjunto de herramientas para crear y modificar objetos para analizar redes. Los objetos creados por este paquete pueden representar un conjunto relevante de datos relacionales y de atributos (para vértices, lazos y/o grafos) (Butts, 2008a). En cambio, **sna** es otro paquete que tiene como objetivo analizar redes sociales, considerando medidas al nivel de índices de los nodos o al nivel del grafo, distancias estructurales y métodos de covarianzas, detección de equivalencias estructurales, regresión de redes, generación de grafos aleatorios y para visualizar redes en formato 2D/3D (Butts, 2008b).

### Visualización de redes utilizando R

Para visualizar redes existen otros paquetes especializados que no serán revisados en este tutorial. Algunos de estos paquetes son: **ggraph** (Pedersen, 2021) el cual es una extensión de **ggplot2** (Wickham, 2016) o **sna-helper** el cual es una extensión de **ggraph** que provee de una interfaz gráfica de usuario (GUI) para crear y vsualizar redes sociales. Otras opciones son **visNetwork** (Almende et al., 2021) para generar redes interactivas, **networksD3** (Gandrud et al., 2017) para visualizar redes en tres dimensiones o **ndtv** (Bender-deMoll, 2016) para visualizar y analizar redes dinámicas, por nombrar algunos ejemplos.

Para comenzar el siguiente tutorial, utilizaremos los paquetes **network**, **sna** y **igraph** para realizar los análisis descriptivos.

```
library( network )
library( sna )
```

```
library( igraph )
```

En particular, visualizar las redes nos permite tener una primera aproximación para identificar características relevantes de las redes tales como valores atípicos que muestran un nivel alto de grados (o rango) nodales de entrada y salida o explorar potenciales aglomeraciones en la red. Esta inspección es

particularmente útil para redes de pequeño y mediano tamaño, y tal como mencionaba Jacob Moreno en colaboración con Helen Hall Jennings (1934: 103, *traducción propia*) “la posición de un individuo puede ser mejor entendida a través de un sociograma en comparación a las ecuaciones sociométricas”

A continuación, continuaremos con el ejemplo utilizado en el tutorial previo (Espinosa-Rada, 2022c).

```
library(RSiena)
```

```
friend.data.w1 <- s501
friend.data.w2 <- s502
friend.data.w3 <- s503
drink <- s50a
smoke <- s50s
```

Posteriormente, transformaremos las matrices a objetos de clase *network*:

```
net1 <- as.network( friend.data.w1 )
net2 <- as.network( friend.data.w2 )
net3 <- as.network( friend.data.w3 )
```

En donde podemos realizar una primera visualización de la sociomatrix:

```
plot.sociomatrix( net1,drawlab = F, diaglab = F,
xlab = 'friendship t1' )
plot.sociomatrix( net2,drawlab = F, diaglab = F,
xlab = 'friendship t2' )
plot.sociomatrix( net3,drawlab = F, diaglab = F,
xlab = 'friendship t3' )
```

Algo similar puede realizarse para analizar objetos de clase *igraph*:

```
g1 <- graph.adjacency( friend.data.w1 )
g2 <- graph.adjacency( friend.data.w2 )
g3 <- graph.adjacency( friend.data.w3 )
```

Estos nuevos objetos (*network* o *igraph*) tienen ciertas propiedades que permiten utilizar las funciones asociadas a sus respectivos paquetes. En primer lugar, podemos chequear la clase de estos objetos:

```
class( net1 )
class( g1 )
```

Y podemos visualizar las redes para cada caso. Como ejemplo, es posible visualizar objetos *network* utilizando *sna*:

```
plot( net1, xlab = 'friendship t1' )
```

En el caso de objetos *igraph* agregaremos algunos parámetros adicionales dado que la visualización por defecto suele contener más información en comparación al caso de los

objetos *network*. En particular, las etiquetas de los nodos serán extraídas (i.e., `vertex.label = NA`), el tamaño de los nodos serán modificados (i.e., `vertex.size = 5`) al igual que el tamaño de las flechas (i.e., `edge.arrow.size = 0.1`):

```
plot( g1, vertex.label = NA, vertex.size = 5,
edge.arrow.size=0.1, main = 'friendship t1' )
```

Para realizar otros descriptivos, los atributos de la red pueden ser incorporados a los objetos respectivamente. Para ambos casos incorporaremos el atributo “drink” a la red:

```
# network
net1 %v% "drink" <- drink[ , 1 ]

# igraph
V(g1)$drink <- drink[ , 1 ]
```

Incorporando estos atributos a las redes permiten que queden disponibles para ser utilizados con posterioridad. Para efectos de este tutorial, esta información será utilizada para colorear la red:

```
# network/sna
plot(net1, vertex.col = "drink", xlab = 'friendship
t1')
```

```
# igraph
plot(g1, vertex.label = NA, vertex.size = 5,
edge.arrow.size=0.1, vertex.color = V(g1)$drink,
main = 'friendship t1')
```

Además de colorear los nodos, podemos cambiar su tamaño acorde a alguna característica del actor. En este caso, utilizaremos los grados nodales (de salida) de los actores:

```
deg <- rowSums( as.matrix(net1) )
```

Considérese que la función `rowSums` está definida para objetos *matrix*. El caso anterior es equivalente a trabajar con la matriz de origen de este tutorial:

```
deg <- rowSums( friend.data.w1 )
```

La diferencia es que transformamos el objeto de clase *network* (i.e., `net1`) de vuelta a un objeto *matrix* utilizando la función `as.matrix(net1)`. Para los objetos clase *igraph* esto puede realizarse con el siguiente comando:

```
get.adjacency(g1, sparse = FALSE)
```

Todo junto nos lleva al mismo resultado:

```
deg <-rowSums( get.adjacency(g1, sparse =
FALSE) )
```

Una vez calculado el grado nodal de los actores, podemos observar la distribución de los grados:

```
table( deg, useNA = 'always' )
```

Dado que disponemos de la información de los grados nodales, esta información será utilizada para visualizar la red y modificar el tamaño de los nodos acorde a dicho atributo:

```
# network/sna
plot( net1, vertex.col = "drink", vertex.cex = (deg
+ 1)/1.5 )
```

```
# igraph
plot( g1, vertex.label = NA, vertex.size = deg*2,
edge.arrow.size=0.1, vertex.color = V(g1)$drink,
main = 'friendship t1' )
```

Considérese que el parámetro `deg` fue modificado de forma arbitraria para mejorar la visualización.

## Visualización de redes longitudinales

Hasta ahora hemos visualizado la red en tiempo 1. A continuación modificaremos las redes en los siguientes momentos del tiempo. Para ello, partiremos incorporando los mismos atributos calculados para la red 1:

```
# network
net2 %v% "drink" <- drink[ , 2 ]
net3 %v% "drink" <- drink[ , 3 ]
```

```
deg2 <- rowSums( as.matrix( net2 ) )
deg3 <- rowSums( as.matrix( net3 ) )
```

```
# igraph
V(g2)$drink <- drink[ , 2 ]
V(g3)$drink <- drink[ , 3 ]
```

Para que las visualizaciones puedan ser observadas simultáneamente, crearemos un conjunto de paneles en **R**. En donde `mfrow` entrega las dimensiones del panel. En nuestro caso, esto corresponderá a una fila y tres columnas:

```
par( mfrow = c( 1, 3 ) )
```

A continuación, podremos crear tres visualizaciones una al lado de la otra en paneles consecutivos:

```
# network
plot( net1, vertex.col = "drink", vertex.cex = (deg
+ 1)/1.5 )
plot( net2, vertex.col = "drink", vertex.cex = (deg2
+ 1)/1.5 )
plot( net3, vertex.col = "drink", vertex.cex = (deg3
+ 1)/1.5 )
```

```
# igraph
```

```
plot( g1, vertex.label = NA, vertex.size = 5,
edge.arrow.size=0.1, vertex.color = V(g1)$drink )
plot( g2, vertex.label = NA, vertex.size = 5,
edge.arrow.size=0.1, vertex.color = V(g3)$drink )
plot( g3, vertex.label = NA, vertex.size = 5,
edge.arrow.size=0.1, vertex.color = V(g2)$drink )
```

Como podrá apreciar, cada vez que se visualiza una red las coordenadas de los nodos cambia debido a que para generar el algoritmo los valores de partida son asignados de forma aleatoria. Por ello, podemos guardar las coordenadas y utilizarlas para las siguientes visualizaciones:

```
par( mfrow = c( 1, 3 ) )
```

```
# network
coordin <- plot( net1, vertex.col = "drink",
vertex.cex = (deg + 1 )/1.5 )
plot( net2, coord = coordin, vertex.col = "drink",
vertex.cex = (deg2 + 1)/1.5 )
plot( net3, coord = coordin, vertex.col = "drink",
vertex.cex = (deg3 + 1) /1.5 )
```

```
# igraph
par( mfrow = c( 1, 3 ) )
coord <- layout.fruchterman.reingold( g1 )
plot( g1, vertex.label = NA, vertex.size = deg*2,
edge.arrow.size=0.1, vertex.color = V(g1)$drink,
layout = coord )
plot( g2, vertex.label = NA, vertex.size = deg*2,
edge.arrow.size=0.1, vertex.color = V(g3)$drink,
layout = coord )
plot( g3, vertex.label = NA, vertex.size = deg*2,
edge.arrow.size=0.1, vertex.color = V(g2)$drink,
layout = coord )
```

Para extraer las coordenadas utilizando las tres redes:

```
# network
coordin <- plot( net1 + net2 + net3 )
```

```
# igraph
coord <- layout.fruchterman.reingold( g1 + g2 + g3 )
```

En ambos paquetes hay varios algoritmos que se pueden utilizar para visualizar las redes. Aquí se utilizó el algoritmo de Fruchterman Reingold (opción por defecto en **network**). Para ver otras opciones disponibles:

```
# network/sna
?gplot
?gplot.layout
```

```
# igraph
?layout_
```

## Estadísticos básicos para analizar redes

Los paquetes **igraph** y **sna** pueden ser utilizados para realizar varias descripciones y análisis. Al respecto, uno de los estadísticos

importantes para caracterizar la red completa es la densidad de la red:

```
# network/sna
gden( net1 ) # density
```

```
# igraph
graph.density( g1 )
```

De la misma forma, es posible calcular la proporción de díadas que son simétricas. Esto es lo que realiza **sna** por defecto. Sin embargo, para el caso de **igraph** la misma medida puede calcularse directamente considerando el censo de díadas de esta red:

```
# network/sna
grecip( net1 )

# igraph
( dyad.census(g1)$mut+dyad.census(g1)$null
)/sum( unlist(dyad.census(g1) ) )
```

Otra opción es considerar la reciprocidad ignorando las díadas nulas:

```
# network/sna
grecip( net1, measure = "dyadic.nonnull" )

# igraph
reciprocity( g1, mode = "ratio" )
```

También podemos calcular la transitividad de la red. Sin embargo, existen ciertas diferencias en la forma en que ambos programas hacen el cálculo. Para el caso de **sna** la transitividad (Holland y Leinhardt, 1972; Wasserman y Faust, 1994) utilizada por defecto considera la proporción de triángulos en transitividades débiles (suponiendo que el nodo "a" envía un lazo a nodo "b", éste a su vez envía un lazo a nodo "c" y el nodo "a" envía también un lazo a nodo "c") en comparación con el total de caminos de distancia dos sin considerar los lazos auto-reflexivos. En cambio, **igraph** utiliza por defecto el índice de clusterización global propuesto por Watts y Strogatz (1998) el cual calcula el promedio del índice de clusterización local para cada actor de la red, el cual es entendido como la proporción de vecinos del actor "i" que se encuentran conectados en comparación con el número de pares de vecinos del mismo actor "i". No obstante, en **igraph** se utiliza una red simetrizada en donde la dirección de la relación no se considera y del cual dicha función desaconseja su utilización para el caso de redes direccionadas. Por lo tanto, los resultados y su interpretación difieren.

```
# network/sna
gtrans( net1 )

# igraph
transitivity( g1 )
```

Otra posibilidad es realizar un censo de díadas y tríadas. Sin embargo, dado que las funciones tanto en el paquete **igraph** como **sna** asignaron el mismo nombre a dichas funciones es posible que ambos paquetes entren en conflicto cuando son utilizados simultáneamente. En cuyo caso, es mejor especificar de dónde proviene la función que se quiere utilizar. Para ello, asignaremos **sna::** cuando se utilizar **sna** y **igraph::** para el caso de **igraph**:

```
# network/sna
sna::dyad.census( net1 )
sna::triad.census( net1 )
```

```
# igraph
igraph::dyad.census( g1 )
igraph::triad.census( g1 )
```

Un caso similar ocurre con los grados nodales de salida para ambos paquetes. A continuación, calcularemos la distribución de los grados nodales de salida y sus quintiles:

```
# network/sna
outdegree <- sna::degree( net1, cmode =
"outdegree" )

# igraph
outdegree <- igraph::degree( g1, mode = "out" )

outdegree

hist( outdegree )
quantile( outdegree )
```

Otro conjunto de medidas descriptivas corresponde a la conectividad y distancia de la red. Entre algunas de las medidas más utilizadas es la matriz de distancias geodésicas (distancias más cortas entre cualquier par de nodos):

```
# network/sna
dist <- geodist( net1, inf.replace = Inf, count.paths
= TRUE )
dist$gd

# igraph
distances( g1, mode="out" )
```

Para el caso de **sna**, este paquete permite explorar también la matriz de distancias geodésicas que contiene el número de geodésicas entre cada par de nodos y la matriz de accesibilidad (*reachability*):

```
# matrix of geodesic distances
dist$counts
table(dist$counts)

# reachability matrix
?reachability
reach <- reachability( net1
reach
```



## Autocorrelación de red

La última medida que revisaremos en este tutorial es la autocorrelación de Moran I, medida que se encuentra disponible en **sna**.

En la implementación actualmente disponible en **sna** la medida no se encuentra estandarizada por filas, lo que se puede interpretar como la propensión de cada actor "i" de ser influido de forma equivalente por todos los miembros de su vecindario independiente de qué tan grande sea la comunidad (detalles sobre esta corrección en Anselin, 1995). De la misma forma, en la siguiente función utilizaremos como ponderación a aquellos actores que son vecinos del actor de referencia (i.e., [2]). Para estimar la medida considerando únicamente los lazos de salida:

```
nacf(net1, drink[, 1], type="moran",
neighborhood.type='out')[2]
nacf(net2, drink[, 2], type="moran",
neighborhood.type='out')[2]
nacf(net3, drink[, 3], type="moran",
neighborhood.type='out')[2]
```

Esta medida también puede considerar tanto los lazos de salida como aquellos de entrada:

```
nacf(net1, drink[, 1], type="moran",
neighborhood.type='total')[2]
nacf(net2, drink[, 2], type="moran",
neighborhood.type='total')[2]
nacf(net3, drink[, 3], type="moran",
neighborhood.type='total')[2]
```

## Comentarios finales

En este tutorial se revisó cómo a través de una matriz es posible visualizar y realizar análisis descriptivos antes de comenzar a estimar modelos SAOMs utilizando el software **RSiena**. Durante el tutorial se utilizaron los paquetes **network**, **sna** y **igraph**. Estos paquetes suelen ser los más populares para analizar redes sociales utilizando **R**. Sin embargo, poseen ciertas diferencias que los investigadores debiesen considerar al momento de realizar los análisis. En particular, aquí ejemplificamos dicha diferencia a través del cálculo de las medidas de transitividad, medida que es estimada de forma diferente especialmente para redes direccionadas. También pudimos apreciar que hay funciones que poseen el mismo nombre y que pueden entrar en conflicto una vez que se utilizan simultáneamente y hay otras funciones que no se encuentran necesariamente disponible en ambos paquetes.

Los siguientes guiones revisarán el formato de los datos antes de utilizar **RSiena**. Los últimos dos guiones de esta serie se concentrarán principalmente en la estimación del modelo. Esperamos que esta introducción permita dar una aproximación a los modelos estocásticos basados en el actor a través del uso práctico del programa, acorde al estado actual de su desarrollo y que sea accesible a la comunidad de habla hispana.

## Agradecimientos

Agradecemos particularmente a Tom A.B. Snijders (University of Oxford y University of Groningen) por su amabilidad en permitirnos difundir, modificar y extender los siguientes guiones los cuales contaron también con contribuciones previas de Robin Gauthier, Ruth Ripley, Johan Koskinen, Paulina Preciado, Zsofia Boda y Christian Steglich. En un principio estos documentos iban ser una traducción textual a los guiones disponibles en la página de RSiena. No obstante, por sugerencia de Tom A. B. Snijders los guiones traducidos y modificados adquieren con esta contribución independencia de sus homólogos en inglés con el objeto de facilitar su difusión, mantención y responsabilidad en su contenido en español del cual esperamos reflejar adecuadamente sus intenciones originarias.

## FINANCIAMIENTO

Alejandro Espinosa-Rada contó con el apoyo del proyecto Fondecyt Regular N° 1220560 para realizar esta investigación, proyecto financiado por la Agencia Nacional de Investigación y Desarrollo (ANID, Chile).

## REFERENCIAS

- Almende, B.V., Thieurmen, B. & Robert, Titouan (2021)**. visNetwork: Network Visualization using 'vis.js' Library. R package version 2.1.0. <https://CRAN.R-project.org/package=visNetwork>
- Anselin, L. (1995)**. Local indicators of spatial association—LISA. *Geographical analysis*, 27(2), 93-115.
- Bender-deMoll, S. (2016)**. ndtv: Network Dynamic Temporal Visualization. R package version 0.10. <http://statnet.org>
- Borgatti, S., Everett, M., Johnson, J. & Agneessens (2022)**. *Analyzing Social Networks Using R*. SAGE.
- Butts, C. T. (2008a)**. network: A package for Managing Relational Data in R. *Journal of*

*Statistical Software*, 24(2), 1-36. DOI: <https://doi.org/10.18637/jss.v024.i02>

**Butts, C. T. (2008b).** Social network analysis with sna. *Journal of Statistical Software*, 24(6), 1-51. DOI: <https://doi.org/10.18637/jss.v024.i06>

**Csárdi, G., & Nepusz, T. (2006).** The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5), 1-9.

**Espinosa-Rada, A. (2022a).** "Usos contemporáneos de los modelos estocásticos orientados en el actor (SAOMs) para redes dinámicas". En: Francisca Ortiz y Alejandro Espinosa-Rada (Eds.). *El Análisis de Redes Sociales en América Latina. En prensa.*

**Espinosa-Rada, A. (2022b).** Modelos estocásticos orientados en el actor utilizando RSiena (I). Guión básico introductorio. *Revista Hispana para el Análisis de Redes Sociales*, 33(1), 100-111. DOI: <https://doi.org/10.5565/rev/redes.936>

**Espinosa-Rada, A. (2022c).** Modelos estocásticos orientados en el actor utilizando RSiena (II). Formato de los Datos. *Revista Hispana para el Análisis de Redes Sociales*, 33(1), 92-99. DOI: <https://doi.org/10.5565/rev/redes.937>

**Gandrud, C., Allaire, J.J. Rusell, K. & Yetman, C.J. (2017).** networkD3: D3 JavaScript Network Graph from R. R package version 0.4. <https://CRAN.R-project.org/package=networkD3>

**Holland, P.W., & Leinhardt, S. (1972).** Some Evidence on the Transitivity of Positive Interpersonal Sentiment. *American Journal of Sociology*, 72, 1205-1209. DOI: <https://doi.org/10.1086/225266>

**Moreno, J. (1934).** *Who Shall Survive?* Nervous and Mental Disease Publishing.

**Newman, M. (2018).** *Networks.* Oxford University Press.

**Pedersen, T. L. (2017).** ggraph: An implementation of Grammar of Graphics for Graphs and Networks. R package version 2.0.5. <https://CRAN.R-project.org/package=ggraph>

**R Core Team (2022).** *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. URL: <https://www.R-project.org/>

**Ripley, R., Snijders, T. A.B., Boda, Z. Vörös, A. & Preciado, P. (2022).** *Manual for SIENA version 4.0.* Oxford: University of Oxford, Department of Statistics, <http://www.stats.ox.ac.uk/~snijders/siena/>.

**Snijders, T.A.B. (2001)** The statistical evaluation of social network dynamics, *Sociological Methodology* - 2001, 40: 361-395.

**Snijders, T.A.B. (2017).** [Stochastic Actor-Oriented Models for Network Dynamics.](https://doi.org/10.1146/annurev-statistics-060116-054035) *Annual Review of Statistics and Its Application*, 4, 343-363.

DOI: <http://dx.doi.org/10.1146/annurev-statistics-060116-054035>

[Here is the e-print access to this article.](#)

**Wasserman, S., & Faust, K. (1994).** *Social network analysis: Methods and applications.* Cambridge University Press.

**Watts, D. J. & Strogatz, S. H. (1998).** Collective dynamics of 'small-world' networks. *Nature*, 393, 440-442. DOI: <https://doi.org/10.1038/30918>

**Wickham, H. (2016).** *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag, New York. URL: <https://ggplot2.tidyverse.org>

**Remitido:** 28-03-2022

**Corregido:** 30-03-2022

**Aceptado:** 04-04-2022

