Sumario

[# Cadenas de transmisión 1](#__RefHeading___Toc31989_2034561403)

[Resumen 1](#__RefHeading___Toc31991_2034561403)

[Preparación 1](#__RefHeading___Toc31993_2034561403)

[Cargar paquetes 1](#__RefHeading___Toc33248_485595530)

[Importar datos 2](#__RefHeading___Toc33250_485595530)

[Creación de un objeto epicontactos 2](#__RefHeading___Toc33252_485595530)

[Manejando 3](#__RefHeading___Toc31995_2034561403)

[Subconjunto 3](#__RefHeading___Toc33254_485595530)

[Acceso a las identificaciones 3](#__RefHeading___Toc33256_485595530)

[Visualización 4](#__RefHeading___Toc31997_2034561403)

[Trazado básico 4](#__RefHeading___Toc33258_485595530)

[Visualización de los atributos de los nodos 4](#__RefHeading___Toc33260_485595530)

[Visualización de los atributos de los bordes 5](#__RefHeading___Toc33262_485595530)

[Eje temporal 5](#__RefHeading___Toc33264_485595530)

[Especificación de la forma del árbol de transmisión 5](#__RefHeading___Toc33266_485595530)

[Cómo guardar los gráficos y las cifras 6](#__RefHeading___Toc33268_485595530)

[Líneas de tiempo 6](#__RefHeading___Toc33270_485595530)

[Análisis 7](#__RefHeading___Toc31999_2034561403)

[Resumiendo 7](#__RefHeading___Toc33272_485595530)

[Características de los pares 7](#__RefHeading___Toc33274_485595530)

[Identificación de grupos de empresas 7](#__RefHeading___Toc33276_485595530)

[Cálculo de grados 8](#__RefHeading___Toc33278_485595530)

[Recursos 8](#__RefHeading___Toc32001_2034561403)

# # Cadenas de transmisión

{#transmission-chains}

## Resumen

La principal herramienta para manejar, analizar y visualizar las cadenas de transmisión y los datos de rastreo de contactos es el paquete **epicontacts**, desarrollado por la gente de RECON. Pruebe el gráfico interactivo que se muestra a continuación pasando el cursor por encima de los nodos para obtener más información, arrastrándolos para moverlos y clicando sobre ellos para resaltar los casos posteriores.

## Preparación

### Cargar paquetes

Primero cargue los paquetes estándar necesarios para la importación y manipulación de datos. En este manual destacamos p\_load() de **pacman**, que instala el paquete si es necesario y lo carga para su uso. También se pueden cargar paquetes con library() desde **el .** Consulta la página sobre [los fundamentos de R](#r-basics) para obtener más información sobre los paquetes de R.

Necesitará la versión de desarrollo de **epicontacts**, que puede instalarse desde github utilizando la función p\_install\_github() de **pacman**. Sólo necesita ejecutar este comando a continuación una vez, no cada vez que utiliza el paquete (a partir de entonces, puede utilizar p\_load() como de costumbre).

### Importar datos

Importamos el conjunto de datos de casos de una epidemia de ébola simulada. Si desea descargar los datos para seguirlos paso a paso, consulta las instrucciones en la página de [descarga de manuales y datos](#download-handbook-and-data). El conjunto de datos se importa utilizando la función import() del paquete **rio**. Consulta la página sobre [importación y exportación](#import-and-export) para conocer las distintas formas de importar datos.

A continuación se muestran las primeras 50 filas del listado. Son especialmente interesantes las columnas case\_id, generation, infector y source.

### Creación de un objeto epicontactos

A continuación, tenemos que crear un objeto **epicontactos**, que requiere dos tipos de datos:

* un listado que documenta los casos en los que las columnas son variables y las filas corresponden a casos únicos
* una lista de aristas que definen los vínculos entre los casos sobre la base de sus identificadores únicos (pueden ser contactos, eventos de transmisión, etc.)

Como ya tenemos un listado, sólo tenemos que crear una lista de aristas entre los casos, más concretamente entre sus ID. Podemos extraer los enlaces de transmisión del listado vinculando la columna infector con la columna case\_id. En este punto también podemos añadir "propiedades de borde", con lo que nos referimos a cualquier variable que describa el vínculo entre los dos casos, no los casos en sí. Por ejemplo, añadiremos una variable de localización que describa la ubicación del evento de transmisión, y una variable de duración que describa la duración del contacto en días.

En el código siguiente, la función **dplyr** transmutar es similar a mutar, excepto que sólo mantiene las columnas que hemos especificado dentro de la función. La función drop\_na filtrará cualquier fila en la que las columnas especificadas tengan un valor NA; en este caso, sólo queremos mantener las filas en las que se conoce el infector.

Ahora podemos crear el objeto **epicontactos** utilizando la función make\_epicontacts. Necesitamos especificar qué columna del listado apunta al identificador único del caso, así como qué columnas de los contactos apuntan a los identificadores únicos de los casos involucrados en cada enlace. Estos enlaces son direccionales en el sentido de que la infección va del infector al caso, por lo que necesitamos especificar los argumentos desde y hacia en consecuencia. Por lo tanto, también establecemos el argumento dirigido a TRUE, que afectará a las operaciones futuras.

Al examinar los objetos **epicontactos**, podemos ver que la columna case\_id del listado ha sido renombrada a id y las columnas case\_id e infector de los contactos han sido renombradas a from y to. Esto garantiza la coherencia en las operaciones posteriores de manipulación, visualización y análisis.

## Manejando

### Subconjunto

El método subset() para los objetos epicontactos permite, entre otras cosas, filtrar las redes en función de las propiedades del listado ("atributos de nodos") y de la base de datos de contactos ("atributos de aristas"). Estos valores deben pasarse como listas con nombre al argumento respectivo. Por ejemplo, en el código que sigue mantenemos en linelist sólo los casos masculinos que tienen una fecha de infección entre abril y julio de 2014 (las fechas se especifican como rangos), y los enlaces de transmisión que se produjeron en el hospital.

Podemos utilizar la función thin para filtrar linelist para incluir los casos que se encuentran en los contactos estableciendo el argumento what = "linelist", o filtrar los contactos para incluir los casos que se encuentran en linelist estableciendo el argumento what = "contacts". En el código siguiente, filtramos aún más el objeto epicontactos para mantener sólo los enlaces de transmisión que implican los casos masculinos infectados entre abril y julio que habíamos filtrado anteriormente. Podemos ver que sólo dos enlaces de transmisión conocidos se ajustan a esa especificación.

Además de la subdivisión por atributos de nodos y aristas, las redes pueden podarse para incluir sólo los componentes que están conectados a ciertos nodos. El argumento cluster\_id toma un vector de IDs de casos y devuelve linelist de individuos que están vinculados, directa o indirectamente, a esos IDs. En el código siguiente, podemos ver que un total de 13 casos del listado están involucrados en los clusters que contienen 2ae019 y 71577a.

El método subset() para los objetos epicontactos también permite filtrar por tamaño de cluster usando los argumentos cs, cs\_min y cs\_max. En el código siguiente, estamos manteniendo sólo los casos vinculados a clusters de 10 casos o más, y podemos ver que 271 casos del listado están involucrados en tales clusters.

### Acceso a las identificaciones

La función get\_id() recupera información sobre los ID de los casos en el conjunto de datos, y puede parametrizarse como sigue:

* lista de **líneas**: IDs en los datos del listado
* **contactos**: IDs en el conjunto de datos de los contactos ("desde" y "hasta" combinados)
* **from**: IDs en la columna "from" de los datos del contacto
* **a** los identificadores de la columna "a" de los datos de los contactos
* **todos**: Las identificaciones que aparecen en cualquier parte de cualquiera de los conjuntos de datos
* **común**: identificaciones que aparecen tanto en el conjunto de datos de contactos como en linelist

Por ejemplo, ¿cuáles son las diez primeras identificaciones de los datos de contactos?

¿Cuántas identificaciones se encuentran tanto en linelist como en los contactos?

## Visualización

### Trazado básico

Todas las visualizaciones de los objetos **epicontactos** son manejadas por la función plot. En primer lugar, filtraremos el objeto **epicontactos** para incluir solo los casos con fechas de inicio en junio de 2014 utilizando la función de subconjunto, y solo incluiremos los contactos vinculados a esos casos utilizando la función thin.

A continuación, podemos crear el gráfico básico e interactivo de la siguiente manera:

Puedes mover los nodos arrastrándolos, pasar por encima de ellos para obtener más información y clicar en ellos para resaltar los casos conectados.

Hay un gran número de argumentos para modificar este gráfico. Aquí cubriremos los principales, pero consulta la documentación a través de ?vis\_epicontacts (la función a la que se llama cuando se utiliza plot en un objeto **epicontacts**) para obtener una descripción completa de los argumentos de la función.

#### Visualización de los atributos de los nodos

El color del nodo, la forma del nodo y el tamaño del nodo se pueden asignar a una columna determinada en linelist utilizando los argumentos node\_color, node\_shape y node\_size. Esto es similar a la sintaxis aes que puede reconocer de **ggplot2**.

Los colores, formas y tamaños específicos de los nodos pueden especificarse de la siguiente manera:

* **Los colores** a través del argumento col\_pal, ya sea proporcionando una lista de nombres para la especificación manual de cada color como se hace a continuación, o proporcionando una función de paleta de colores como colorRampPalette(c("negro", "rojo", "naranja")), que proporcionaría un gradiente de colores entre los especificados.
* **Formas** pasando una lista con nombre al argumento shapes, especificando una forma para cada elemento único en la columna del listado especificada por el argumento node\_shape. Ver codeawesome para las formas disponibles.
* **Tamaño** pasando un rango de tamaño de los nodos al argumento size\_range.

Aquí un ejemplo, donde el color representa el resultado, la forma el género y el tamaño la edad:

#### Visualización de los atributos de los bordes

El color, la anchura y el tipo de línea de los bordes pueden asignarse a una columna determinada del dataframe de los contactos utilizando los argumentos edge\_color, edge\_width y edge\_linetype. Los colores y anchos específicos de los bordes se pueden especificar como sigue:

* **Colores** a través del argumento edge\_col\_pal, de la misma manera que se utiliza para col\_pal.
* **Anchos** pasando un rango de tamaño de los nodos al argumento width\_range.

Aquí un ejemplo:

### Eje temporal

También podemos visualizar la red a lo largo de un eje temporal asignando el argumento eje\_x a una columna del listado. En el ejemplo siguiente, el eje x representa la fecha de inicio de los síntomas. También hemos especificado el argumento arrow\_size para asegurarnos de que las flechas no son demasiado grandes, y hemos establecido label = FALSE para que la figura esté menos recargada.

Hay un gran número de argumentos adicionales para especificar aún más cómo se visualiza esta red a lo largo de un eje temporal, que puede comprobar a través de ?vis\_temporal\_interactive (la función que se llama cuando se utiliza plot en un objeto **epicontactos** con el eje x especificado). A continuación veremos algunos.

#### Especificación de la forma del árbol de transmisión

Hay dos formas principales que puede adoptar el árbol de transmisión, especificadas mediante el argumento network\_shape. La primera es una forma ramificada, como se muestra arriba, en la que un borde recto conecta dos nodos cualesquiera. Esta es la representación más intuitiva, pero puede dar lugar a la superposición de aristas en una red densamente conectada. La segunda forma es el rectángulo, que produce un árbol parecido a una filogenia. Por ejemplo:

A cada nodo del caso se le puede asignar una posición vertical única mediante el argumento position\_dodge. La posición de los casos no conectados (es decir, sin contactos reportados) se especifica utilizando el argumento unlinked\_pos.

La posición del nodo padre respecto a los nodos hijos puede especificarse mediante el argumento parent\_pos. La opción por defecto es colocar el nodo padre en el centro, sin embargo puede colocarse en la parte inferior (parent\_pos = 'bottom') o en la parte superior (parent\_pos = 'top').

#### Cómo guardar los gráficos y las cifras

Puedes guardar un gráfico como un archivo html interactivo y autónomo con la función visSave del paquete **VisNetwork**:

Guardar estas salidas de red como una imagen es desafortunadamente menos fácil y requiere que guardes el archivo como un html y luego tomes una captura de pantalla de este archivo usando el paquete webshot. En el código siguiente, estamos convirtiendo el archivo html guardado anteriormente en un PNG:

### Líneas de tiempo

También se pueden incluir líneas de tiempo en la red, que se representan en el eje de abscisas de cada caso. Esto puede servir para visualizar la ubicación de los casos, por ejemplo, o el tiempo hasta el resultado. Para generar una línea de tiempo, tenemos que crear un data.frame de al menos tres columnas que indiquen el ID del caso, la fecha de inicio del "evento" y la fecha de finalización del "evento". También se puede añadir cualquier número de otras columnas que luego se pueden asignar a las propiedades de los nodos y aristas de la línea de tiempo. En el código siguiente, generamos una línea de tiempo que va desde la fecha de inicio de los síntomas hasta la fecha del desenlace, y mantenemos las variables de desenlace y hospital que utilizamos para definir la forma y el color de los nodos. Ten en cuenta que puede tener más de una fila/evento de la línea de tiempo por caso, por ejemplo si un caso es transferido entre varios hospitales.

A continuación, pasamos el elemento de la línea de tiempo al argumento de la línea de tiempo. Podemos mapear los atributos de la línea de tiempo a los colores, formas y tamaños de los nodos de la línea de tiempo de la misma manera definida en las secciones anteriores, excepto que tenemos dos nodos: el nodo de inicio y el nodo final de cada línea de tiempo, que tienen argumentos separados. Por ejemplo, tl\_start\_node\_color define qué columna de la línea de tiempo se asigna al color del nodo inicial, mientras que tl\_end\_node\_shape define qué columna de la línea de tiempo se asigna a la forma del nodo final. También podemos asignar el color, la anchura, el tipo de línea y las etiquetas al borde de la línea de tiempo mediante los argumentos tl\_edge\_\*.

Consulta ?vis\_temporal\_interactive (la función a la que se llama cuando se traza un objeto epicontacto) para obtener documentación detallada sobre los argumentos. Cada argumento está anotado también en el código de abajo:

## Análisis

### Resumiendo

Podemos obtener una visión general de algunas de las propiedades de la red utilizando la función de resumen.

Por ejemplo, podemos ver que sólo el 57% de los contactos tienen ambos casos en linelist; esto significa que no tenemos datos del listado sobre un número significativo de casos involucrados en estas cadenas de transmisión.

### Características de los pares

La función get\_pairwise() permite procesar la(s) variable(s) del listado según cada par de los datos de contactos. En el siguiente ejemplo, la fecha de inicio de la enfermedad se extrae del listado para calcular la diferencia entre la fecha de inicio de la enfermedad para cada par. El valor que se obtiene de esta comparación representa el **intervalo de serie (si)**.

La función get\_pairwise() interpretará el tipo de la columna que se utiliza para la comparación, y ajustará su método de comparación de los valores en consecuencia. Para los números y las fechas (como en el ejemplo de **si**), la función restará los valores. Cuando se aplica a columnas que son caracteres o categóricas, get\_pairwise() pegará los valores. Dado que la función también permite un procesamiento arbitrario (véase el argumento "f"), estas combinaciones discretas pueden ser fácilmente tabuladas y analizadas.

En este caso, se observa una asociación significativa entre los vínculos de transmisión y el género.

### Identificación de grupos de empresas

La función get\_clusters() puede utilizarse para identificar componentes conectados en un objeto epicontacto. En primer lugar, la utilizamos para recuperar un data.frame que contenga la información de los clusters:

Veamos los clusters más grandes. Para ello, añadimos la información de los clústeres al objeto epicontactos y luego lo subconjuntamos para mantener sólo los clústeres más grandes:

### Cálculo de grados

El grado de un nodo corresponde a su número de aristas o conexiones con otros nodos. get\_degree() proporciona un método sencillo para calcular este valor para las redes de epicontactos. Un grado alto en este contexto indica un individuo que estuvo en contacto con muchos otros. El argumento type indica que queremos contar tanto el grado de entrada como el de salida, el argumento only\_linelist indica que sólo queremos calcular el grado para los casos del listado.

¿Qué personas son las que tienen más contactos?

¿Cuál es el número medio de contactos?

## Recursos

La [página de epicontacts](https://www.repidemicsconsortium.org/epicontacts/index.html) ofrece una visión general de las funciones del paquete e incluye algunas viñetas más detalladas.

La [página de github](http://github.com/reconhub/epicontacts) puede utilizarse para plantear problemas y solicitar funciones.