Análisis Estadístico de Redes

Juan Sosa

2025-11-29

Contents

Introducción

Este libro presenta una introducción al análisis estadístico de redes, combinando conceptos teóricos con ejemplos prácticos en R.

Chapter 1

Conceptos preliminares

1.1 Introducción

Una **red** es una colección de objetos interconectados.

Un **grafo** es una colección de objetos (vértices o nodos) unidos por enlaces (aristas o arcos).

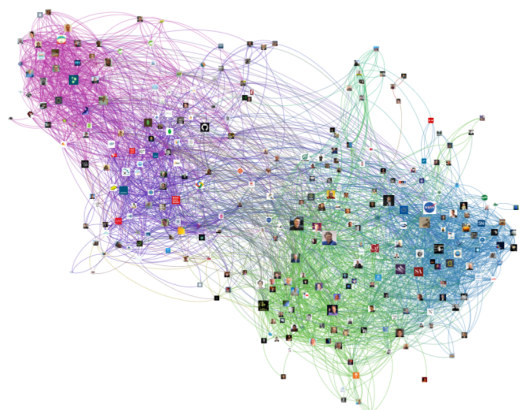
Los **objetos** se denominan comúnmente como actores, individuos, nodos o vértices; mientras que las **conexiones** entre ellos como enlaces, conexiones, aristas o arcos.

Las **interacciones** de las partes que constituyen un sistema conducen a **comportamientos colectivos** y **propiedades a nivel global**.

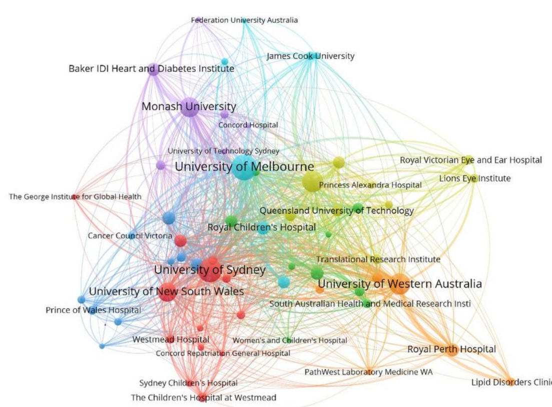
Estudiar redes es importante porque nos permite entender cómo las conexiones entre individuos configuran **sistemas complejos** como la dinámica de conflictos y alianzas, la difusión de ideas, y la propagación de epidemias, entre muchos otros.

1.2 Ejemplos

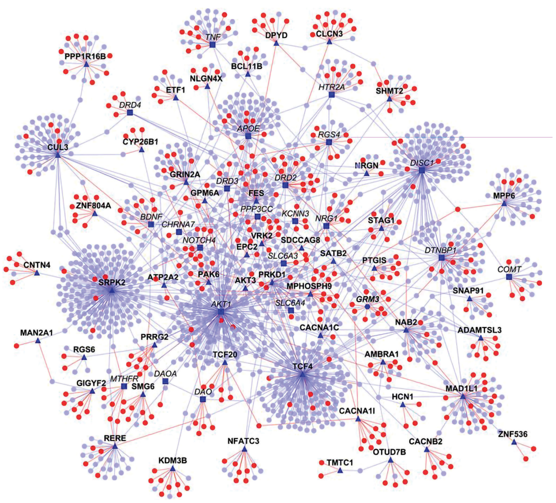
- Redes **sociales on-line**: e.g., Twitter.



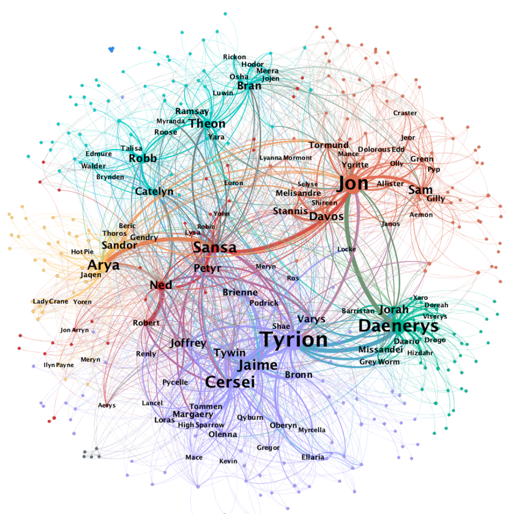
- Redes **laborales**: e.g., red de colaboración entre instituciones de investigación.



- Redes **biológicas**: e.g., red de proteínas.



- Redes de **personajes**: e.g., *Game of Thrones*.



Y **muchas más**: redes **semánticas**, redes de **información**, redes de **contagio**, redes de **transporte**, redes de **comercio**, redes de **corrupción**, etc.

1.3 Datos relacionales

Los **datos relacionales** están constituidos por una colección de **objetos**, sus **atributos** y un conjunto de **relaciones** observadas entre ellos.

La presencia de **variables diádicas** (medidas sobre pares de individuos o díadas) es la característica distintiva de los datos relacionales.

Almacenar, gestionar, caracterizar, visualizar y modelar datos relacionales utilizando:

- **Teoría de grafos**.
- **Modelos estadísticos**.
- **Algoritmos y procesos**.
- **Herramientas computacionales**.

1.4 Tipos de redes

1.4.1 Redes no dirigidas y dirigidas

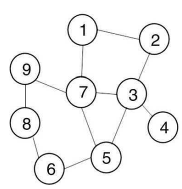
Una relación **no dirigida** (simétrica) tiene uno y solo un valor por díada.

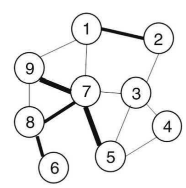
Una relación **dirigida** (asimétrica) tiene dos valores por díada, i.e., un valor para cada miembro de la pareja.

Se dice que una red es una **red no dirigida** (grafo) si todas las relaciones en ella no están dirigidas, y se denomina **red dirigida** (dígrafo) en caso contrario.

Ejemplos de relaciones **no dirigidas**:

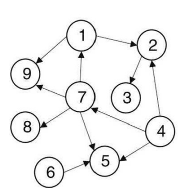
- Amistades de una red social *on-line* (e.g., Facebook).
- Cantidad de tiempo que personas pasan juntas.

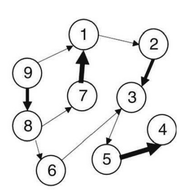




Ejemplos de relaciones **dirigidas**:

- Amistades auto-informadas.
- Número de correos electrónicos enviados entre compañeros de trabajo.





1.4.2 Redes binarias y ponderadas

Una relación **binaria** (dicotómica) únicamente asume dos valores, ausencia o presencia de la relación.

Una relación **ponderada** (numérica) toma más de dos valores para caracterizar las relaciones entre las díadas.

Ejemplos de relaciones **binarias**:

- Presencia de un conflicto militar entre países.
- Presencia de una alianza estratégica entre compañías.

Ejemplos de relaciones **ponderadas**:

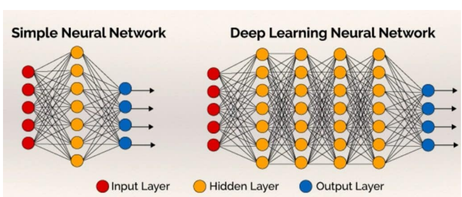
- Número de veces que congresistas apoyan un proyecto de ley juntos.
- Distancia geográfica entre ciudades.

1.5 Otros tipos de redes

1.5.1 Redes neuronales

Una **red neuronal** es un **modelo computacional** con varias **capas de nodos**, cuyo comportamiento está determinado por la forma en que se conectan los nodos y la ponderación de las conexiones.

Usualmente se utilizan para **tareas de identificación**, como el reconocimiento facial, el reconocimiento de voz y el procesamiento de lenguaje natural.

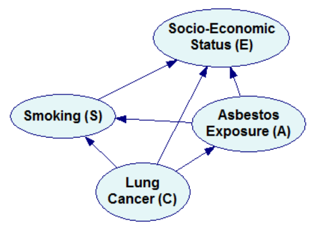


Ver por ejemplo:

- Ghatak, A. (2019). *Deep learning with R*. Springer.
- ¿Pero qué “es” una Red neuronal? de **3Blue1Brown**.

1.5.2 Redes Bayesianas

Las **redes Bayesianas** son una clase de **modelos gráficos** que permiten una representación de las **dependencias probabilísticas** entre un conjunto dado de variables aleatorias por medio de un **gráfico acíclico dirigido**.



Ver por ejemplo:

- Scutari, M. & Denis, J. B. (2021). *Bayesian networks: with examples in R*. CRC press.
- Bayesian Networks de **Bert Huang**.

1.6 Software

Los paquetes igraph (R y Python) y NetworkX (Python) proporcionan herramientas versátiles para la visualización y el análisis de redes.

Más alternativas:

- **R**: statnet, network, sna, tidygrap, ggnet2, ggraph, networkD3.
- **Python**: Graph-tool, Networkit.

1.7 Fuentes de datos

Entre muchos otros repositorios:

- Duke Network Analysis Center.
- Stanford Network Analysis Project.
- Awesome Public Datasets.
- LINQS.
- Mark Newman.
- Katya Ognyanova.

1.8 Algunas referencias

La literatura sobre análisis y ciencia de redes es vasta, con aportes tanto teóricos como computacionales.

Los siguientes libros ofrecen una síntesis desde los fundamentos matemáticos hasta la implementación práctica en `R` y `Python`:

- Kolaczyk, E. D. y Csárdi, G. (2020). ***Statistical Analysis of Network Data with R*** (2nd ed.). Springer, Cham.
- Menczer, F., Fortunato, S. y Davis, C. A. (2020). ***A First Course in Network Science***. Cambridge University Press, Cambridge.
- Newman, M. E. J. (2018). ***Networks*** (2nd ed.). Oxford University Press, Oxford.
- Al-Taie, M. Z. y Kadry, S. (2017). ***Python for Graph and Network Analysis***. Springer, Cham.
- Barabási, A.-L. y Pósfai, M. (2016). ***Network Science***. Cambridge University Press, Cambridge.
- Luke, D. A. (2015). ***A User's Guide to Network Analysis in R***. Springer, New York.

Chapter 2

Gestión de datos relacionales

2.1 Introducción

Conformación, **almacenamiento** y **gestión** de datos relacionales:

- Variables **diádicas**.
- Variables **nodales**.
- Variables **derivadas** (e.g., indicadoras de grupos).

Un **grafo** por sí solo (una colección de vértices y aristas) suele ser insuficiente para representar todos los atributos una red.

La **decoración** de un grafo corresponde a la **conjunción de vértices y aristas con otras variables** de interés (atributos).

2.2 Grafos

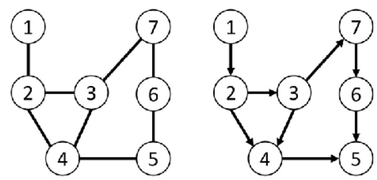
Un **grafo** $G = (V, E)$ es una estructura que consiste de un conjunto de **vértices** (nodos) V y de un conjunto de **aristas** (enlaces) E, donde los elementos de E son parejas de la forma $e = \{u, v\}$, con $u, v \in V$.

El **número de vértices** $|V|$ y el **número de aristas** $|E|$ se conocen como el **orden** y el **tamaño** del grafo, respectivamente.

Los **vértices** del grafo **se enumeran** con los números enteros $1, \dots, n$, con $n = |V|$.

2.2.1 Grafos y digrafos

Un grafo para el que cada arista $\{u, v\} \in E$ es tal que $\{u, v\} \not\equiv \{v, u\}$, para todo $u, v \in V$ se denomina **grafo dirigido** o **digrafo**. De lo contrario se llama **grafo no dirigido** o simplemente **grafo**.



2.2.2 Ejemplo: red binaria no dirigida

```
suppressMessages(suppressWarnings(library(igraph)))

# Red binaria no dirigida
g <- graph_from_literal(1-2, 1-3, 2-3, 2-4, 3-5, 4-5, 4-6, 4-7, 5-6, 6-7)
```

```
# Clase de objeto
class(g)
```

```
## [1] "igraph"
```

```
# Vértices
V(g)
```

```
## + 7/7 vertices, named, from efa1244:
## [1] 1 2 3 4 5 6 7
```

```
# Orden
vcount(g)
```

```
## [1] 7
```

```
# Aristas
E(g)
```

```
## + 10/10 edges from efa1244 (vertex names):
##  [1] 1--2 1--3 2--3 2--4 3--5 4--5 4--6 4--7 5--6 6--7
```

```r
# Tamaño
ecount(g)
```

```
## [1] 10
```

```r
# Dirigida?
is_directed(g)
```

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

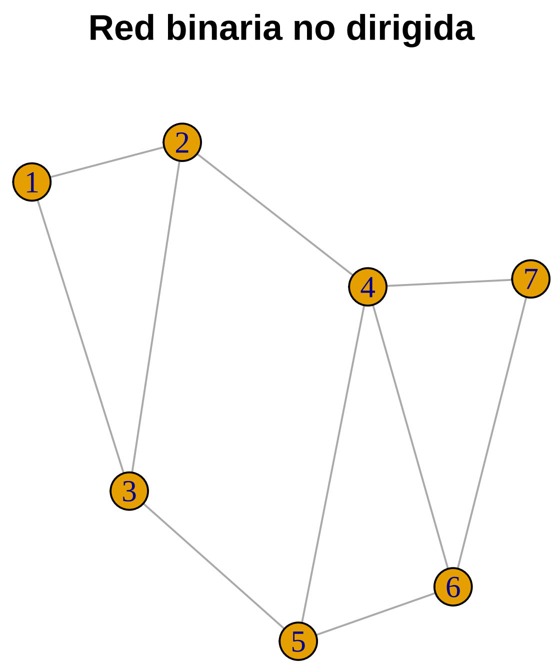
```r
# Ponderada?
is_weighted(g)
```

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

```r
# Simple?
is_simple(g)
```

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

```r
# Visualización
set.seed(123)
plot(g, main = "Red binaria no dirigida")
```



2.2.3 Ejemplo: red ponderada no dirigida

```
# Red ponderada no dirigida
wg <- g

# Pesos
set.seed(123)
(E(wg)$weight <- round(runif(ecount(g), 0, 1), 2))
```

```
##  [1] 0.29 0.79 0.41 0.88 0.94 0.05 0.53 0.89 0.55 0.46
```

```
# Dirigida?
is_directed(wg)
```

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

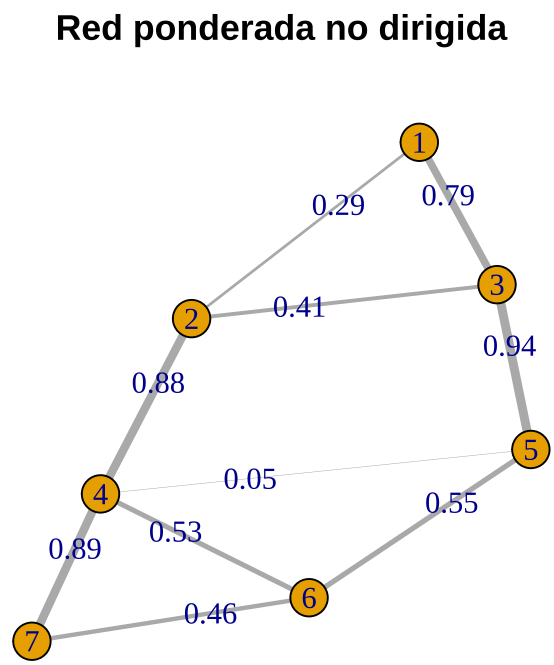
```
# Ponderada?
is_weighted(wg)
```

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

```
# Simple?
is_simple(wg)
```

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

```
# Visualización
set.seed(123)
plot(wg,
     edge.width = 5*E(wg)$weight,
     edge.label = E(wg)$weight,
     main = "Red ponderada no dirigida")
```



2.2.4 Ejemplo: red binaria dirigida

```
# Red binaria dirigida
dg <- graph_from_literal(1-+2, 1++3, 2-+3, 3++4, 4-+5, 5-+6, 6++7)
```

```
# Aristas
E(dg)
```

```
## + 10/10 edges from ad1f251 (vertex names):
##  [1] 1->2 1->3 2->3 3->1 3->4 4->3 4->5 5->6 6->7 7->6
```

```r
# Tamaño
ecount(dg)
```

```
## [1] 10
```

```r
# Dirigida?
is_directed(dg)
```

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

```r
# Ponderada?
is_weighted(dg)
```

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

```r
# Simple?
is_simple(dg)
```

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

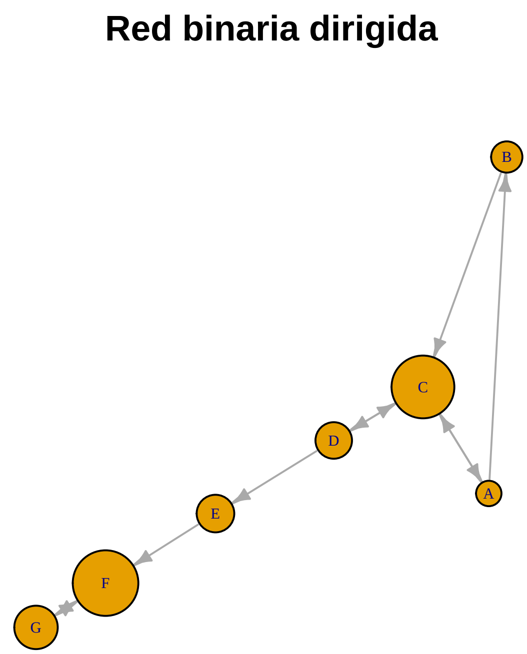
```r
# Etiquetas de los vértices
(V(dg)$name <- LETTERS[1:vcount(dg)])
```

```
## [1] "A" "B" "C" "D" "E" "F" "G"
```

```r
# Agregar 'x' como atributo
set.seed(123)
(V(dg)$x <- round(rnorm(vcount(dg), 10, 5), 2))
```

```
## [1]  7.20  8.85 17.79 10.35 10.65 18.58 12.30
```

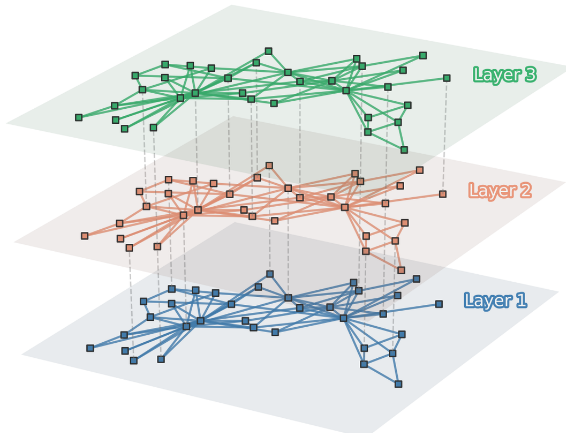
```r
# Visualización
set.seed(123)
plot(dg,
     vertex.size = 1.5*V(dg)$x,
     vertex.label.cex = 0.5,
     edge.arrow.size = 0.5,
     main = "Red binaria dirigida")
```



2.2.5 Multigrafos

Un **multigrafo** es aquel grafo que permite **múltiples aristas** entre el mismo par de vértices y aristas de un vértice a sí mismo.

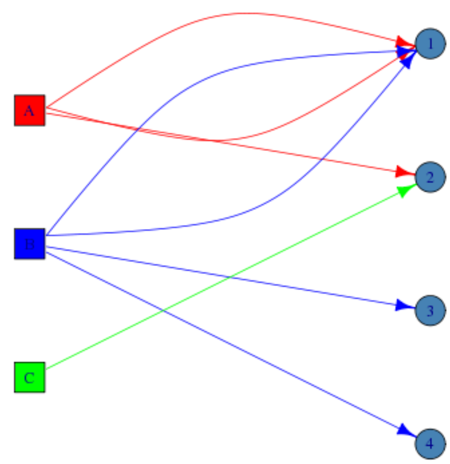
Un grafo que no es un multigrafo se denomina **grafo simple**.



2.2.6 Grafos bipartitos

Un **grafo bipartito** es un grafo $G = (V, E)$ tal que el conjunto de **vértices** V puede dividirse en dos **subconjuntos disjuntos**, V_1 y V_2, y cada **arista** en E tiene un extremo en V_1 y el otro en V_2.

Este tipo de **grafos** se usa típicamente para representar **redes de pertenencia**. Por ejemplo, con los **miembros** representados por vértices en V_1 y las correspondientes **organizaciones** por vértices en V_2.



2.3 Estructuras de datos

Generalmente los grafos **no se definen manualmente** ya que en la práctica la mayoría de las redes son **grandes**.

Los datos para construir un grafo comúnmente se almacenarán en un **archivo de datos** (`.txt`, `.csc`, `.dat`, etc.).

2.3.1 Matriz de adyacencia

La **matriz de adyacencia** o **socio-matriz** $\mathbf{Y} = [y_{i,j}]$ asociada con un grafo $G = (V, E)$ de orden n es una **matriz binaria** de $n \times n$ tal que $y_{i,j} = 1$ si $\{i, j\} \in E$ y $y_{i,j} = 0$ en otro caso.

La **diagonal principal** de una matriz de adyacencia está llena de **ceros estructurales**.

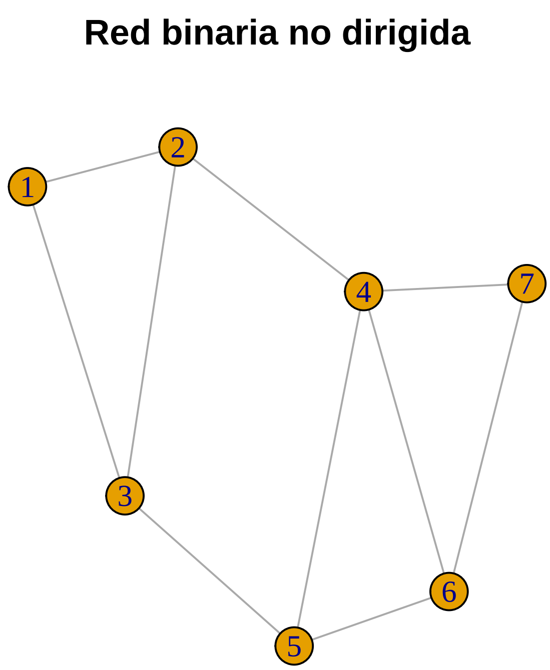
La matriz de adyacencia de un **grafo no dirigido** es **necesariamente simétrica**.

La matriz de adyacencia de un **grafo dirigido** es **posiblemente asimétrica**.

2.3.2 Ejemplo: red binaria no dirigida

```r
# Red binaria no dirigida
g <- graph_from_literal(1-2, 1-3, 2-3, 2-4, 3-5, 4-5, 4-6, 4-7, 5-6, 6-7)
```

```r
# Visualización
set.seed(123)
plot(g, main = "Red binaria no dirigida")
```




```r
# Matriz de adyacencia
(A <- as_adjacency_matrix(g))
```

```
## 7 x 7 sparse Matrix of class "dgCMatrix"
```

```
##   1 2 3 4 5 6 7
## 1 . 1 1 . . . .
## 2 1 . 1 1 . . .
## 3 1 1 . . 1 . .
## 4 . 1 . . 1 1 1
## 5 . . 1 1 . 1 .
## 6 . . . 1 1 . 1
## 7 . . . 1 . 1 .
```

```
# Clase de objeto
class(A)
```

```
## [1] "dgCMatrix"
## attr(,"package")
## [1] "Matrix"
```

```
# Formato 'matrix array'
(Y <- as_adjacency_matrix(g, sparse = F))
```

```
##   1 2 3 4 5 6 7
## 1 0 1 1 0 0 0 0
## 2 1 0 1 1 0 0 0
## 3 1 1 0 0 1 0 0
## 4 0 1 0 0 1 1 1
## 5 0 0 1 1 0 1 0
## 6 0 0 0 1 1 0 1
## 7 0 0 0 1 0 1 0
```

```
# Clase de objeto
class(Y)
```

```
## [1] "matrix" "array"
```

```
# Simétrica?
isSymmetric(Y)
```

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

```
# Versión vectorizada exhaustiva
(yvec1 <- Y[lower.tri(Y)])
```

```
##  [1] 1 1 0 0 0 0 1 1 0 0 0 0 1 0 0 1 1 1 1 0 1
```

```r
# Versión vectorizada indexada
(yvec2 <- which(yvec1 == 1))
```

```
##  [1]  1  2  7  8 13 16 17 18 19 21
```

2.3.3 Matriz de aristas

Una **matriz de aristas** es un **arreglo de dos columnas** conformado por todos los **pares de vértices** que están **unidos** por una arista.

En el caso de redes ponderadas, la matriz de aristas incluye una columna adicional que registra el peso asociado a cada arista.

2.3.4 Ejemplo: red binaria no dirigida (cont.)

```r
# Matriz de aristas
n <- dim(Y)[1]
A <- NULL
for (i in 1:(n-1))
  for (j in (i+1):n)
    if (Y[i,j] == 1)
      A <- rbind(A, c(i,j))
```

```r
# Clase de objeto
class(A)
```

```
## [1] "matrix" "array"
```

```r
print(A)
```

```
##       [,1] [,2]
##  [1,]    1    2
##  [2,]    1    3
##  [3,]    2    3
##  [4,]    2    4
##  [5,]    3    5
##  [6,]    4    5
##  [7,]    4    6
##  [8,]    4    7
##  [9,]    5    6
## [10,]    6    7
```

2.4 Ejemplo: trabajo colaborativo

Red de **relaciones de trabajo colaborativo** entre los miembros de una firma de abogados (SG&R). Estos datos fueron recolectados para **estudiar la cooperación** entre los actores de una organización.

$y_{i,j} = 1$ si los miembros i y j **trabajaron juntos** en al menos un caso y $y_{i,j} = 0$ en otro caso.

Una descripción completa de los datos se puede encontrar aquí.

Disponible en el paquete `sand` de R.

Lazega, E. (2001). *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership.* Oxford University Press on Demand.

```
suppressMessages(suppressWarnings(library(sand)))
suppressMessages(suppressWarnings(library(corrplot)))
```

```
# Aristas
head(elist.lazega)
```

```
##   V1  V2
## 1 V1 V17
## 2 V2  V7
## 3 V2 V16
## 4 V2 V17
## 5 V2 V22
## 6 V2 V26
```

```
# Dimensión
dim(elist.lazega)
```

```
## [1] 115   2
```

```
# Atributos
head(v.attr.lazega)
```

```
##   Name Seniority Status Gender Office Years Age Practice School
## 1   V1         1      1      1      1    31  64        1      1
## 2   V2         2      1      1      1    32  62        2      1
## 3   V3         3      1      1      2    13  67        1      1
## 4   V4         4      1      1      1    31  59        2      3
## 5   V5         5      1      1      2    31  59        1      2
## 6   V6         6      1      1      2    29  55        1      1
```

```
# Dimensión
dim(v.attr.lazega)
```

```
## [1] 36  9
```

```
# Grafo
# Ver también 'graph_from_edgelist' y 'graph_from_adjacency_matrix'
g <- graph_from_data_frame(d = elist.lazega, directed = "F", vertices = v.attr.lazega)

V(g)$name <- 1:vcount(g)
```

```
# Matriz de adyacencia
Y <- as_adjacency_matrix(g, sparse = F)
```

```
# Orden
vcount(g)
```

```
## [1] 36
```

```
# Tamaño
ecount(g)
```

```
## [1] 115
```

```
# Dirigida?
is_directed(g)
```

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

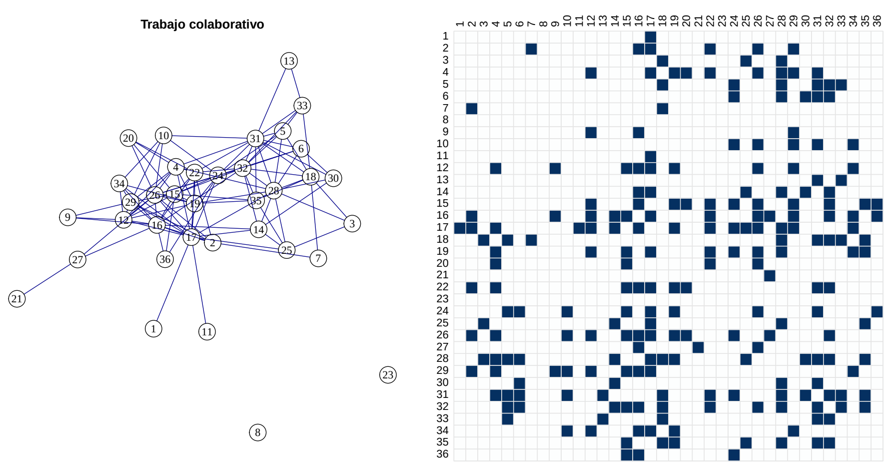
```
# Ponderada?
is_weighted(g)
```

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

```
# Simple?
is_simple(g)
```

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

```r
# Visualización
par(mfrow = c(1,2), mar = c(1, 1, 2, 1))

# Grafo
set.seed(123)
plot(
  g,
  vertex.size        = 9,
  vertex.label.color = "black",
  vertex.color       = NA,
  vertex.frame.color = "black",
  edge.color         = "blue4",
  main               = "Trabajo colaborativo"
)

# Matriz de adyacencia
corrplot(
  corr        = as.matrix(Y),
  method      = "color",
  tl.col      = "black",
  addgrid.col = "gray90",
  cl.pos      = "n"
)
```



2.5 Ejemplo: relaciones sociales

Datos de empleados de *Western Electric* (*Hawthorne plant*) en el banco de conexiones. El estudio buscaba entender cómo las **relaciones sociales informales** entre los trabajadores influían en su comportamiento y productividad.

Los empleados trabajaban en una sola sala e incluían **dos inspectores** (actores 1 y 2), **tres soldadores** (actores 12, 13 y 14) y **nueve técnicos de cables** (actores 3 a 11).

Se recopilaron datos sobre cuatro categorías de interacción simétrica, que incluyen: participación en juegos pesados (*Horseplay*, red 1), participación en discusiones (*Arguments*, red 2), amistad (*Friendship*, red 3) y comportamiento antagónico (*Antagonism*, red 4).

Estos datos se organizan como una **red multicapa** formada por **redes binarias no dirigidas**, donde cada **capa** k representa las conexiones entre $n = 14$ individuos mediante la **matriz de adyacencia** $\mathbf{Y}_k = [y_{i,j,k}]$. Los elementos de esta matriz son $y_{i,j,k} \in \{0, 1\}$, para $1 \leq i < j \leq n$ y $k = 1, \ldots, K$.

Roethlisberger, F. J., & Dickson, W. J. (2003). *Management and the Worker: An Account of a Research Program Conducted by the Western Electric Company, Hawthorne Works, Chicago.* Psychology Press.

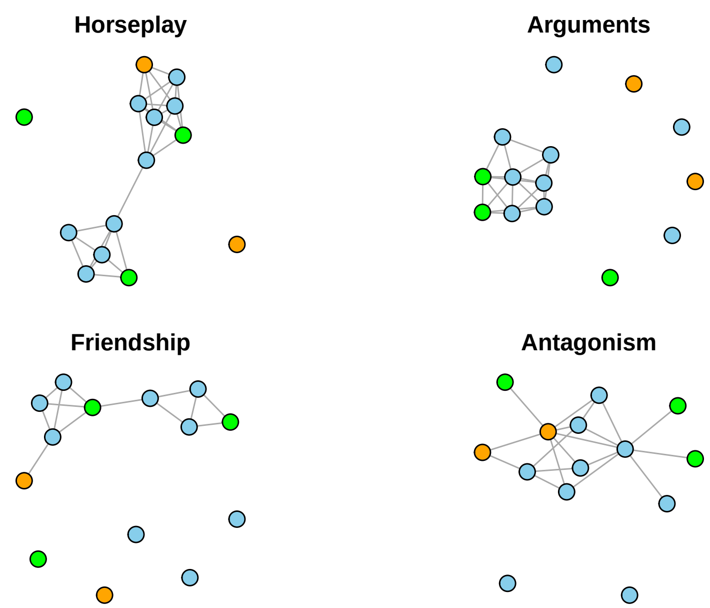
```
# Datos
load("wiring_data.RData")

# Dimensiones
dim(Y)
```

```
## [1] 14 14  4
```

```
n <- dim(Y)[1]
K <- dim(Y)[3]
```

```
# Definir los colores para los grupos de actores
node_colors <- c(rep("orange", 2), rep("skyblue", 9), rep("green", 3))

# Títulos personalizados para cada capa
titles <- c("Horseplay", "Arguments", "Friendship", "Antagonism")

# Configurar el layout del gráfico
par(mfrow = c(2, 2), mar = c(1, 1, 2, 1))

# Dibujar los gráficos para cada capa
for (k in 1:4) {
```

```
  # Crear el grafo desde la matriz de adyacencia
  g_k <- graph_from_adjacency_matrix(Y[, , k], mode = "undirected")

  # Asignar etiquetas de nodos numeradas de 1 a n
  V(g_k)$label <- 1:vcount(g_k)

  # Graficar
  set.seed(123)
  plot(
    g_k,
    main = titles[k],
    vertex.label = NA,
    vertex.color = node_colors
  )
}
```



La **red de consenso binaria** se define como la red cuya matriz de adyacencia $\mathbf{Y} = [y_{i,j}]$ tiene entradas $y_{i,j} = 1$ si el promedio $\frac{1}{K}\sum_{k=1}^{K} y_{i,j,k}$ es mayor que 0.5 (i.e., si el enlace aparece en más de la mitad de las capas), y $y_{i,j} = 0$ en caso contrario.

La **red de consenso ponderada** se define como la red cuya matriz de adyacencia $\mathbf{Y} = [y_{i,j}]$ tiene entradas $y_{i,j}$ dadas por el total $\sum_{k=1}^{K} y_{i,j,k}$, el promedio $\frac{1}{K}\sum_{k=1}^{K} y_{i,j,k}$ o alguna otra **medida de resumen** adecuada de los enlaces a través de las K capas.

```r
# Matriz de consenso ponderada (total)
Y_c <- apply(X = Y, MARGIN = c(1, 2), FUN = sum)

# Dimensiones
dim(Y_c)
```

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

```r
# Grafo de consenso ponderado
g_c <- graph_from_adjacency_matrix(Y_c, mode = "undirected", weighted = TRUE)

# Orden
vcount(g_c)
```

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

```r
# Tamaño
ecount(g_c)
```

```
## [1] 51
```

```r
# Dirigida?
is_directed(g_c)
```

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

```r
# Ponderada?
is_weighted(g_c)
```

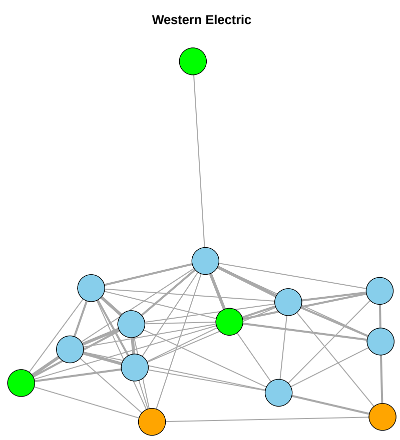
```
## [1] TRUE
```

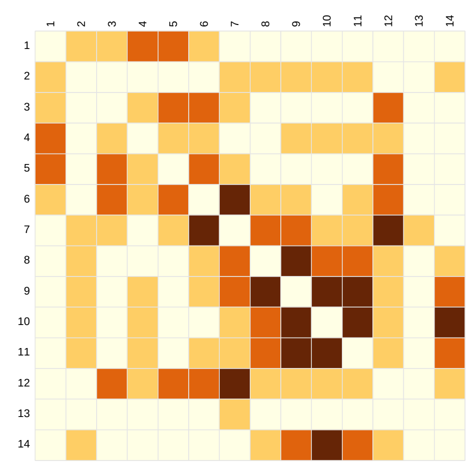
```r
# Simple?
is_simple(g_c)
```

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

```r
# Visualización
par(mfrow = c(1, 2), mar = c(1, 1, 2, 1))

# Grafo
set.seed(123)
```

```r
plot(
  g_c,
  vertex.label       = NA,
  vertex.color       = node_colors,
  vertex.frame.color = "black",
  edge.width         = 1.5 * E(g_c)$weight,
  main               = "Western Electric"
)

# Matriz de adyacencia de consenso
corrplot(
  corr         = Y_c,
  is.corr      = FALSE,
  method       = "color",
  tl.col       = "black",
  addgrid.col  = "gray90",
  cl.pos       = "n",
  main         = ""
)
```





2.6 Ejemplo: redes terroristas y contra-terroristas

Basado en la **red terrorista de Noordin Top**, donde se analizan 79 individuos y sus vínculos con 32 **organizaciones terroristas**, instituciones educativas, negocios y vínculos personales (familia, amistad, correligión, apoyo logístico y participación en operaciones).

El objetivo es diseñar una **estrategia estructural** para desarticular o debilitar de forma eficiente la red terrorista.

Estos datos de **pertenencia** a organizaciones terroristas se almacenan en una **matriz de adyacencia bipartita** $\mathbf{Y} = [y_{i,j}]$, de dimensión $n \times m$, donde $y_{i,j} = 1$ si el **individuo** i pertenece a la **organización terrorista** j y $y_{i,j} = 0$ en caso contrario.

Datos disponibles en https://thearda.com/data-archive?fid=TERRNET

Everton, S. F. (2012). *Network Topography, Key Players and Terrorist Networks*. Connections 31(1):12-19.

```
# Importar datos
noordin <- read.delim("~/Dropbox/UN/netwroks_lectures/noordin.txt")

# Identificar las columnas que comienzan con "ORGAN"
organ_cols <- grep(pattern = "^ORGAN", x = names(noordin), value = TRUE)

# Matriz de adyacencia bipartita
Y <- as.matrix(noordin[, organ_cols])
```

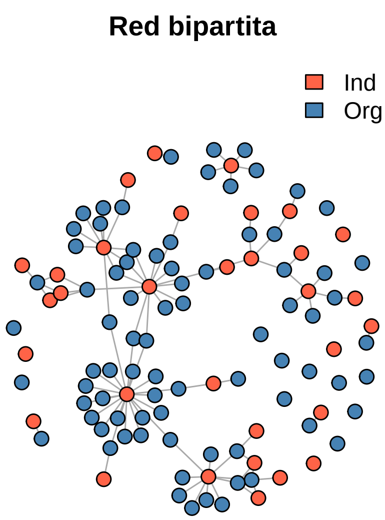
```
# Grafo bipartito a partir de la matriz de adyacencia
# filas = tipo 0 (individuos), columnas = tipo 1 (organizaciones)
g <- graph_from_biadjacency_matrix(Y)

# Tipo de nodos
table(V(g)$type)
```

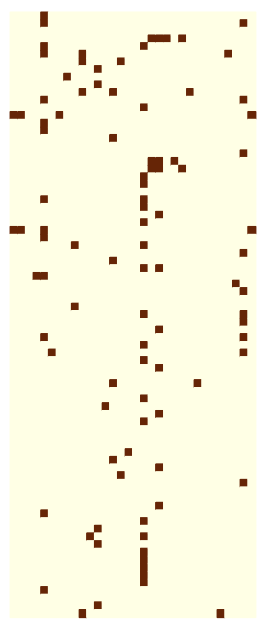
```
##
## FALSE  TRUE
##    79    32
```

```
# Colores de acuerdo el tipo de nodo
node_colors <- ifelse(V(g)$type, "tomato", "steelblue")
```

```
# Visualización
par(mfrow = c(1, 2), mar = c(1, 1, 2, 1))

# Grafo
set.seed(123)
plot(
  g,
  vertex.label = NA,
  vertex.color = node_colors,
  vertex.size  = 8,
```

```
  main          = "Red bipartita"
)
legend("topright", legend = c("Ind", "Org"), fill = c("tomato", "steelblue"), bty = "n

# Matriz de adyacencia bipartita
corrplot(
  corr    = Y,
  is.corr = FALSE,
  method  = "color",
  tl.pos  = "n",
  cl.pos  = "n"
)
```





A partir de la matriz de adyacencia de una **red bipartita** $\mathbf{Y}$ de tamaño $n \times m$ se pueden construir dos **redes ponderadas** mediante proyecciones sobre cada conjunto de nodos:

- **Red proyectada de las filas**: $\mathbf{Y}_{\mathrm{F}} = \mathbf{Y}\,\mathbf{Y}^{\top}$, de tamaño $n \times n$.

- **Red proyectada de las columnas**: $\mathbf{Y}_{\mathrm{C}} = \mathbf{Y}^{\top}\mathbf{Y}$, de tamaño $m \times m$.

En ambos casos, los pesos se pueden interpretar como una **medida de similitud** entre nodos del mismo tipo, inducida por sus patrones de conexión en la red bipartita original:

- En la proyección sobre filas, el peso en la arista (i, i') es el **número de organizaciones compartidas** por los individuos i e i'.
- En la proyección sobre columnas, el peso en la arista (j, j') es el **número de individuos compartidos** por las organizaciones j y j'.

```
# Proyección sobre filas
Y_F <- Y %*% t(Y)
diag(Y_F) <- 0

# Proyección sobre columnas
Y_C <- t(Y) %*% Y
diag(Y_C) <- 0

# Grafos ponderados a partir de las proyecciones
g_F <- graph_from_adjacency_matrix(
  Y_F,
  mode     = "undirected",
  weighted = TRUE,
  diag     = FALSE
)

g_C <- graph_from_adjacency_matrix(
  Y_C,
  mode     = "undirected",
  weighted = TRUE,
  diag     = FALSE
)
```

```
# Visualización
par(mfrow = c(1, 2), mar = c(1, 1, 2, 1))

# Grafo proyección filas
set.seed(123)
plot(
  g_F,
  main          = "Individuos",
  vertex.size  = 4,
  vertex.color = "tomato",
  vertex.label = NA,
  edge.width   = 0.5 + E(g_F)$weight
)

# Grafo proyección columnas
set.seed(123)
plot(
  g_C,
```

```
  main          = "Organizaciones",
  vertex.size   = 8,
  vertex.color  = "steelblue",
  vertex.label  = NA,
  edge.width    = 0.5 + 1.5 * E(g_C)$weight
)
```