Departamento de Economía

Facultad de Ciencias
Administrativas y Económicas

Economics Lecture Notes

Breve tutorial para visualizar y calcular métricas de Redes (grafos) en R (para Economistas)

Julio César Alonso C
Jaime Andres Carabali M

Icesi ECONOMICS LN No. 7

Abril 2019

Breve tutorial para visualizar y calcular métricas de Redes (grafos) en R (para Economistas)

Julio César Alonso C
Jaime Andres Carabali M

Icesi
ECONOMICS LN No. 7
Abril 2019

Universidad Icesi

Editor:
Carlos Giovanni González Espitia
Profesor tiempo completo, Universidad Icesi
cggonzalez@icesi.edu.co

Asistente editorial:
Diana Marcela Zambrano Lucumí

Gestión Editorial
Departamento de Economía - Universidad Icesi

Apuntes de Economía es una publicación del Departamento de Economía de la Universidad Icesi, cuya finalidad es divulgar las notas de clase de los docentes y brindar material didáctico para la instrucción en el área económica a diferentes niveles. El contenido de esta publicación es responsabilidad absoluta de los autores.

www.icesi.edu.co
Tel: 5552334. Fax: 5551441
Calle 18 # 122-135 Cali, Valle del Cauca, Colombia

Breve tutorial para visualizar y calcular métricas de Redes (grafos) en R (para Economistas).

Julio César Alonso C.*
Jaime Andres Carabali M. **
Departamento de Economía - Universidad Icesi
Cali - Colombia

8 de abril de 2019

Objetivos de aprendizaje

Al finalizar la lectura de este documento se espera que el lector esté en capacidad de:

- Explicar con sus propias palabras el concepto de grafo y sus posibles aplicaciones.
- Explicar en sus propias palabras el significado de las métricas básicas de un grafo.
- Emplear el paquete *igraph* de R para graficar y describir las principales propiedades de un grafo.
- Escoger el tipo de grafo más apropiado según la información disponible y los objetivos del estudio.
- Identificar comunidades en un grafo.

1. Introducción

Hasta hace relativamente poco tiempo, solamente un pequeño grupo de economistas empleaba la teoría de las redes sociales como herramientas para abordar fenómenos económicos, para los cuales, la teoría tradicional no ofrecía una buena explicación. Esto ha cambiado de manera progresiva, sobre todo, después de algunos trabajos en el área de sociología, y posteriormente en el área de la economía, que tuvieron un gran impacto. Por ejemplo, Granovetter (1985) presenta una reflexión teórica sobre la actividad económica y la estructura social, y muestra que algunos patrones observados en el mercado, aparentemente inexplicables, pueden ser fácilmente caracterizados y explicados a través de la teoría de las redes sociales. Desde Granovetter (1985) y otros documentos influyentes (como por ejemplo: Granovetter (1978), Hendricks, Piccione y Tan (1999) y Travers y Milgram (1977)), la teoría de grafos se ha aplicado a problemas económicos muy diversos, desde la búsqueda de empleo (Calvo-Armengol y Jackson, 2004), la teoría de juegos (Avrachenkov, Kondratev y Mazalov, 2017) y el comercio internacional (De Benedictis, Nenci, Santoni, Tajoli y Vicarelli, 2014) hasta las revoluciones científicas (Salazar y Otero, 2015).

El análisis de redes sociales, a diferencia del *mainstream* de la teoría económica, permite evaluar de manera simultánea las relaciones dentro de un grupo y las cualidades de cada miembro, permitiendo al

*Director del centro de investigación en economía y finanzas (Cienfi) y profesor titular del departamento de Economía de la Universidad Icesi
**Asistente de investigación Cienfi

investigador entender detalles de la interdependencia y las conexiones informales y formales dentro de una sociedad. Este documento no pretende ser una introducción a la teoría de redes sociales (o también conocida como teoría de grafos), pero sí una breve mirada a un conjunto de instrumentos de ésta que puede ser útil tener en la caja de herramientas de un economista. En este documento se presenta una aplicación de la teoría de grafos, para analizar el desempeño individual y grupal de los seleccionados de Francia y Croacia en el partido final de la Copa Mundial de fútbol Rusia 2018. Ejemplo que podrá desarrollar el lector empleando el archivo anexo.

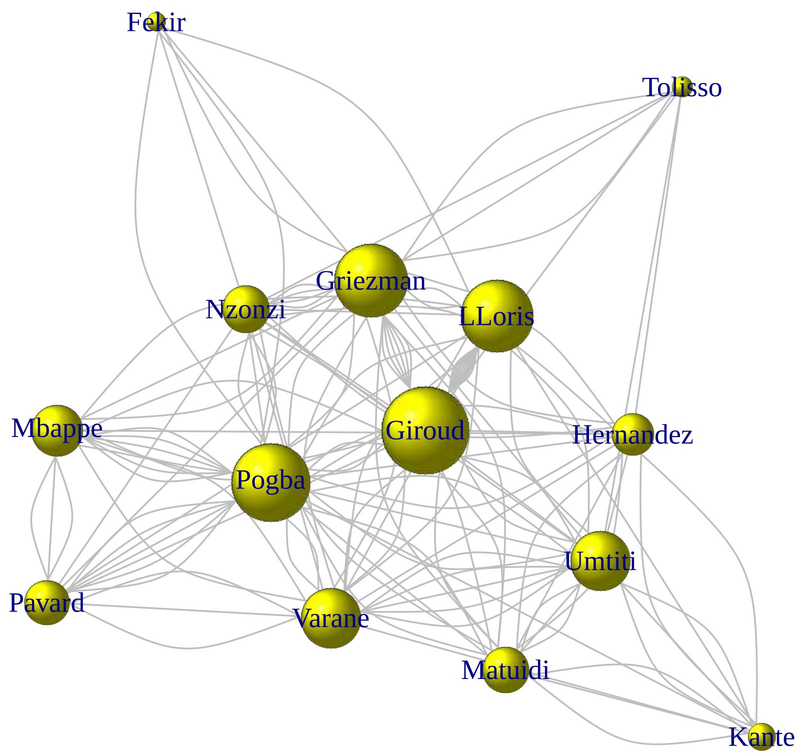
Los partidos de fútbol ofrecen un excelente objeto de estudio a ser abordado mediante la teoría de grafos por al menos, dos factores. Primero, debido a la naturaleza de este deporte es importante que los integrantes de un equipo colaboren y armen una estructura colectiva compacta y sólidamente interconectada, para superar al contrincante. Una adecuada explotación de las sinergias que pueden surgir entre los miembros podría permitir que el equipo alcance un desempeño mayor a la sumatoria de los desempeños individuales, dado que, en este tipo de competencias la sincronización e interacción es clave. Segundo, tal cooperación puede ser vista como relaciones sociales que surgen en un sistema micro-social (Lusher, Robins y Kremer, 2010). Es más, se han identificado dos hechos estilizados en los partidos de fútbol. Primero, las aplicaciones del análisis de grafos encuentran una fuerte correlación positiva entre las densidades de las redes y el rendimiento obtenido de un equipo (una medida de la cantidad de interconexiones en la red). Y segundo, las redes fuertemente dependientes de pocos miembros del equipo (redes altamente centralizadas) están asociadas con un bajo rendimiento (Lusher, Robins y Kremer, 2010). Estos dos conceptos, densidad y centralidad, se discutirán en las siguientes secciones. Pero en palabras más simples, estos dos hechos estilizados que se ratifican con esta herramienta no son más que dos principios que cualquier seguidor del fútbol conoce. Primero que un once que juega en equipo tiene mejor rendimiento que uno en el que los jugadores no se acoplan. Y dos, si en un equipo solo se crean sociedades entre un conjunto pequeño de ellos donde se concentra el juego, el once no tendrá un buen rendimiento.

No obstante, los partidos de fútbol no son más que uno entre muchos de los fenómenos que pueden ser abordados mediante la teoría de grafos. Por ejemplo, la teoría de grafos ha enriquecido mucho el estudio sobre el emparejamiento entre desempleados y empleadores. Dicho emparejamiento es más probable si ambos agentes tienen redes sociales amplias, densas e interconectadas. Esto, debido a que, mayor tamaño, densidad y conexión de las redes facilita el flujo de la información y aumenta el número de rutas posibles mediante las cuales un desempleado y un empleador con vacantes libres se pueden encontrar. En este caso, el grafo de las redes sociales de ambos agentes puede servir de base, por ejemplo, para predecir la frecuencia de emparejamientos, predecir el desempleo, entre otras métricas de interés (Calvo-Armengol y Jackson, 2004).

Este documento se suma al conjunto de esfuerzos enfocados en difundir el análisis de redes (también conocido como análisis de grafos) en la disciplina económica. Aunque se está avanzando en el proceso de inclusión de este tipo de métodos a la caja de herramientas tradicionales de los economistas, tal avance es incipiente. Tal es la falta de progreso en esta área que, aunque estas herramientas se vienen utilizando en otras ciencias hace bastante tiempo ya, en nuestra disciplina es novedosa. Si se tiene en cuenta lo importante que es contar con un enfoque que permita abordar el problema de las interdependencias inherentes a las estructuras de un conjunto de agentes, entonces se justifica el esfuerzo. Aquí se presentan y explican las redes y algunos de los conceptos asociados a estas, empleando el software estadístico R (R Core Team, 2018) y el paquete *igraph* (Csardi y Nepusz, 2006).

Antes de continuar veamos un ejemplo. La figura 1 presenta un grafo no dirigido asociado a los pases del equipo de Francia durante la final de la Copa Mundial de fútbol Rusia 2018. Los jugadores constituyen los nodos o vértices (círculos) y están conectados por las aristas (pases). El grafo se denomina no dirigido porque las aristas no indican cual es el origen y el punto de llegada del pase. Como veremos, se pueden emplear métricas que resuman parte de la interrelación entre los individuos (las aristas o pases). Por ahora, debe ser claro de esta visualización que los jugadores con más aristas se representan con nodos más grandes, mostrando que éstos son más importantes para el equipo; como por ejemplo, Giroud, Griezman, Lloris y Pogba. Adicionalmente, podemos ver cómo jugadores como Tolisso y Fekir están en la situación opuesta. Es decir, el grafo permite resumir el comportamiento de un equipo durante un partido. Al final de este documento, usted estará en capacidad de construir este tipo de visualizaciones y calcular e interpretar métricas asociadas al grafo.

Figura 1: Grafo de los pases durante la final de Rusia 2018 del seleccionado francés



Fuente: Elaboración propia con datos de la FIFA

Lo que resta de este documento está organizado de la siguiente manera. En la segunda sección se muestran algunos ejemplos para introducir la notación y explicar la forma de los grafos. En la tercera sección se presenta y describe nuestra fuente de datos. En la sección cuatro se definen algunas estadísticas resumen del grafo. En las secciones cinco y seis se describen las aplicaciones en R, en lo referente a las representaciones de un grafo y las estadísticas descriptivas asociadas a este. En la sección siete se describe el proceso para identificar comunidades al interior de un grafo. En la octava sección se presentan unas reflexiones finales.

2. Un ejemplo sencillo

Figura 2: Ejemplo 1 de matriz de adyacencia hipotética

Ejemplo 1

	A	B	C	D	E
A	0	1	0	2	0
B	0	0	0	1	0
C	1	1	0	0	1
D	0	1	1	0	0
E	0	0	2	0	0

Fuente: Elaboración propia

La figura 2 muestra una matriz hipotética, que se conoce con el nombre de matriz de adyacencia, con su respectivo grafo. En esta matriz se representa las interacciones entre los miembros de la red. Cada columna y fila identifica a un miembro de la red. En este ejemplo hemos empleado letras para identificar a cada jugador. También es importante mencionar que el nombre de cada fila en la matriz indica el punto de partida de la conexión y el nombre de cada columna el punto de llegada. En el caso de un partido de fútbol esto nos diría quién dio el pase (el nombre de la columna) y quién lo recibió (el nombre de la fila). En la matriz de adyacencia del ejemplo 1, se puede observar que el jugador A le hace un pase al B y dos al D. Y el jugador A recibe un pase del jugador C. El ejemplo contiene dos representaciones gráficas (grafos) de la matriz. Un grafo no dirigido y uno dirigido. Se dice que un grafo es *no dirigido* cuando las aristas (las líneas que conectan los jugadores) no indican si la conexión entre un par de jugadores parte del primero o del segundo. En caso contrario el grafo es dirigido. En resumen, la matriz de adyacencia tiene como objetivo informar sobre las interrelaciones entre jugadores, y el grafo corresponde a la contraparte gráfica de dichas interrelaciones.

Figura 3: Ejemplo 2 de matriz hipotética

Ejemplo 2

	A	B	C	D	E
A	0	20	2	3	4
B	20	0	1	1	3
C	1	1	0	39	2
D	1	1	30	0	1
E	2	7	3	2	0

Fuente: Elaboración propia

Ahora consideremos el ejemplo 2 (ver figura 3). La matriz de adyacencia del ejemplo 2 sugiere que A y B se relacionan de manera recurrente (20 conexiones que parten de A y otras 20 que parten de B), al igual que C y D (39 conexiones que parten de C y otras 30 que parten de D). Mientras que las demás conexiones directas entre pares distintos de jugadores, en términos relativos, son escasas. El ejemplo muestra dos grafos no direccionados: uno no ponderado y otro ponderado. Un grafo es *ponderado* si, independientemente del número de enlaces que existen entre un par jugadores, estos están conectados por medio de una sola línea y dicha línea tiene asociada una etiqueta o algo que de alguna manera revele el número de conexiones que unen ese par de jugadores. En el grafo ponderado del ejemplo 2, el grosor o nivel de solidez de cada línea refleja la recurrencia con la cual se conectan un par de jugadores. Como se puede observar, las líneas más gruesas son aquellas que conectan a A con B y a D con C.

3. Los datos para nuestro ejemplo

Para llevar a cabo este ejercicio se emplea una base de datos que corresponde a los pases de las selecciones de Francia y Croacia en la final del Mundial de Rusia de 2018. La base de datos fue extraída de la página de la FIFA. Los cuadros 1 y 2 muestran la distribución de pases entre los jugadores de las selecciones de Croacia y de Francia, respectivamente. Estas son las correspondientes matrices de adyacencia. Las filas indican el origen del pase y las columnas el punto de llegada. Por ejemplo, en la distribución de pases de Croacia, el valor 7 en fila 1 y la columna 5 indica que el jugador Lovren recibió 7 pases exitosos del jugador Subasic. Análogamente, la casilla en la fila 5 y columna 1 indica que el jugador Subasic recibió tres pases de Lovren. Las demás casillas se interpretan de manera similar. Esta base de datos se encuentra en el archivo "Partidos.xlsx" que acompaña a este documento.

Cuadro 1: Distribución de pases (matriz de adyacencia) en la final de Rusia 2018 - Seleccionado de Croacia

	Subasic	Vrsaljko	Strinic	Perisic	Lovren	Rakitic	Modric	Brozovic	Mandzukic	Rebic	Vida	Kramaric	Pjaca
Subasic	0	1	0	0	7	1	1	1	1	0	3	0	0
Vrsaljko	1	0	0	5	15	4	10	10	5	2	0	0	1
Strinic	0	0	0	6	0	5	0	4	1	1	1	0	0
Perisic	0	2	3	0	0	3	8	0	1	1	2	0	0
Lovren	3	10	0	1	0	8	9	17	3	0	7	0	0
Rakitic	1	2	6	5	5	0	6	9	2	7	5	1	0
Modric	0	19	3	4	8	3	0	16	8	4	3	2	1
Brozovic	0	15	6	5	7	16	22	0	1	2	9	4	0
Mandzukic	0	4	0	4	0	2	3	6	0	4	0	0	0
Rebic	0	2	1	2	0	0	0	1	0	0	1	0	0
Vida	2	1	2	2	8	11	4	9	1	0	0	2	0
Kramaric	0	1	2	2	0	3	0	1	0	0	1	0	1
Pjaca	0	1	0	0	0	0	0	1	0	0	0	0	0

Fuente: Elaboración propia

Cuadro 2: Distribución de pases (matriz de adyacencia) en la final de Rusia 2018 - Seleccionado de Francia

	LLoris	Pavard	Varane	Umtiti	Pogba	Griezman	Giroud	Mbappe	Kante	Matuidi	Hernandez	Tolisso	Nzonzi	Fekir
LLoris	0	1	1	1	2	2	10	0	1	0	0	0	1	2
Pavard	1	0	3	0	2	0	1	3	0	0	0	0	1	0
Varane	3	3	0	5	2	2	2	2	1	0	1	0	1	0
Umtiti	2	0	5	0	3	1	3	0	3	1	0	1	1	0
Pogba	0	5	0	2	0	2	2	6	0	2	3	0	3	2
Griezman	0	1	2	0	2	0	1	3	0	2	2	3	3	1
Giroud	0	0	2	0	2	6	0	2	0	1	1	0	1	0
Mbappe	0	3	2	0	3	1	0	0	0	0	0	0	0	0
Kante	0	0	0	2	1	0	0	0	0	3	1	0	0	0
Matuidi	0	0	0	3	3	2	2	0	3	0	3	0	2	0
Hernandez	0	0	0	1	1	1	1	1	2	2	0	1	1	0
Tolisso	0	0	0	0	0	2	1	0	0	0	1	0	1	0
Nzonzi	0	0	1	1	3	4	0	0	0	2	2	0	0	1
Fekir	0	0	0	0	0	1	0	0	0	0	0	0	1	0

Fuente: Elaboración propia

Antes de mostrar cuál es el proceso para construir el grafo mediante el programa estadístico R (R Core Team, 2018) y el paquete *igraph* (Csardi y Nepusz, 2006), se describen en la siguiente sección algunos conceptos y métricas importantes. Estas métricas permiten interpretar o extraer información relevante de un grafo.

4. Un poco de definiciones y conceptos

En esta sección se introduce la notación a utilizar y se definen algunos conceptos. Empecemos por los componentes de un grafo: los nodos y las aristas. Los nodos siempre representan a los agentes del fenómeno que se esté analizando y las aristas simbolizan las conexiones entre estos. En nuestro caso, como se puede observar en la figura 1, los nodos del grafo representan a los jugadores del equipo francés y aquellas líneas que los conectan (las aristas) representan a los pases. En la notación a utilizar, los nodos son representados por la letras i, j y v; mientras que el conjunto de nodos será representado por la letra mayúscula V, y el número de aristas que partan del nodo i y lleguen al nodo j será representado por *A(i,j)*. Por ejemplo, para la matriz de adyacencia de Francia tenemos que $V = \{$Lloris, Pavard, Varane, ..., Fekir$\}$. Adicionalmente, en la dirección (Lloris, Varane) tenemos una arista (un pase). Es decir, $A(Lloris, Varane) = 1$. En la dirección (Varane, Lloris) tenemos 3 aristas, por lo tanto A(Varane, Lloris) = 3. Ahora, en función de los nodos y las aristas se definen algunas métricas o estadísticas descriptivas, las cuales ayudan a caracterizar los grafos.

4.1. Centralidad

Cuando se analizan redes es común que se desee encontrar los nodos que son más importantes para la red. En términos generales, la importancia de un nodo depende de su posición en la red. A medida que las redes se vuelven más complejas, será necesario emplear medidas de centralidad para describirlas. Así como en la estadística se cuenta con medidas de tendencia central como la mediana y la moda para caracterizar dónde se encuentra el "centro" de los datos, en el caso de las redes se cuenta con medidas de centralidad para determinar los nodos más importantes. Continuando con la analogía con la estadística, es importante tener en cuenta que la centralidad no es la única característica de una red, como no lo es la media para una muestra de datos. La centralidad refleja la dependencia de la red o prominencia de un individuo (o un sub-grupo pequeño con relación al tamaño del equipo) dentro de la red. Un rápido vistazo a la figura 1 revela que 4 jugadores destacan frente a los demás, esto, por el volumen de los nodos. Como se puede observar, los jugadores Griezman, Giroud, Pogba y LLoris parecen ser lo más influyentes de Francia. A continuación se describen las medidas de centralidad más comunes, las cuales se pueden clasificar en dos grupos: locales y globales.

4.1.1. Medidas locales

Las métricas locales reflejan la influencia que tienen los actores en sus compañeros más cercanos y resaltan a los jugadores que pueden conectarse rápidamente con la red más amplia. El adjetivo *local* se debe a que estas medidas solo toman en cuenta la influencia de un jugador (nodo) sobre sus compañeros más cercanos. Estas medidas locales de centralidad son:

- ***Grado de centralidad***: Para obtener esta medida basta con contar el número de enlaces relacionados con cada nodo (Tanto los entrantes como los salientes de cada nodo). Tal y como está definida, esta estadística hará que destaquen aquellos jugadores conectados a las sub-redes más amplias. Un valor más grande de esta medida indicará una mayor centralidad del nodo. En el cuadro 4.2 se presenta esta medida para cada uno de los jugadores del equipo francés. Según esta métrica, los jugadores que más dieron y recibieron pases en la final fueron Pogba, Giroud y

Griezman. Mientras que los jugadores menos activos en el equipo, en términos de pases, fueron Fekir y Tolisso.

Esta métrica se puede presentar también de manera normalizada; es decir, dividiendo el número de enlaces relacionados con cada uno de los n nodos dividido por *n-1*. Otras variantes de esta medida de centralidad solo consideran los enlaces de salida (*out-degree*) de cada nodo o solo los de entrada (*in-degree*). Estas últimas dos métricas pueden estar normalizadas o no. La normalización puede ser importante cuando se quiere comparar esta métrica entre diferentes grafos con números de nodos muy diferentes.

- ***Betweenness***: esta medida cuantifica el número de veces que un nodo actúa como un puente a lo largo de la ruta geodésica (esto es, la de menor longitud) entre dos nodos. Su cálculo implica los siguientes pasos: (i) contar el número de rutas que unen un par de nodos (i y j), (ii) encontrar la proporción de esas rutas que pasan por un tercer nodo (v), (iii) repetir el cálculo para todas las parejas de nodos posibles (con la condición de que i y j sean diferentes de v) hallando en cada caso la proporción de rutas que pasan por el nodo v y, por último, sumar todas las proporciones obtenidas. Tomando σ_{ij} como el número de rutas de mínima distancia que unen a los nodos i y j, y $\sigma_{ij}(v)$ como el número de rutas de distancia mínima que unen a estos nodos y que pasan por el nodo v, entonces la métrica de *Betweenness* para el nodo v está dada por:

$$Betweenness(v) = \sum_{i \neq v \varepsilon V} \sum_{j \neq v \varepsilon V} \delta_{ij}$$

  donde $\delta_{ij}(v) = \frac{\sigma_{ij}(v)}{\sigma_{ij}}$.

  Esta métrica solo toma valores entre cero y el número de parejas posibles dentro del grafo (excluyendo el nodo para el cual se está efectuando el cálculo). Esto es, si N es el número total de nodos, entonces la cota superior toma el valor de $\frac{(N-1)!}{2!((N-1)-2)!}$. Por otra parte, tal y como se ha mencionado de manera implícita, entre mayor sea el *Betweenness* mayor será la centralidad del nodo. En el cuadro 4.2 se presenta el valor de esta estadística para cada jugador del equipo francés en la final de Rusia 2018. Según el *Betweenness*, los jugadores más propensos a actuar como puentes entre sus compañeros son Pogba, Griezman y Giroud, precisamente los que más dan y reciben pases. También se cumple que Fekir y Tolisso, los que menos pases dan y reciben, son los que tienen el menor *Betweenness*.

  Este estadístico también se puede presentar de manera normalizada multiplicando cada *Betweenness(v)* por la siguiente constante:

$$\frac{2}{n\,(n-3)\,(n+2)}$$

  .

4.1.2. Medidas globales

En otra dirección, las medidas globales toman en cuenta la totalidad de la red. De manera general, estas medidas identifican a aquellos agentes que están mejor ubicados para influir en toda la red lo más rápidamente posible. Ejemplos, de esta medida globales de centralidad son:

- ***Closeness***: calcula las rutas más cortas entre todos los nodos y asigna una puntuación a cada uno en función de la suma de sus rutas. Mide cuántos pasos se requieren para conectarse a cada uno de los nodos desde un nodo determinado. En general, es una medida de cuánto tarda en llegar la información y en este caso en particular, es una medida de cuánto tarda en llegar el balón a un jugador determinado. Esta medida se calcula de la siguiente manera:

$$Closeness(v) = \frac{1}{\sum_{i \varepsilon V} dist(v,i)}$$

donde, $dist(v,i)$ denota la distancia del nodo v a un nodo i (un poco más abajo se define esta medida). Es fácil deducir que esta medida estará entre cero y uno. Adicionalmente, entre más cercana a 1 esté, mayor será la centralidad del nodo. En el cuadro 4.2 se presenta el cálculo de esta estadística para el grafo de la selección francesa durante el partido final de Rusia 2018. De nuevo, en términos de centralidad (pero ahora global) sobresalen Pogba, Griezman y Giroud, y los peores puntuados son Fekir y Tolisso. Según esta medida, los primeros son los jugadores más cercanos a todo el equipo y los segundos los más lejanos.

Esta métrica se puede presentar también de manera normalizada; es decir, dividiendo el número de enlaces relacionados con cada uno de los n nodos dividido por $n-1$. Así como el grado de centralidad, esta métrica puede calcularse considerando solo las rutas de salida (out) de cada nodo o solo las rutas de entrada (in). Estas últimas dos métricas pueden estar normalizadas o no.

- ***Puntaje de autoridad***: mide la importancia relativa de cada jugador y está definida como el respectivo elemento del vector propio principal de la matriz $A^T A$, donde A es la matriz de adyacencia. En el cuadro 4.2 se muestra este puntaje para cada jugador y, con base en estos se determinó el tamaño de los nodos de la figura 1. El puntaje sugiere que el jugador más importante de Francia en la final de Rusia 2018 fue Giroud.

Cuadro 3: Estadísticas descriptivas por nodo de la red para la final de Rusia 2018 para la selección de Francia

	G. de Central.	Betweenness	Closeness	Excentricidad	Punt. de Auto.
LLoris	24.00	2.73	0.06	2.00	0.83
Pavard	15.00	0.14	0.05	2.00	0.51
Varane	22.00	1.83	0.06	2.00	0.68
Umtiti	23.00	2.64	0.06	2.00	0.68
Pogba	34.00	9.63	0.07	2.00	0.90
Griezman	29.00	5.90	0.07	2.00	0.84
Giroud	31.00	3.19	0.07	2.00	1.00
Mbappe	17.00	0.12	0.05	2.00	0.59
Kante	11.00	0.22	0.05	2.00	0.31
Matuidi	18.00	0.84	0.05	2.00	0.53
Hernandez	17.00	1.87	0.06	2.00	0.48
Tolisso	7.00	0.00	0.05	2.00	0.23
Nzonzi	18.00	1.91	0.07	2.00	0.54
Fekir	6.00	0.00	0.04	2.00	0.21

Fuente: Elaboración propia con datos de la FIFA
Notas: G. de Central. y Punt. de Auto. representan el Grado de Centralidad y el Puntaje de Autoridad, respectivamente.

4.2. Otras medidas para detectar nodos importantes

- ***Distancia***: mide la cantidad de aristas en la ruta más corta que conecta un par de nodos. Si un camino que une a los nodos i y j pasa por los nodos v_1, v_2,... y v_k, entonces, su longitud estará dada por:

$$Longitud(i,j) = \sum_{h=1}^{k} (1) = k$$

La distancia entre i y j estará dada por el camino de menor longitud que los una, es decir:

$$dist(i,j) = min[Longitud(i,j)]$$

Para un grafo no ponderado, se supone que cada arista tiene una longitud igual a 1. Si el grafo es ponderado, se deberá cambiar el 1 en la fórmula por la distancia de cada arista en la ruta que une a i y j, la cual estará en función de su ponderación. Esta medida toma el valor de cero cuando se trata de la distancia de un nodo a el mismo, toma valores positivos cuando se trata de nodos diferentes y está acotada superiormente por el diámetro. En un grafo, hay tantas distancias como pares posibles de nodos, así que conviene tomar el promedio para toda la red (en nuestro caso el equipo). Dado que esta métrica se calcula para cada una de las posibles parejas de nodos, es común que se presente el promedio de la distancia y no cada una de las distancias. El cuadro presenta la distancia promedio. Dada la cercanía a la unidad de este promedio, refleja que en promedio basta con un pase para que un par de jugadores se conecten.

- ***Excentricidad***: se define como la distancia de un nodo particular (el nodo i) al nodo más lejano (el más lejano respecto a i).

  $$Excentricidad(i) = max[dist(i,j)], \quad \forall j$$

  Los valores calculados de esta medida para cada jugador de la selección de Francia en la final de la copa mundo se encuentran reportados en el cuadro . Los cálculos revelan que ésta no varía entre jugadores, pues, en todos los casos bastan dos pases para que un jugador se conecte con su compañero más lejano en la red.

  Así como el grado de centralidad y el *Closeness*, esta métrica puede calcularse considerando solo las rutas de salida (out) de cada nodo o solo las rutas de entrada (in).

4.3. Medidas agregadas para la red

- ***Diámetro***: mide la máxima excentricidad. Es decir,

  $$Dimetro = max[Excentricidad(i)], \quad \forall i$$

  En el cuadro se puede observar que el diámetro del grafo de la selección francesa para el partido de la final es igual a dos. Esto es, el par de jugadores más alejados en toda la red requieren de dos pases para conectarse.

- ***Densidad***: es la proporción de pares de nodos conectados en la red sobre todos los posibles. Si esta medida es 1 (o 100 %), entonces el grafo es totalmente conexo (todos los jugadores se conectan de manera directa con todos). Tomando n como el número total de nodos y E como el número de pares conectados en el grafo, entonces, la densidad se calcula como:

  $$D = \frac{E}{\frac{1}{2}n(n-1)}$$

  donde $\frac{1}{2}n(n-1)$ es el número total de pares de nodos conectados que pueden existir en el grafo. El cálculo de esta medida solo exige el conocimiento de si un par de jugadores se conectaron o no, no se requiere saber el número de conexiones. Así que se debe transformar la matriz de adyacencia para que contenga solo dos valores: ceros (en caso de que no haya conexión) y unos (cuando hay conexión). Para el equipo francés, según el cálculo mostrado en el cuadro , el parámetro de la densidad toma un valor igual a 66 %. Es decir, de todas las posibles conexiones (100 %), se hicieron efectivas el 66 %.

- ***Transitividad***: es una medida de la ocurrencia de pequeñas subredes de tres jugadores totalmente conexas. Esta se calcula de la siguiente manera:

$$T = \frac{\Lambda}{\Delta}$$

donde Λ es el número de triadas transitivas y Δ es el número de potenciales triadas transitivas. En otras palabras, si i está conectado a j, y j está conectado a k, esta mide la probabilidad de que i esté conectado a k. En el cuadro se muestra que la probabilidad de este evento es igual a 73 %, para el caso de la selección de Francia.

- ***Reciprocidad***: mide la probabilidad de que, dado que existe una conexión entre i y j que parte de i, exista una conexión entre estos que parta de j. En otras palabras, si *A(i,j)* es mayor a cero, esta medida es una estimación de la probabilidad de que *A(j,i)* también sea positiva. En el cuadro reporta que esta probabilidad es uno. Es decir, si un jugador de Francia en la final de la copa mundo dio un pase a algún compañero, entonces, siempre ocurrió que, eventualmente, este último le dio un pase al primero.

Cuadro 4: Estadísticas descriptivas agregadas de la red para la final de Rusia 2018 para la selección de Francia

Distancia	Diametro	Densidad	Transitividad	Reciprocidad
1.34	2.00	0.66	0.73	1.00

Fuente: Elaboración propia con datos de la FIFA

5. Aplicación en R

En esta sección ilustremos cómo crear un grafo y cómo calcular las métricas de redes discutidas en la sección anterior. Para tal fin emplearemos R (R Core Team, 2018) y los datos del seleccionado de Croacia que se encuentran en el archivo adjunto *Partidos.xlsx* en la hoja “Croacia". Emplearemos el paquete *igraph* (Csardi y Nepusz, 2006) que permite hacer grafos y calcular una gran variedad de métricas relacionadas a estos. Adicionalmente, emplearemos el paquete *readxl* (Wickham y Bryan, 2018) que permite leer diferentes hojas de un archivo de Excel.

Empecemos por instalar los paquetes y cargar la información. Los paquetes se pueden instalar empleando la función *install.packages* de la siguiente manera:

```
install.packages("igraph")
install.packages("readxl")
```

Ahora carguemos la información de los pases del equipo; es decir, la matriz de adyacencia. En este caso se tienen los datos almacenados en Excel (archivo Partidos.xls) en diferentes hojas. Se puede emplear el paquete *readxl* (Wickham y Bryan, 2018) para cargar los datos deseados. La siguiente línea carga los datos, los convierte en un objeto de la clase *data.frame* y los almacena en el objeto que denominaremos *Croacia*.

```r
Croacia <- as.data.frame(read_excel("Partidos.xlsx",
                                    sheet = "Croacia"))
```

Usted puede constatar que los datos leídos son los mismos que se reportan en el cuadro 1, llamando al objeto **Croacia**.

5.1. Construcción del grafo

Ahora procedamos a la construcción del grafo con los datos de la distribución de pases del equipo croata que se presentaron en el Cuadro 1. Como se mencionó anteriormente, el paquete que emplearemos para generar los grafos es *igraph* (Csardi y Nepusz, 2006). No obstante existen muchos paquetes que permiten construir grafos, este paquete crea grafos de una forma rápida y sencilla mediante la función graph.adjacency. Esta función tiene varios argumentos que permiten personalizar el grafo que se produzca. Esta función tiene la siguiente estructura:

```
   graph_from_adjacency_matrix(adjmatrix,
mode = c("directed", "undirected","max", "min", "upper", "lower", "plus"),
weighted = NULL, diag = TRUE, add.colnames = NULL, add.rownames = NA)
```

A continuación se describe cada uno de los argumentos.

- *adjmatrix*: Matriz de adyacencia. En nuestro caso corresponde a los pases de la selección croata en la final del mundial.

- *mode*: Esta opción establece el tipo de grafo que se construirá. Si se desea un **grafo dirigido**, entonces `mode = "directed"`. En este caso cada elemento de la matriz da el número de aristas entre dos nodos, en este caso, pases y jugadores, respectivamente. Si se desea obtener un **grafo no dirigido** existen varias opciones.

  mode = "undirected": el grafo será no dirigido y el número de aristas que conecten a un par de jugadores (i, j) será igual al máximo entre el número de aristas en la dirección (i, j) y la dirección (j, i).

  mode = "max": el resultado es idéntico a `mode = "undirected"`.

  mode = "min": el grafo será no dirigido y el número de aristas que conecten a un par de jugadores (i, j) será igual al mínimo entre el número de aristas en la dirección (i, j) y la dirección (j, i).

  mode = "upper": el grafo será no dirigido utilizando solo la información por encima de la diagonal principal de la matriz, incluyendo la diagonal.

  mode = "lower": el grafo será no dirigido empleando solo la información por debajo de la diagonal principal de la matriz, incluyendo la diagonal.

  mode = "plus": el grafo será no dirigido y el número de aristas que conecten a un par de jugadores (i, j) debe ser igual a la suma del número de aristas en la dirección (i, j) con el número de aristas en la dirección (j, i).

- *weighted*: este argumento detalla si se debe crear un grafo ponderado o no. Por defecto el grafo es no ponderado. Si se desea un grafo ponderado, entonces `weighted = TRUE`.

- *diag*: especifica si la diagonal de la matriz de adyacencia debe ser tenida en cuenta en el grafo. Por defecto, la diagonal si es incluida (`diag = TRUE`). En nuestro caso, no tiene sentido un pase de un jugador a sí mismo, por lo que este argumento debe tomar el valor `diag = FALSE`.

- *add.colnames*: Este argumento permite añadir o no los nombres de las columnas a los nodos. Cuando toma el valor "NULL" (valor por defecto) los nombres de las columnas estarán presentes, mientras que, si toma el valor "NA" los nombres de las columnas no será añadidos.

- *add.rownames*: Análogo al argumento anterior, pero con las filas de la matriz (No obstante, en este caso el valor por defecto es NA).

Ahora, empleemos la información del objeto de clase *data.frame* que denominamos *Croacia* para construir un grafo no dirigido sin que tenga en cuenta la información de la diagonal principal de la matriz de adyacencia y con los nombres de las columnas como nombres de los nodos empleando la función *graph.adjacency*. Dado que esta función tiene como argumento un objeto de clase *matriz* y no *data.frame*, será necesario primero transformar los datos. Adicionalmente, noten que los datos que se leyeron del archivo de Excel no tenían información cuando los jugadores no se hacían pases. Por tanto, el objeto *Croacia* actualmente tiene elementos "NA". El primer paso entonces será remplazar los "NA" por ceros y después transformar el objeto a clase *matriz*. Después si podemos proceder a construir el grafo no dirigido omitiendo la diagonal principal de la matriz de adyacencia y con los nombres de las columnas como nombres de los nodos. Esto se logra con las siguientes líneas de código:

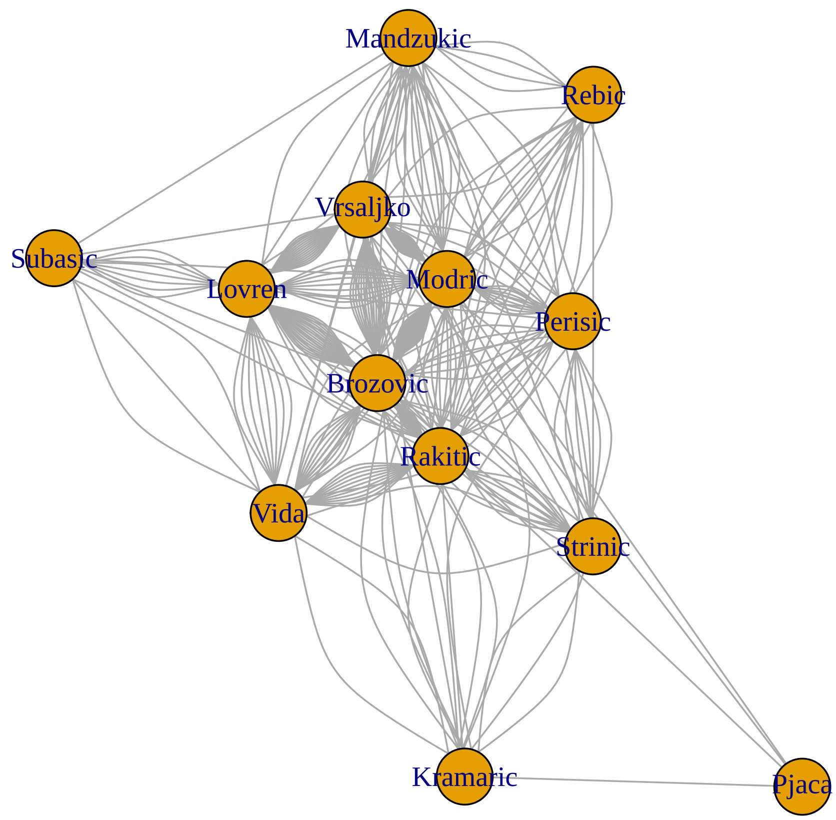
```
head(Croacia)

##   Subasic Vrsaljko Strinic Perisic Lovren Rakitic Modric Brozovic Mandzukic
## 1      NA        1      NA      NA      7       1      1        1         1
## 2       1       NA      NA       5     15       4     10       10         5
## 3      NA       NA      NA       6     NA       5     NA        4         1
## 4      NA        2       3      NA     NA       3      8       NA         1
## 5       3       10      NA       1     NA       8      9       17         3
## 6       1        2       6       5      5      NA      6        9         2
##   Rebic Vida Kramaric Pjaca
## 1    NA    3       NA    NA
## 2     2   NA       NA     1
## 3     1    1       NA    NA
## 4     1    2       NA    NA
## 5    NA    7       NA    NA
## 6     7    5        1    NA

Croacia[is.na(Croacia)] <- 0
Croacia <- as.matrix(Croacia)
class(Croacia)

## [1] "matrix"

head(Croacia)

##      Subasic Vrsaljko Strinic Perisic Lovren Rakitic Modric Brozovic Mandzukic
## [1,]       0        1       0       0      7       1      1        1         1
## [2,]       1        0       0       5     15       4     10       10         5
## [3,]       0        0       0       6      0       5      0        4         1
## [4,]       0        2       3       0      0       3      8        0         1
## [5,]       3       10       0       1      0       8      9       17         3
## [6,]       1        2       6       5      5       0      6        9         2
##      Rebic Vida Kramaric Pjaca
## [1,]     0    3        0     0
## [2,]     2    0        0     1
```

```
## [3,]     1    1        0     0
## [4,]     1    2        0     0
## [5,]     0    7        0     0
## [6,]     7    5        1     0

Croacia.Und.graph <- graph.adjacency(Croacia, mode = "undirected", diag = FALSE)
```

La última línea de código asignó un grafo con las características ya mencionadas al objeto *Croacia.Und.graph*. Este objeto tiene toda la información necesaria para trazar el grafo y otras estadísticas, pero, no lo muestra automáticamente. Para graficarlo se debe emplear la función *plot* de la siguiente manera:

```
plot(Croacia.Und.graph)
```

Figura 4: Grafo de los pases durante la final de Rusia 2018 del seleccionado croata



Fuente: Elaboración propia con datos de la FIFA

Sin embargo, la calidad de este gráfico puede ser mejorada, empleando las opciones del paquete *igraph*:

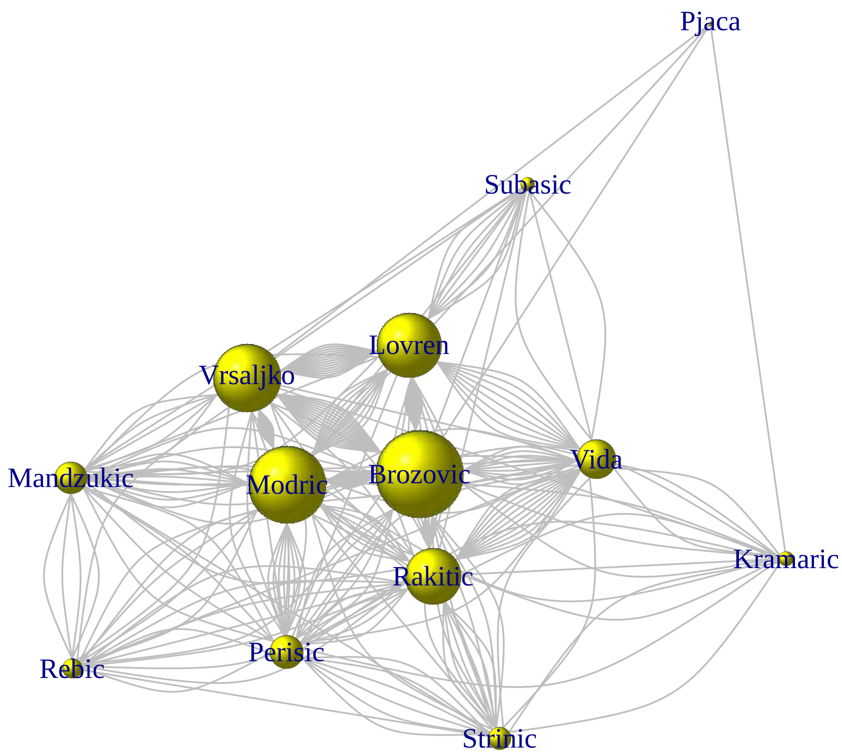
```
V(Croacia.Und.graph)$color <- "yellow"
V(Croacia.Und.graph)$shape <- "sphere"
E(Croacia.Und.graph)$color <- "gray"
```

El comando *V(.)$color* permite modificar el color de los nodos. De manera similar *V(.)$shape* permite modificar la forma que toman los nodos, teniendo como opciones “circle”, “square”, “csquare”, "rectangle", "crectangle", "vrectangle", "pie"(ver vertex.shape.pie), "sphere", y "none". Por otro lado, *E(.)$color* permite modificar el color de las aristas. Es importante aclarar que la función *V(.)* modifica los nodos y la función *E(.)* a las aristas.

Finalmente, podemos hacer que los nodos sean de tamaño diferente para incluir alguna de las métricas discutidas anteriormente. Por ejemplo, supongamos que queremos hacer que el tamaño de los nodos en nuestro grafo dependa del puntaje de autoridad. En este caso la función `authority.score(.)$vector` permite crear un vector que contiene el puntaje de autoridad para cada jugador. Estos valores se utilizan para diferenciar el tamaño de los nodos empleando las siguientes líneas de código:

```
hs1 <- authority.score(Croacia.Und.graph)$vector
plot(Croacia.Und.graph, vertex.size=hs1*25)
```

Figura 5: Segunda versión del Grafo de los pases durante la final de Rusia 2018 del seleccionado croata



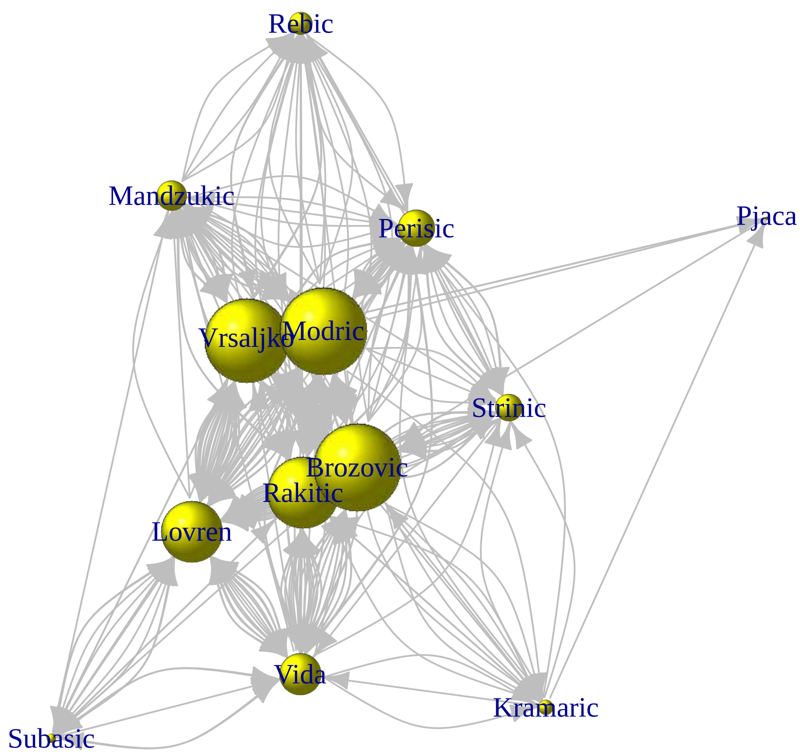
Fuente: Elaboración propia con datos de la FIFA

La calidad de la visualización mejoró bastante. Si se quisiera un grafo dirigido se procedería de la siguiente manera:

```
Croacia.Dir.graph <- graph.adjacency(Croacia, mode = "directed", diag = FALSE)
V(Croacia.Dir.graph)$color <- "yellow"
V(Croacia.Dir.graph)$shape <- "sphere"
E(Croacia.Dir.graph)$color <- "gray"
```

```r
hs1 <- authority.score(Croacia.Dir.graph)$vector
plot(Croacia.Dir.graph, vertex.size=hs1*25, edge.arrow.size=0.8)
```

Figura 6: Tercera versión del Grafo de los pases durante la final de Rusia 2018 del seleccionado croata



Fuente: Elaboración propia con datos de la FIFA

Se introdujeron dos pequeños cambios respecto a la programación anterior. Primero, se cambia la opción `mode` de `undirected` a `directed`. Segundo, en el argumento `plot` se introduce la opción `edge.arrow.size`, la cual controla el tamaño de las flechas del borde de cada arista. Esto último es necesario, dado que por defecto el tamaño de dichas flechas es lo suficientemente grande como para que la claridad del grafo sea nula, así que se debe ajustar a un tamaño adecuado.

5.2. Cálculo de las métricas de la Red

Ahora podemos proceder a calcular las métricas resumen del grafo que se discutieron en una de las secciones anteriores. Procedamos en el mismo orden de la sección anterior.

5.3. Centralidad

5.3.1. Medidas locales

- ***Grado de centralidad***: El grado de centralidad se puede calcular empleando la función *degree* del paquete *igraph*. Esta función tiene tres argumentos importantes. El primero es el grafo al que se le quiere calcular el grado de centralidad. El segundo argumento es denominado `mode`. Este argumento permite calcular el grado de centralidad teniendo en cuenta todos los enlaces (`mode = ‘‘total’’` o `mode = ‘‘all’’`), solo los enlaces saliendo (`mode = ‘‘out’’`) o solo los enlaces entrantes (`mode = ‘‘in’’`). El tercer argumento es `normalized` que determina si el grado de centralidad se normaliza o no; es decir, los resultados para cada uno de los *n* módulos se divide por *n-1* o no. Por defecto esta función no normaliza el grado de centralidad (`normalized = FALSE`). Si se desea normalizar, entonces se requiere fijar (`normalized = TRUE`). Para hacer nuestros resultados comparables con los del seleccionado de Francia, calculemos el grado de centralidad incluyendo tanto los enlaces de entrada como los de salida y sin normalizar[1]. Esto se puede hacer con la siguiente línea de código:

```
degree(Croacia.Und.graph, mode="total")

##   Subasic  Vrsaljko   Strinic   Perisic    Lovren   Rakitic    Modric  Brozovic
##        15        69        27        40        68        69        87       104
## Mandzukic     Rebic      Vida  Kramaric     Pjaca
##        35        23        44        17         4
```

  Según esta métrica, los jugadores que más dieron y recibieron pases en la final fueron Brozovic y Modric. Mientras que los jugadores menos activos en el equipo, en términos de pases, fueron Pjaca y Kramaric.

- ***Betweenness***: Esta métrica se puede calcular con la función *betweennes* del paquete *igraph*. Esta función tiene dos argumentos importantes. El primero es el grafo al que se le quiere calcular el grado de centralidad. El segundo argumento es `normalized` que determina si el grado de centralidad se normaliza o no. Por defecto esta función no normaliza el grado de centralidad (`normalized = FALSE`). Si se desea normalizar, entonces se requiere fijar (`normalized = TRUE`). Para hacer nuestros resultados comparables con los de la selección francesa, calculemos el **Betweenness** sin normalizar. Esto se puede hacer con la siguiente línea de código:

```
betweenness(Croacia.Und.graph)

##     Subasic    Vrsaljko     Strinic     Perisic      Lovren     Rakitic
## 0.000000000 2.336537714 0.074299223 0.408524883 0.175000000 2.278725916
##      Modric    Brozovic   Mandzukic       Rebic        Vida    Kramaric
## 4.401537430 6.109251656 0.433124848 0.009433962 1.253863947 0.519700420
##       Pjaca
## 0.000000000
```

[1] En este caso el número de jugadores de Francia que participa en la final es muy similar a los de Croacia. Por esta razón la normalización no es necesaria.

Esta medida sugiere que los jugadores más propensos a actuar como puentes entre sus compañeros fueron Brozovic y Modric, precisamente los que más dieron y recibieron pases. Por otro lado, los jugadores Subasic y Pjaca nunca actuaron como puentes pues tuvieron un valor nulo en el **betweenness**.

5.3.2. Medidas globales

- ***Closeness***: Esta métrica se puede calcular empleando la función *closeness* del paquete *igraph*. Esta función tiene tres argumentos importantes. El primero es el grafo al que se le quiere calcular el *Closeness*. El segundo argumento es denominado `mode`. Este argumento permite calcular el *Closeness* teniendo en cuenta todas los rutas (`mode = ‘‘all’’`), solo las rutas saliendo (`mode = ‘‘out’’`) o solo las rutas entrantes (`mode = ‘‘in’’`). El tercer argumento es `normalized` que determina si el *Closeness* se normaliza o no; es decir, los resultados para cada uno de los n módulos se divide por $n-1$ o no. Por defecto esta función no normaliza el grado de centralidad (`normalized = FALSE`). Si se desea normalizar, entonces se requiere fijar (`normalized = TRUE`). Para hacer nuestros resultados comparables con los del seleccionado de Francia, calculemos el *closeness* incluyendo tanto los enlaces de entrada como los de salida y sin normalizar. Esto se puede hacer con la siguiente línea de código:

```
closeness(Croacia.Und.graph, mode =  "all")

##    Subasic   Vrsaljko    Strinic    Perisic     Lovren    Rakitic     Modric
## 0.05882353 0.07692308 0.06250000 0.07142857 0.06250000 0.07692308 0.08333333
##   Brozovic  Mandzukic      Rebic       Vida   Kramaric      Pjaca
## 0.08333333 0.07142857 0.06250000 0.07692308 0.06250000 0.05000000
```

  De nuevo, en términos de centralidad (pero ahora global) sobresalen Brozovic y Modric, y los peores puntuados son Subasic y Pjaca.

- ***Puntaje de autoridad***: Esta métrica se puede calcular empleando la función *authority.score* del paquete *igraph*. Esta función tiene solo un argumento importante; el grafo al que se le quiere calcular el puntaje de autoridad. El puntaje de autoridad para cada uno de los módulos se puede obtener de la siguiente manera:

```
authority.score(Croacia.Und.graph)$vector

##    Subasic   Vrsaljko    Strinic    Perisic     Lovren    Rakitic     Modric
## 0.15694159 0.77719921 0.25573956 0.37877595 0.73846071 0.64157987 0.88221620
##   Brozovic  Mandzukic      Rebic       Vida   Kramaric      Pjaca
## 1.00000000 0.36573244 0.22360943 0.44623143 0.16460428 0.04355986
```

  Según esta medida el jugador más influyente de Croacia en la final de Rusia 2018 fue Brozovic, seguido por Modric. Mientras que los menos influyentes fueron Pjaca y Subasic.

5.4. Otras medidas para detectar nodos importantes

- ***Distancia promedio***: Esta métrica se puede calcular empleando la función *mean_distance* del paquete *igraph*. Esta función tiene solo un argumento importante; el grafo al que se le quiere calcular la distancia promedio. Esta estadística se puede obtener de la siguiente manera:

```r
mean_distance(Croacia.Und.graph)

## [1] 1.230769
```

Dada la cercanía de esta métrica a uno, se deduce que en la mayoría de los casos bastaba con un pase para que un par de jugadores se conectaran.

- ***Excentricidad***: Esta métrica se puede calcular empleando la función *eccentricity* del paquete *igraph*. Esta función tiene dos argumentos importantes. El primero es el grafo al que se le quiere calcular la excentricidad. El segundo argumento es denominado `mode`. Este argumento permite calcular la excentricidad teniendo en cuenta todas los rutas (`mode = ‘‘all’’`, esta es la opción predeterminada), solo las rutas saliendo (`mode = ‘‘out’’`) o solo las rutas entrantes (`mode = ‘‘in’’`). Para hacer nuestros resultados comparables con los del seleccionado de Francia, calculemos excentricidad incluyendo tanto los enlaces de entrada como los de salida. Esto se puede hacer con la siguiente línea de código:

```r
eccentricity(Croacia.Und.graph, mode = "all")

##   Subasic  Vrsaljko   Strinic   Perisic    Lovren   Rakitic    Modric  Brozovic
##         2         2         2         2         2         2         1         1
## Mandzukic     Rebic      Vida  Kramaric     Pjaca
##         2         2         2         2         2
```

Esta medida, al igual que las anteriores, reflejan el buen trabajo de los jugadores Brozovic y Modric, pues a estos les bastaba un pase para conectarse con sus compañeros más lejanos en la red. En cuanto a los demás jugadores, se requería de dos pases para que lograran conectarse con sus compañeros más lejanos.

5.5. Medidas agregadas para la red

- ***Diámetro***: La función que calcula esta métrica es *diameter* del paquete *igraph*. Esta función tiene solo un argumento importante; el grafo al que se le quiere calcular el diámetro. Esta estadística se puede obtener de la siguiente manera:

```r
diameter(Croacia.Und.graph)

## [1] 2
```

Esta medida nos muestra que el par de jugadores más alejados en la red podían conectarse por medio de dos pases.

- ***Densidad***: Esta métrica se puede calcular empleando la función *edge_ density* del paquete *igraph*. Esta función tiene solo un argumento importante; el grafo al que se le quiere calcular el diámetro. Como se mencionó antes, la matriz de adyacencia se debe transformar para que contenga solo dos valores: ceros (en caso de que no haya conexión) y unos (cuando hay conexión). Esta transformación y el correspondiente cálculo de la densidad del grafo se puede obtener de la siguiente manera:

```r
Croacia.1.0<-Croacia
for (i in 1:ncol(Croacia.1.0)){
for (j in 1:nrow(Croacia.1.0)){
```

```r
  Croacia.1.0[j,i]<-ifelse(Croacia.1.0[j,i]>0,1,Croacia.1.0[j,i])
}
}
GrafDens <- graph.adjacency(Croacia.1.0, mode = "undirected", weighted = NULL,
diag = FALSE, add.colnames=NULL)
edge_density(GrafDens)

## [1] 0.7692308
```

Esta medida nos muestra que Croacia, durante la final de Rusia 2018, hizo efectivas aproximadamente el 77 % de todas las conexiones posibles.

- ***Transitividad***: Esta métrica se puede calcular empleando la función *transitivity* del paquete *igraph*. Esta función tiene solo un argumento importante; el grafo al que se le quiere calcular la transitividad. Esta estadística se puede obtener de la siguiente manera:

```r
transitivity(Croacia.Und.graph)

## [1] 0.8269962
```

Según esta medida, la proporción de ocurrencias de triadas transitivas fue aproximadamente de 83 %.

- ***Reciprocidad***: Esta métrica se puede calcular empleando la función *reciprocity* del paquete *igraph*. Esta función tiene solo un argumento importante; el grafo al que se le quiere calcular la transitividad. Esta estadística se puede obtener de la siguiente manera:

```r
reciprocity(Croacia.Und.graph)

## [1] 1
```

Esta medida sugiere que siempre existió reciprocidad entre un par de jugadores de Croacia en la final de Rusia 2018.

Cuadro 5: Estadísticas descriptivas por nodo de la red para la final de Rusia 2018 para la selección de Croacia

	G. de Central.	betweenness	closeness	Excentricidad	Punt. de Auto.
Subasic	15.00	0.00	0.06	2.00	0.16
Vrsaljko	69.00	2.34	0.08	2.00	0.78
Strinic	27.00	0.07	0.06	2.00	0.26
Perisic	40.00	0.41	0.07	2.00	0.38
Lovren	68.00	0.18	0.06	2.00	0.74
Rakitic	69.00	2.28	0.08	2.00	0.64
Modric	87.00	4.40	0.08	1.00	0.88
Brozovic	104.00	6.11	0.08	1.00	1.00
Mandzukic	35.00	0.43	0.07	2.00	0.37
Rebic	23.00	0.01	0.06	2.00	0.22
Vida	44.00	1.25	0.08	2.00	0.45
Kramaric	17.00	0.52	0.06	2.00	0.16
Pjaca	4.00	0.00	0.05	2.00	0.04

Fuente: Elaboración propia con datos de la FIFA
Notas: G. de Central. y Punt. de Auto. representan el Grado de Centralidad y el Puntaje de Autoridad, respectivamente.

Cuadro 6: Estadísticas descriptivas de la red para la final de Rusia 2018 - Croacia

Distancia	Diametro	Densidad	Transitividad	Reciprocidad
1.23	2.00	0.77	0.83	1.00

Fuente: Elaboración propia con datos de la FIFA

Los resultados de las métricas para el comportamiento del equipo croata durante la final de la copa mundo 2018 se presentan en los cuadros 5 y 6. Una comparación de estas estadísticas con las del equipo francés revela algunos hechos interesantes. Primero, en términos del número de pases, Croacia superó ampliamente a Francia. Una comparación del desempeño de los tres mejores jugadores de cada equipo, en términos del *betweenness*, revela que Francia tuvo un mejor rendimiento. Según este resultado, el equipo francés contó con mejores jugadores a la hora de actuar como puentes de conexión entre otros. Sin embargo, la estadística *closeness* revela que Modric, Rakitic y Brozovic estuvieron más cerca del resto de su equipo (se conectaban con mayor facilidad) frente a Griezman, Pogba y Giroud. Además, la excentricidad promedio del equipo croata es menor a la del equipo francés, gracias a que los jugadores Modric y Brozovic se podían conectar con sus compañeros más lejanos por medio de solo un pase. Finalmente, en concordancia con los resultados arrojados por la comparación de grados y cercanía, el equipo croata tuvo una menor distancia entre sus jugadores, y una mayor densidad y transitividad.

6. Mas sobre grafos y otras visualizaciones

En esta sección discutiremos maneras alternativas de presentar los grafos.

6.1. Grafo ponderado

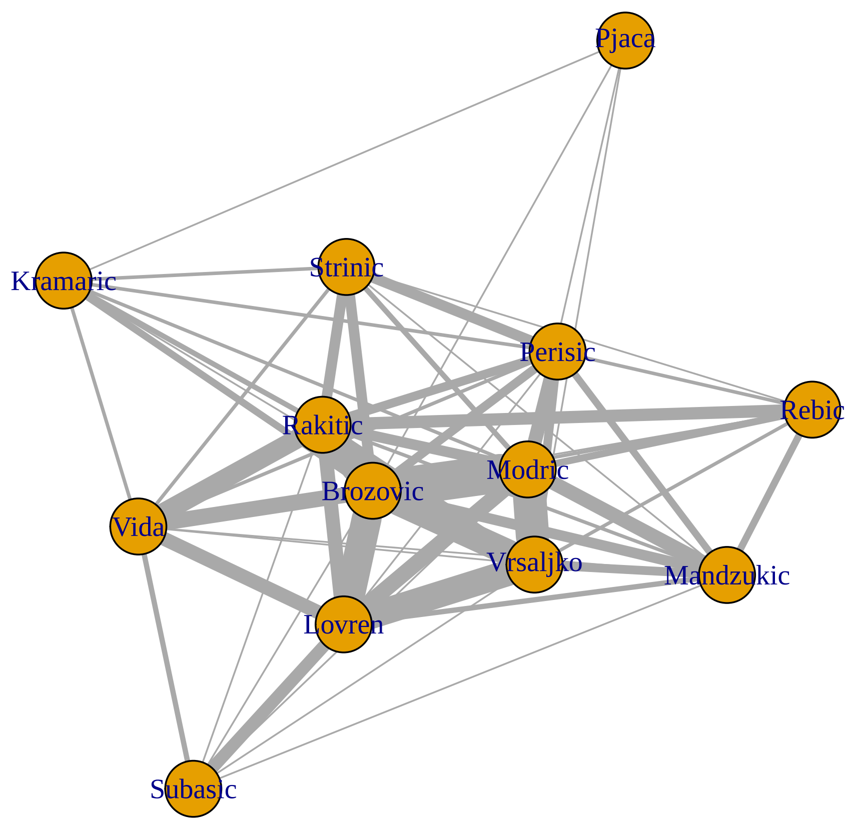
Anteriormente se discutió la diferencia de un grafo ponderado y uno no ponderado. Hasta el momento hemos construido grafos no ponderaros, a continuación, se muestra cómo construir un grafo ponderado. Recordemos que un grafo ponderado es aquel que conecta los nodos por una sola línea (independientemente del número de enlaces que existen entre ellos) y dicha línea tiene asociada una etiqueta o algo que de alguna manera revele el número de conexiones que unen ese par de nodos. Para construir este tipo de grafos, debemos partir de un grafo no dirigido que nos permita emplear alguna forma de ponderar las aristas. Por ejemplo,

```
Croacia.Und.Pond.graph <- graph.adjacency(Croacia, weighted = TRUE, mode = "undirected",
                          diag = FALSE)
```

El lector puede constatar que este grafo aún no está ponderado. Los pesos son calculados, pero no graficados por defecto. Podemos cambiar el grosor de las aristas del grafo empleando el atributo *edge.width* de la función *plot* del paquete *igraph*. Esto se puede hacer de la siguiente manera:

```
plot(Croacia.Und.Pond.graph, edge.width=E(Croacia.Und.Pond.graph)$weight)
```

Figura 7: Grafo ponderado de la final de Rusia 2018 del seleccionado croata



Fuente: Elaboración propia con datos de la FIFA

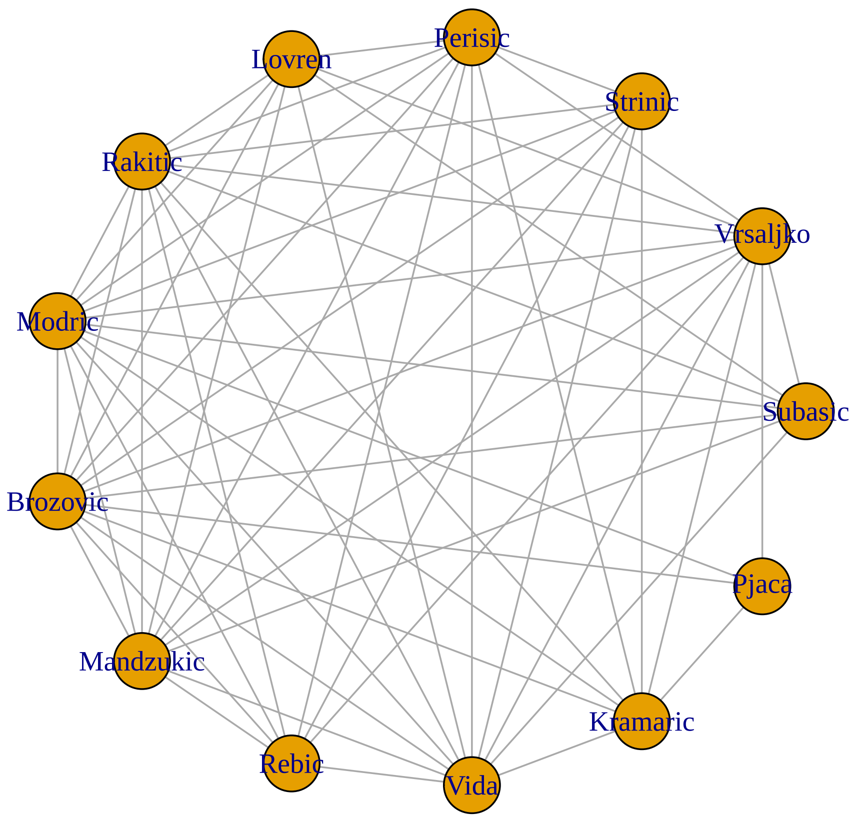
6.2. Disposición del grafo

La disposición de los nodos de un grafo responde a algoritmos que asignan las coordenadas de los nodos de una red. Y la verdad esto es arbitrario; es decir, no existe una forma adecuada de disponer los nodos, existen muchas maneras y el investigador deberá decidir cuál es la que permite visualizar mejor las características de la red.

El argumento *layout* de la función *plot* del paquete *igraph* permite emplear diferentes algoritmos. Continuando con el grafo ponderado no direccionado construido en la sección anterior, podemos, por ejemplo, organizar todos los nodos alrededor de un circulo empleando la siguiente línea de código:

```
plot(Croacia.Und.Pond.graph, layout= layout_in_circle(Croacia.Und.Pond.graph))
```

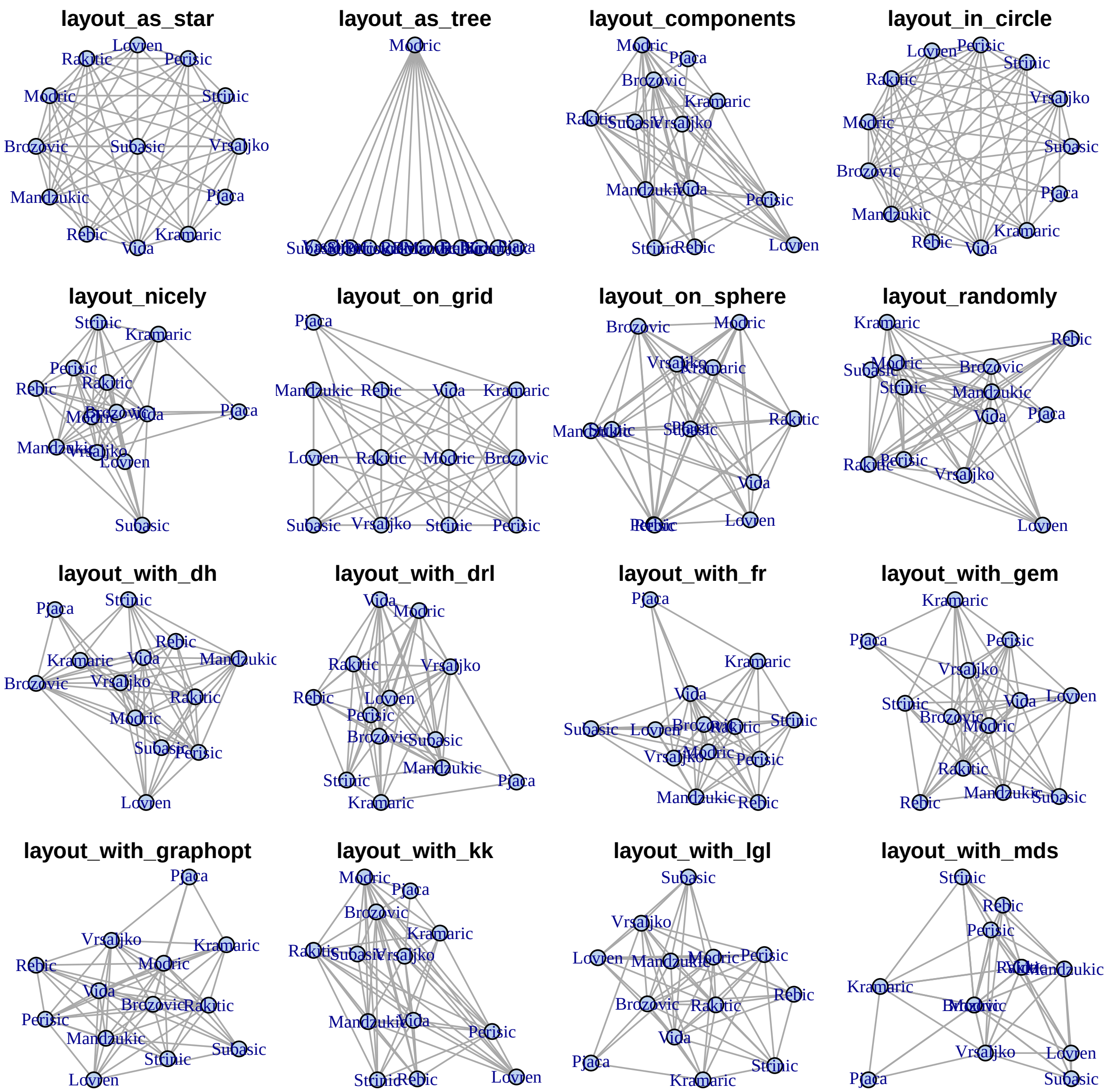
Figura 8: Grafo ponderado en forma circular no direccionado de la final de Rusia 2018 - Croacia



Fuente: Elaboración propia con datos de la FIFA

En la siguiente figura se presentan las múltiples opciones (que tienen sentido para nuestra red) de disposición disponibles en el paquete *igraph*.

Figura 9: Opciones de disposición para grafos ponderado (que tienen sentido para nuestra red) de la final de Rusia 2018 - Croacia



Fuente: Elaboración propia con datos de la FIFA

Como se mencionó es importante que los investigadores seleccionen una disposición del grafo que mejor refleje las características más importantes de la red. No existe una forma correcta de disposición de un grafo, pero si existen unas buenas prácticas. Por ejemplo, es deseable que cuando se construye un grafo se:

- minimice el número de aristas que se cruzan
- no permita la sobreposición de nodos

- garantice hasta donde sea posible que todas las aristas tengan la misma longitud
- logre simetría en el grafo
- ubique los nodos más influyentes en el centro de la red

6.3. Otra manera de representar la red - *heatmap*

En algunas ocasiones puede resultar conveniente representar gráficamente la matriz de adyacencia mediante un mapa de calor. Este es un gráfico de colores con diferente intensidad. En nuestro caso, un mayor grado de intensidad reflejará una mayor cantidad de pases entre jugadores. Este tipo de gráficos se puede hacer empleando el paquete *ggplot2* (Wickham, 2016) (Ver Alonso y González (2012) para una introducción a este paquete).

El primer paso es crear un *data.frame* que tenga la información de adyacencia. Esto se puede lograr fácilmente con la función *melt* del paquete *reshape* (Wickham, 2007). Partiendo de la matriz de adyacencia que denominamos *Croacia*, podemos primero asegurarnos que tanto las filas como las columnas tengan nombre y luego si podemos crear la nueva base de datos. Esto se puede hacer de la siguiente manera:

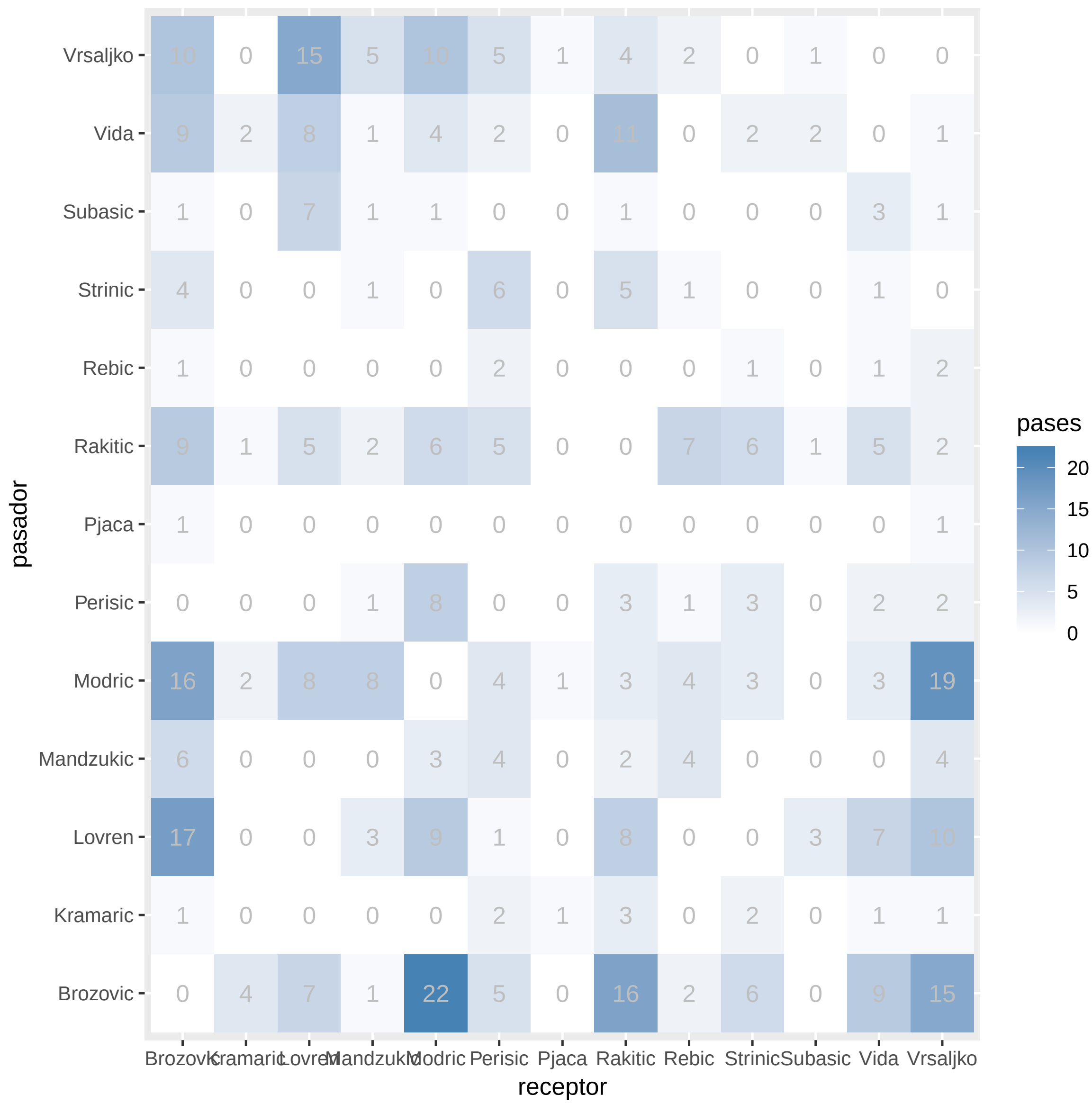
```
row.names(Croacia) <- variable.names(Croacia)
Croacia.por.parejas <- melt(Croacia)
head(Croacia.por.parejas)

##         X1      X2 value
## 1  Subasic Subasic     0
## 2 Vrsaljko Subasic     1
## 3  Strinic Subasic     0
## 4  Perisic Subasic     0
## 5   Lovren Subasic     3
## 6  Rakitic Subasic     1
```

Esto creó un objeto denominado *Croacia.por.parejas* que tiene todas las entradas de la matriz de adyacencia. La primera columna corresponde a la fila (el receptor del pase), la segunda columna (el pasador) y la tercera columna corresponde al valor del correspondiente elemento de la matriz de adyacencia. Esta forma de presentar la información de una red se conoce como una lista de aristas (en inglés se conoce como *Edgelist*).

Ahora podemos construir el correspondiente mapa de calor empleando el paquete *ggplot2*. Esto se logra con la siguiente línea de código.

```
ggplot(data = Croacia.por.parejas, aes(x= X2, y= X1, fill=value)) +  geom_tile() +
  geom_text(aes(X2, X1, label = value), color = "grey", size = 4) +
  scale_fill_gradient(low = "white", high = "steelblue") +
  xlab("receptor") + ylab("pasador") + labs(fill = "pases")
```

Figura 10: Mapa de calor de la matriz de adyacencia del equipo croata durante la final de Rusia 2018



Fuente: Elaboración propia con datos de la FIFA

En el gráfico se puede observar que resaltan duplas como Modric-Brozovic, Brozovic-Rakitic, Subasic-Lovren, entre otras.

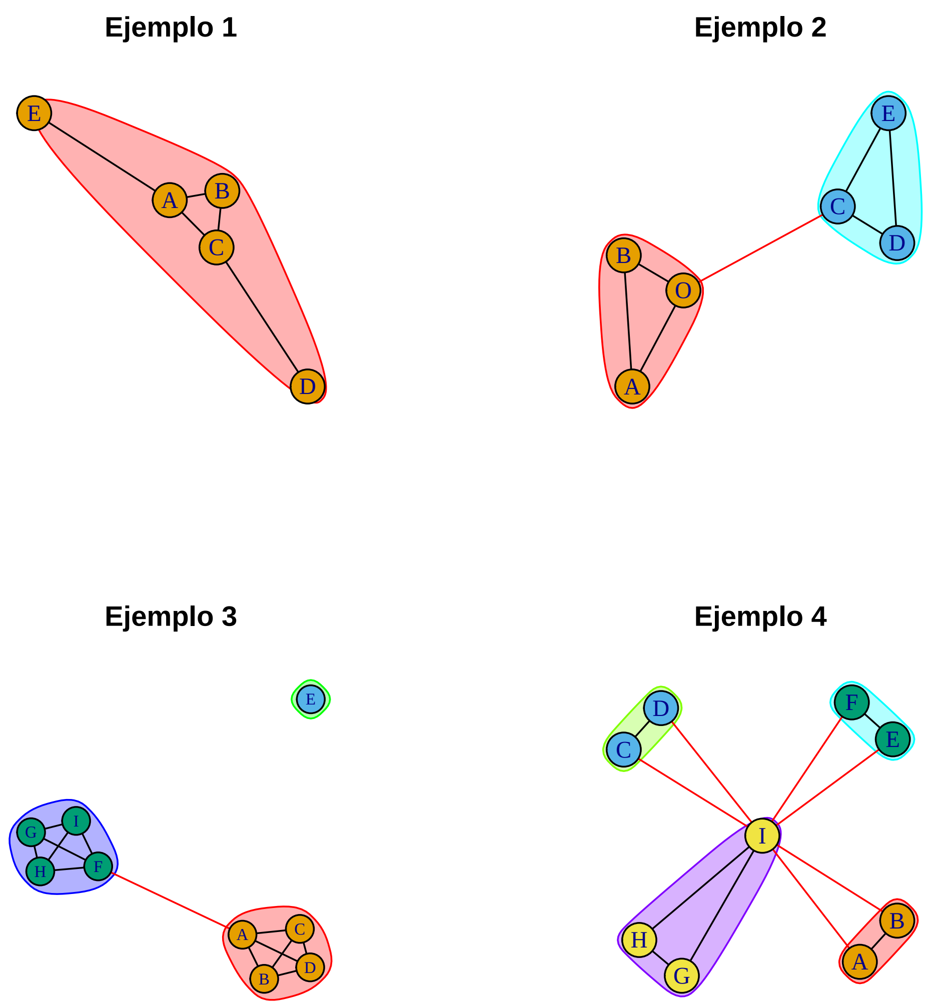
7. Detención de comunidades

(Nota: esta sección supone que el lector está familiarizado con las técnicas de cluster jerárquico.)

En algunas oportunidades será importante detectar subredes al interior de una red. Al subconjunto de nodos que más estrechamente está relacionado se le puede denominar como comunidad. Algunas redes están formadas por comunidades que están densamente conectadas interiormente pero escasamente conectados a otras sub-redes o en otras ocasiones las comunidades pueden estar muy relacionadas entre sí.

En el siguiente gráfico se presentan unos ejemplos de diferentes comunidades al interior de cuatro redes.

Figura 11: Algunos ejemplos de diferentes redes y comunidades



Fuente: Elaboración propia

El ejemplo uno muestra una red en la que existe una sola comunidad. Aunque, al interior de esta los jugadores A, B y C son cercanos entre sí, y a la vez, están relativamente alejados de los jugadores D y E. No obstante, esta aparente lejanía no es lo suficientemente importante como para que la red tenga más de una comunidad. Por otro lado, el ejemplo dos muestra una red con dos comunidades, las cuales se comunican a través de la interacción de los jugadores C y O. En este sentido, el ejemplo tres es similar al dos, dado que en este último, dos de las tres comunidades se relacionan gracias a la interacción de dos jugadores. Sin embargo, en el ejemplo tres existe una tercera comunidad, compuesta por un solo jugador, que no se relaciona con las demás. Finalmente, el ejemplo 4 muestra una red con 4 comunidades. En esta red, la comunidad compuesta por tres jugadores interactúa directamente con cada uno de los miembros de las demás gracias a uno de sus jugadores (el jugador I), mientras que las otras comunidades (las que están compuestas por dos jugadores) se comunican entre sí exclusivamente de manera indirecta a través del jugador I.

En nuestro caso de un equipo de fútbol, una comunidad puede entenderse como esas sociedades que se forman al interior de un equipo. Para detectar comunidades existen diferentes algoritmos. Tal vez el más sencillo se basa en la métrica de *betweenness*, también conocida como el método Newman-Girvanuna. La idea es una aplicación de la técnica de segmentación de clusteres jerárquicos. El método implica remover secuencialmente las aristas con la métrica *betweenness* más alta y recalcular la métrica en cada paso y así seleccionar la partición que permite la mejor segmentación de la red.

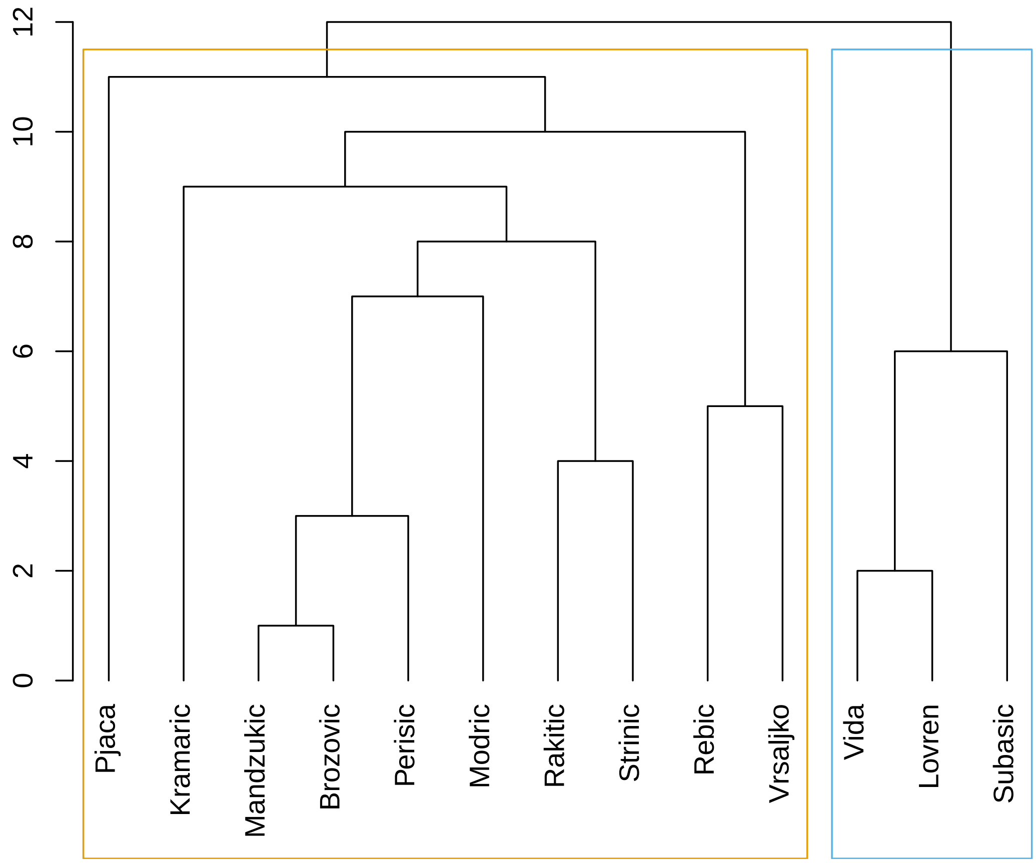
Para realizar un ejemplo de este tipo de análisis para detectar comunidades, partamos del grafo no direccionado ponderado construido en la sección anterior (objeto *Croacia.Und.Pond.graph*). El paquete *igraph* permite detectar las comunidades mediante la función *cluster_ edge_ betweenness*. El argumento más importante de esta función es el grafo que se quiere segmentar. El procedimiento es el siguiente:

```
HC.Croacia.Und.Pond.graph <- cluster_edge_betweenness(Croacia.Und.Pond.graph)
```

Ahora podemos visualizar el dendrograma del ejercicio de segmentación jerárquica. Esto lo podemos hacer de la siguiente manera, especificando que se muestren el número de comunidades que queremos indicado (como se hace en cualquier análisis de cluster jerárquico) mediante el argumento *rect*.

```
dendPlot(HC.Croacia.Und.Pond.graph, mode="hclust", rect = 2)
```

Figura 12: Comunidades en la final de Rusia 2018 - Croacia

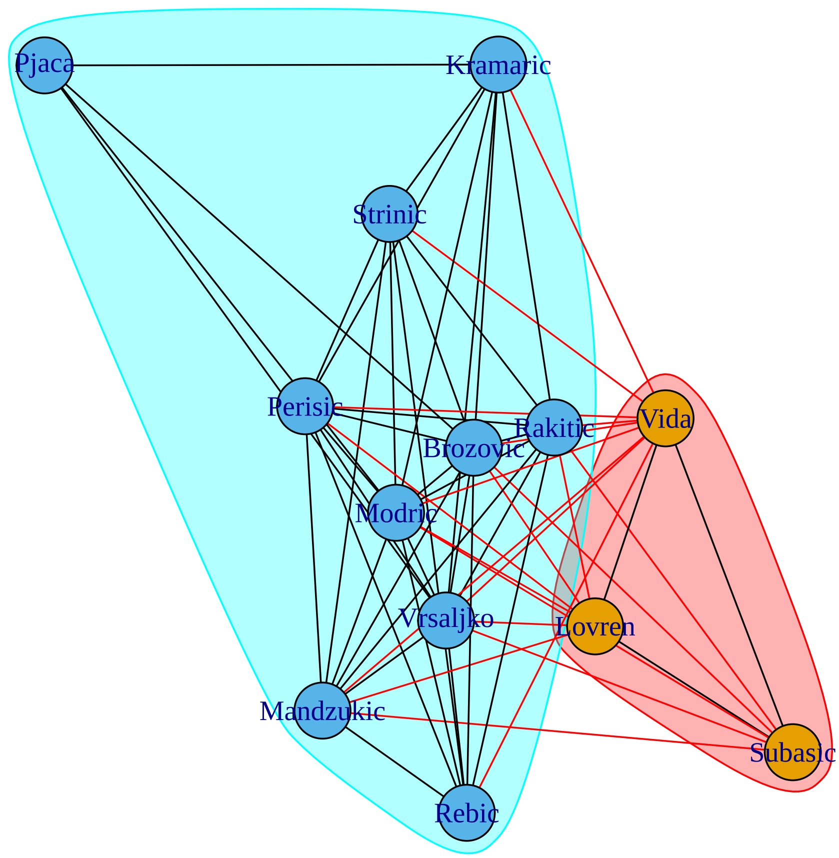


Fuente: Elaboración propia con datos de la FIFA

Del dendrograma se puede intuir la presencia de dos clusters. Ahora podemos proceder a graficarlos mediante la función *plot* de la siguiente manera:

```
plot(HC.Croacia.Und.Pond.graph, Croacia.Und.Pond.graph)
```

Figura 13: Comunidades en la final de Rusia 2018 - Croacia



Fuente: Elaboración propia con datos de la FIFA

Si se desea conocer el número de comunidades y la membresía de cada nodo a una comunidad, esto se puede realizar de la siguiente manera:

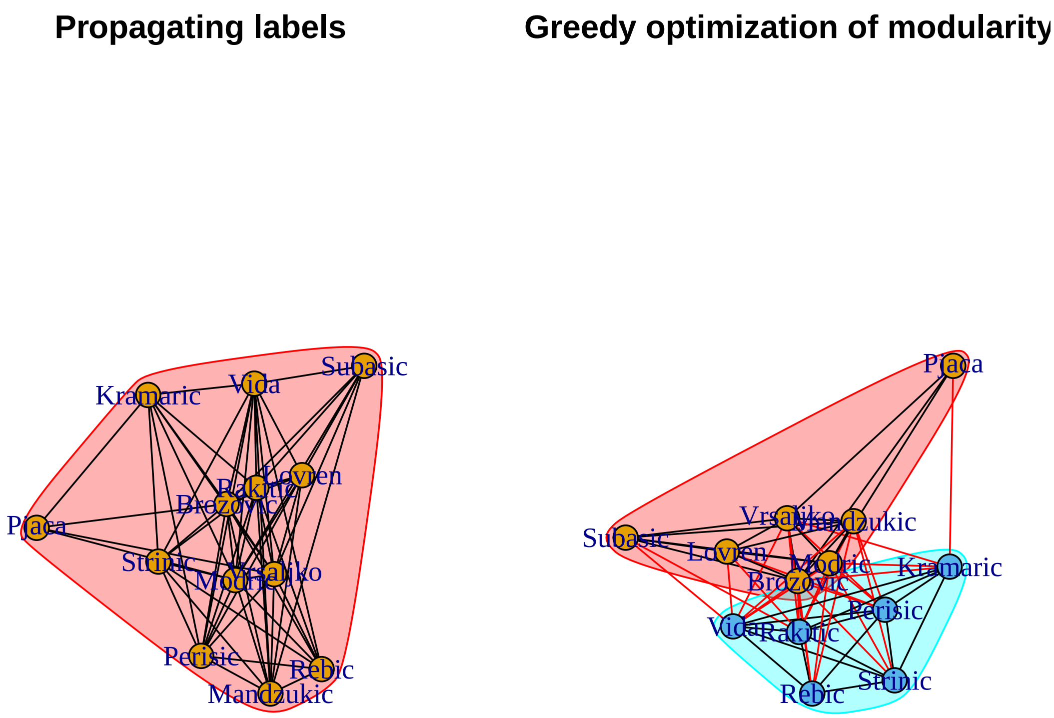
```
length(HC.Croacia.Und.Pond.graph)

## [1] 2

membership(HC.Croacia.Und.Pond.graph)
```

```
##   Subasic  Vrsaljko   Strinic   Perisic    Lovren   Rakitic    Modric  Brozovic
##         1         2         2         2         1         2         2         2
## Mandzukic     Rebic      Vida  Kramaric     Pjaca
##         2         2         1         2         2
```

Estos resultados sugieren que los jugadores Vida, Lovren y Subasic son los menos relacionados con la sub-red o comunidad principal, la cual está densamente conectada y tiene como centro los jugadores Modric, Brozovic y Rakitic.

El paquete *igraph* también permite emplear otros métodos de detección de comunidades como el basado en propagación de etiquetas (propagating labels), optimización codiciosa modular (greedy optimization of modularity). Invitamos al lector para que estudie estas posibilidades por su cuenta. En la siguiente figura se muestran los resultados de aplicar estos algoritmos. Noten que la respuesta puede cambiar sustancialmente con el algoritmo empleado. Pero esa discusión está por fuera del alcance de este documento.

Figura 14: Comunidades en la final de Rusia 2018 con diferentes algoritmos - Croacia



Fuente: Elaboración propia con datos de la FIFA

8. Comentarios finales

El análisis de redes se está convirtiendo en una necesidad para la ciencia económica, dado que permite entender y explicar algunos hechos que la teoría económica tradicional no puede. Respondiendo a esta necesidad, en este documento presentamos una breve introducción a los grafos y al uso del paquete *igraph* (Csardi y Nepusz, 2006) de R (R Core Team, 2018). Este paquete permite generar y describir grafos de gran calidad de manera rápida y sencilla. Se hace hincapié en que esta es solo una introducción, pues aquí solo se mostró una pequeña parte de lo que estos pueden hacer. El lector debería de empezar por reproducir lo que se hizo con los datos de Croacia para el equipo de Francia. También es importante estudiar la documentación del paquete y algunos de los numerosos blogs que existen sobre el tema.

Referencias

Alonso, Julio Cesar y Alejandra González (2012). "Ggplot: gráficos de alta calidad". En: *Apuntes de Economía* 33, pág. 29.

Avrachenkov, Konstantin E, Aleksei Yu Kondratev y Vladimir V Mazalov (2017). "Cooperative Game Theory Approaches for Network Partitioning". En: *International Computing and Combinatorics Conference*. Springer, págs. 591-602.

Calvo-Armengol, Antoni y Matthew O Jackson (2004). "The effects of social networks on employment and inequality". En: *American economic review* 94.3, págs. 426-454.

Csardi, Gabor y Tamas Nepusz (2006). "The igraph software package for complex network research". En: *InterJournal* Complex Systems, pág. 1695. URL: `http://igraph.org`.

De Benedictis, Luca y col. (2014). "Network Analysis of World Trade using the BACI-CEPII dataset". En: *Global Economy Journal* 14.3-4, págs. 287-343.

Granovetter, Mark (1978). "Threshold models of collective behavior". En: *American journal of sociology* 83.6, págs. 1420-1443.

— (1985). "Economic action and social structure: The problem of embeddedness". En: *American journal of sociology* 91.3, págs. 481-510.

Hendricks, Ken, Michele Piccione y Guofu Tan (1999). "Equilibria in networks". En: *Econometrica* 67.6, págs. 1407-1434.

Lusher, Dean, Garry Robins y Peter Kremer (2010). "The application of social network analysis to team sports". En: *Measurement in physical education and exercise science* 14.4, págs. 211-224.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/`.

Salazar, Boris y Daniel Otero (2015). "La revolución de los nuevos clásicos&58; redes, influencia y metodología". En: *Revista de Economía Institucional* 17.32, págs. 39-69.

Travers, Jeffrey y Stanley Milgram (1977). "An experimental study of the small world problem". En: *Social Networks*. Elsevier, págs. 179-197.

Wickham, Hadley (2007). "Reshaping data with the reshape package". En: *Journal of Statistical Software* 21.12. URL: `http://www.jstatsoft.org/v21/i12/paper`.

— (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN: 978-3-319-24277-4. URL: `http://ggplot2.org`.

Wickham, Hadley y Jennifer Bryan (2018). *readxl: Read Excel Files*. R package version 1.1.0. URL: `https://CRAN.R-project.org/package=readxl`.