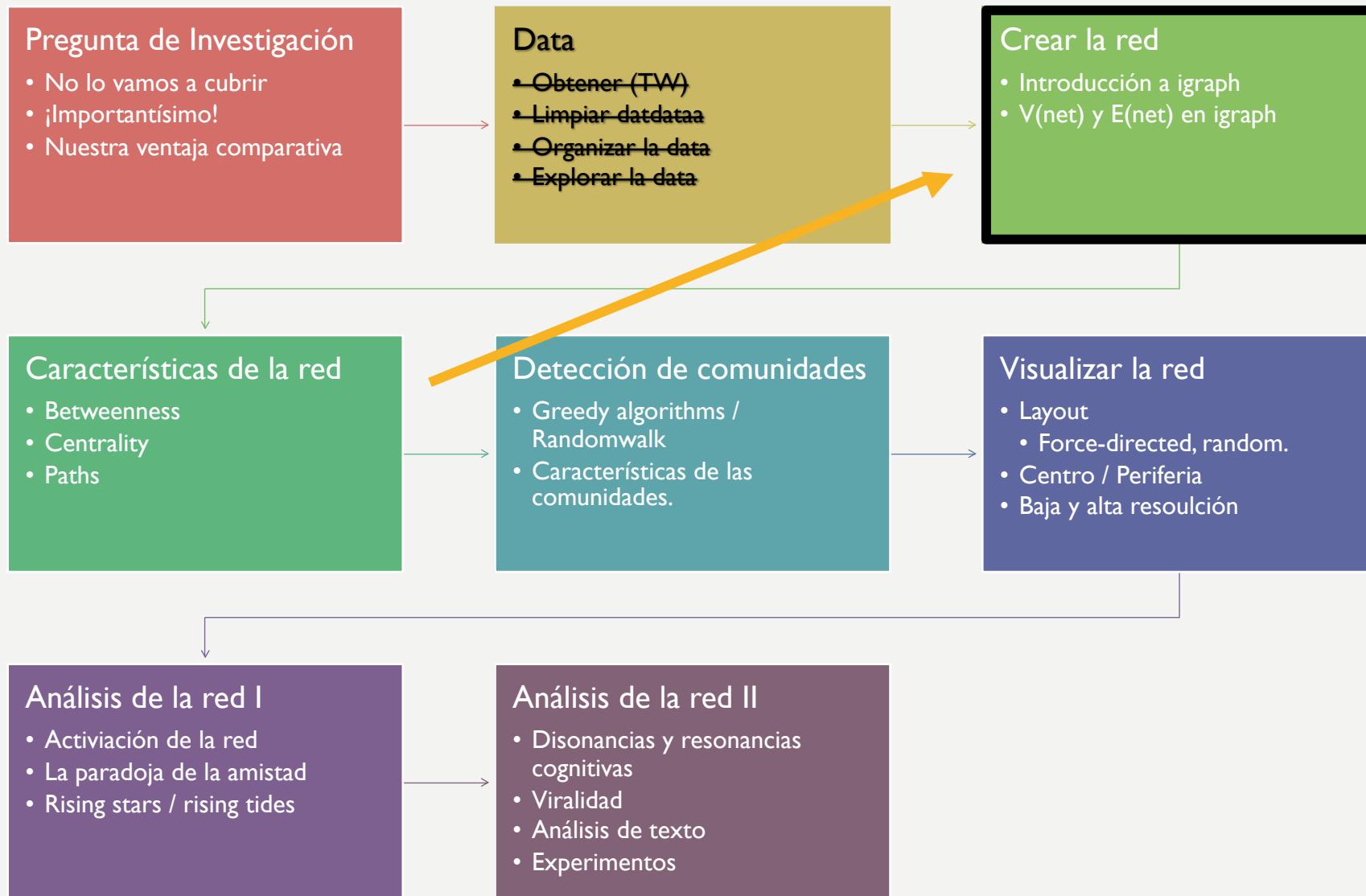


CREAR LA RED

(INTRODUCCIÓN A *IGRAPH*)





INTRODUCCIÓN A IGRAPH

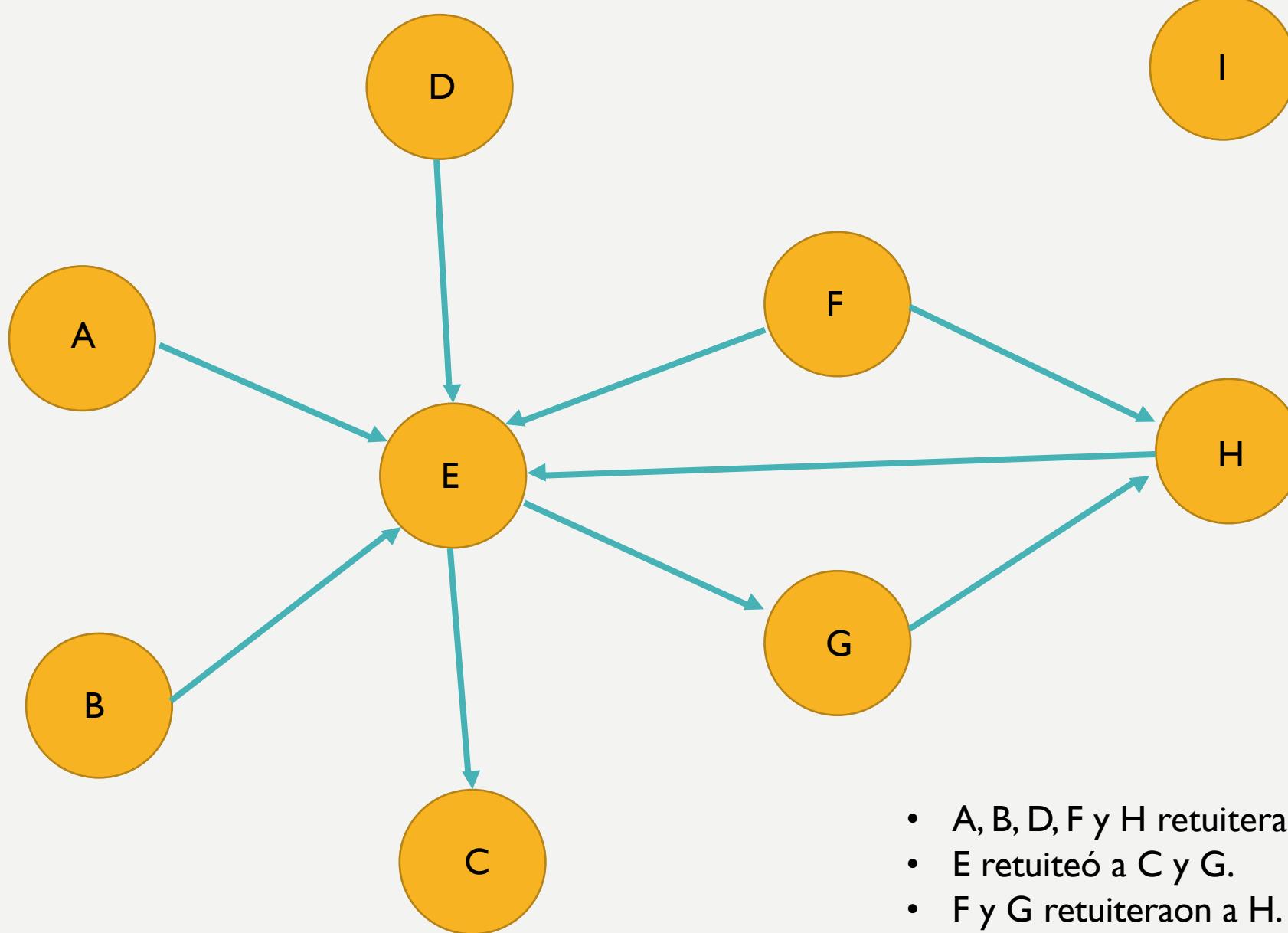
V(NET) Y E(NET)

NUESTRA RED CON IGRAPH

- Para crear nuestra red en R nos vamos a valer del paquete *igraph* (Csardi y Nepusz, 2006).
- Antes de crear nuestra red debemos pensar quiénes van a ser nuestros nodos y qué los van a unir (i.e., cuáles van a ser nuestros edges o conexiones).
- Entonces... ¿cuáles son nuestro nodos y cuáles son nuestros edges?
- En nuestra red de Twitter, nuestros nodos van a ser los usuarios y estos usuarios serán conectados (existirá un edge entre los dos) cuando uno retuitee a otro.

NUESTRA RED CON IGGRAPH

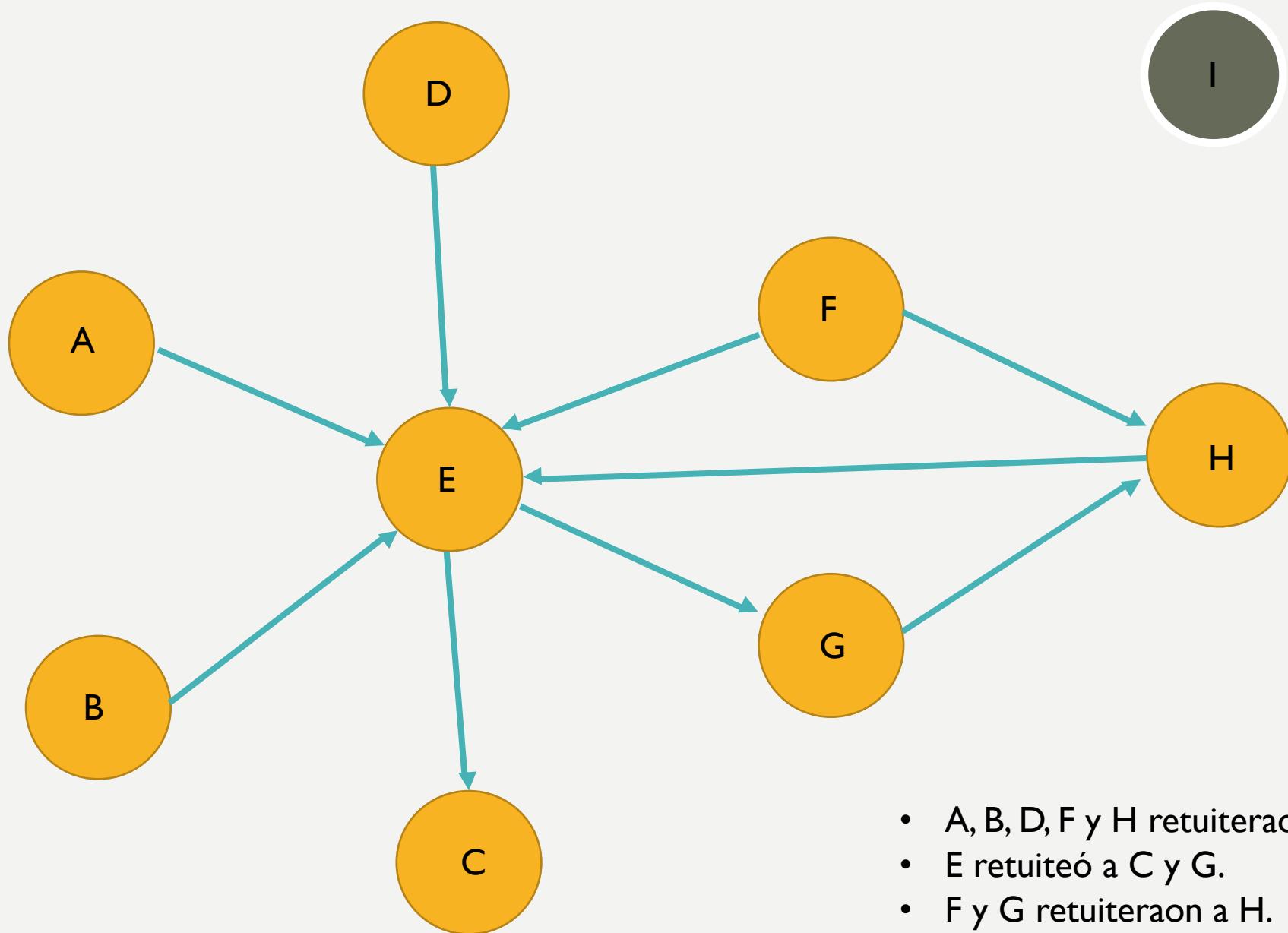
- Vamos a darle nombre a este tipo de usuarios. El usuario que crea el tuit original lo vamos a llamar **autoridad** (o *authority*). El usuario que reproduce el tuit original (el que da RT) lo vamos a llamar **hub**.
- Es decir que, y aquí nos vamos a poner un poco matemáticos, la relación entre nodos se dará así: $H_{rtuit} \rightarrow A_{tuit}$
- Esto significa que vamos a tener una red con conexiones *direccionalas*. No va a ser lo mismo, en términos de la red, alguien que produce un tuit y quien lo reproduce (más adelante hablaremos más de esto).
- Y se va a ver algo así...



- A, B, D, F y H retuitearon a E.
- E retuiteó a C y G.
- F y G retuitearon a H.
- I no retuiteó ni fue retuiteado

NUESTRA RED CON IGRAPH

- Por lo tanto, la información principal para nuestra red es los usuarios que retuitearon y los que fueron retuiteados.
- ¿Qué pasa con aquellos nodos que no hicieron ni lo uno ni lo otro?



- A, B, D, F y H retuitearon a E.
- E retuiteó a C y G.
- F y G retuitearon a H.
- I no retuiteó ni fue retuiteado

NUESTRA RED CON IGRAPH

- Por lo tanto, la información principal para nuestra red es los usuarios que retuitearon y los que fueron retuiteados.
- ¿Qué pasa con aquellos nodos que no hicieron ni lo uno ni lo otro?
- Esos nodos no proporcionan información. Están aislados y, por lo tanto, "hablando al vacío". Por lo tanto, nos deshacemos de ellos.
- Ahora sí, armemos nuestra red en igraph...

NUESTRA RED CON IGRAPH

- Como siempre, comenzamos cargando nuestros paquetes necesario:

```
> ## Carga las bibliotecas necesarias: ----  
>  
> library(tidyverse) # tidyverse nos ayuda con la manipulación de datos  
> library(viridis) # viridis nos da colores bonitos  
> library(igraph) # Con igraph creamos y evaluamos nuestra red  
>
```

- Y eliminamos los nodos que no están conectados:

```
> ## Primero nos quedamos únicamente con los tuits que tuvieron RTs
> ## El resto no nos da información
>
> duque_rts <- duque_tw[!is.na(duque_tw$retweet_screen_name),]
```

- Extraemos únicamente los nombres de los hubs y de las autoridades, y creamos una matriz:

```
> ## Creamos una matriz únicamente con los hubs y las autoridades  
>  
> t_duque <- duque_rts[c(4,22)]  
> t_duque <- as.matrix(t_duque) # la matriz  
> View(t_duque)
```

	screen_name	retweet_screen_name
1	EdisonS93934866	carlosecaicedo
2	EdisonS93934866	mrafael70
3	EdisonS93934866	Marianiniecheve
4	EdisonS93934866	elespectador
5	EdisonS93934866	AlexanderRuaP
6	EdisonS93934866	a_colombianos
7	WILOSOL111	AlexLopezMaya
8	WILOSOL111	carlosecaicedo
9	WILOSOL111	mrafael70
10	Rossy7773	VTVcanal8
11	Rossy7773	VTVcanal8
12	Antohes08	a_colombianos
13	Antohes08	mrafael70
14	Antohes08	derlilopeza
15	AMartinGuio	Carolalon1
16	manchozongo	pulzo
17	manchozongo	SandraComunes
18	manchozongo	NoticiasUno
19	Isaura07324497	mayertuberquia
20	Isaura07324497	IvanCepedaCast

NUESTRA RED:

```
> ## Ahora creamos nuestra red
>
> duque_net <- graph.empty() # Un red vacía
> duque_net <- add.vertices(duque_net, length(unique(c(t_duque))), # Nuestros nodos serán todos nuestros usuarios
+                               name=as.character(unique(c(t_duque)))) # Y tendrán como nombre... su nombre de usuario
>
> duque_net <- add.edges(duque_net, t(t_duque)) # Trasponemos la matriz para crear las conexiones
>
> summary(duque_net) # Una red con 82817 nodos y 252482 conexiones (edges)
IGRAPH f47ca82 DN-- 82817 252482 --
+ attr: name (v/c)
```

- En realidad lo que *igraph* está haciendo es la siguiente operación: AA^T .
- En nuestro caso, la matriz A tiene dimensiones 2×252482 por lo tanto la matriz resultante de la multiplicación será de dimensiones...
- ¡Correcto! 252482×252482 . Es una tremenda matriz. Sin embargo, es una matriz dispersa (*sparse matrix*) con un montón de ceros.
- Por lo tanto, *igraph* toma únicamente las celdas que no son ceros, y crea así los vértices (edges) entre los nodos.

V(NET) Y E(NET)

- Lo que ahora tenemos es un objeto *igraph* al cual le podemos agregar información.
- El “lenguaje” de *igraph* es un poco diferente al que estamos acostumbrados en R. Esto se debe a que el objeto de la red (o grafo) tiene dos tipos de información diferente: la información de los nodos o vectores—representada por V(net)— y la información de las conexiones o edges—representada por E(net)—.
- Por lo tanto, cuando agregamos información a nuestra red, debemos primero pensar si esta es información acerca de los *nodos* o de las *conexiones*.

E(NET)

```
> ## Primero las conexiones
>
> # Utilizamos la forma E(net) para agregar información a las conexiones
> E(duque_net)$text <- duque_rts$text # El texto del tuit
> E(duque_net)$text_width <- duque_rts$display_text_width # El largo del texto
> E(duque_net)$urls <- duque_rts$urls_url # Si hay algun link
>
> E(duque_net)$friends_hub <- duque_rts$friends_count # Siguiendo del hub
> E(duque_net)$followers_hub <- duque_rts$followers_count # Seguidores del hub
> E(duque_net)$time_hub <- duque_rts$created_at # hora en que hub dio RT
> E(duque_net)$verified_hub <- duque_rts$verified # Si el hub es verificado
>
> E(duque_net)$friends_auto <- duque_rts$retweet_friends_count # Siguiendo de la auto
> E(duque_net)$followers_auto <- duque_rts$retweet_followers_count # Seguidores de la auto
> E(duque_net)$time_auto <- duque_rts$retweet_created_at # hora en que auto creo tuit
> E(duque_net)$verified_auto <- duque_rts$retweet_verified # Si el auto es verificado
>
> summary(duque_net) # Una red con más información
IGRAPH 1a2e0af DN-- 82817 252482 --
+ attr: name (v/c), text (e/c), text_width (e/n), urls (e/x), friends_hub (e/n), followers_hub
| (e/n), time_hub (e/n), verified_hub (e/l), friends_auto (e/n), followers_auto (e/n), time_auto
| (e/n), verified_auto (e/l)
```

V(NET)

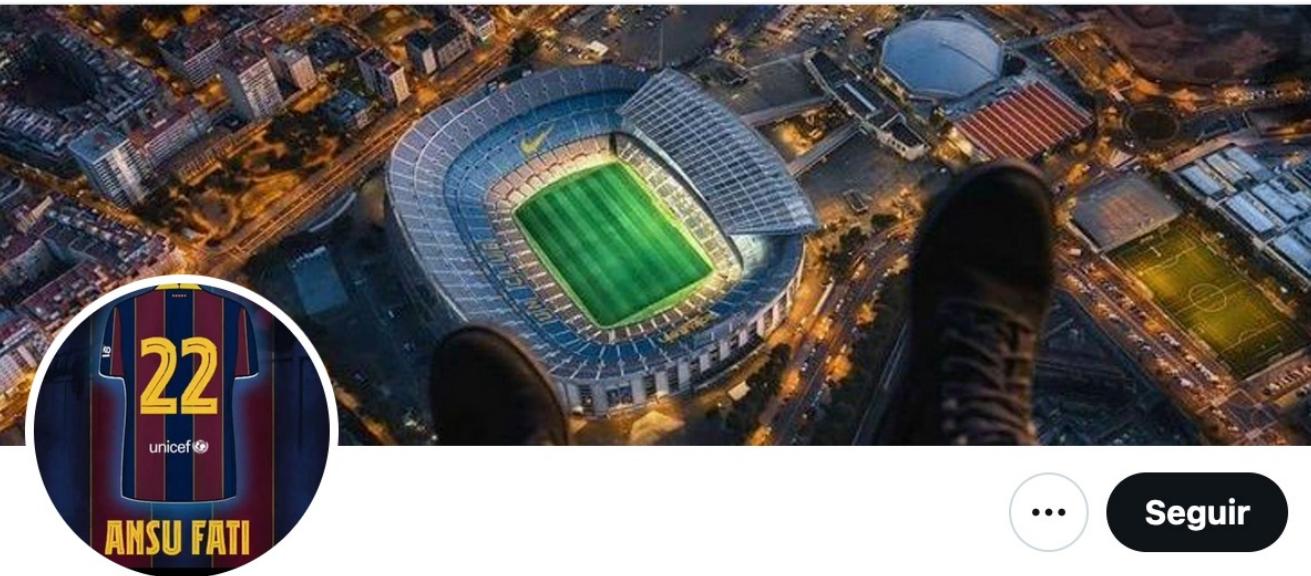
- En V(net) está la información relacionada al nodo:

```
> V(duque_net)$name[60000]  
[1] "MiguelS86480040"
```



el imparable

3.738 Tweets



...

Seguir

el imparable

@MiguelS86480040

visça el barça

◎ Bogotá, D.C., Colombia Se unió en marzo de 2018

994 Siguiendo **169** Seguidores

Ninguna de las cuentas que sigues sigue a este usuario

↑↓ el imparable lo retwitteó



Jenny from the block ❤️ @eldeltapabocas · 7 sept.

...

Bancolombia me abudineó una plata. Salí a la calle y me carrasquillearon el celular. No sé si es porque me están viendo cara de Duque o qué pero ésta ciudad está llena de Álvaro Uribe...

92

↑↓ 1,6 mil

❤️ 5 mil



V(NET)

- Este es un poco más complicado. Recuerden que tengo que empatar la información de los nodos (que están repetidos y en diferentes columnas de mi base de datos original), con la información de los nodos como la organizar *igraph*.
- Para hacer esto se necesita un poco de magia...

```
> # Como los nodos están en un orden distinto en nuestro objeto igraph que en nuestro
> # df de tuits, hay que hacer un poco de magia:
>
> nodos <- as.data.frame(V(duque_net)$name) # Me quedo solo con los nodos (y mantengo el orden)
> colnames(nodos) <- "nodos"
>
> duque_auto <- duque_rts %>% # Ahora solo veo autoridades y la info que quiero
+   distinct(retweet_screen_name,.keep_all=T) %>%
+   select(retweet_screen_name,retweet_followers_count,retweet_friends_count,retweet_verified)
distinct: removed 244,489 rows (97%), 7,993 rows remaining
select: dropped 31 variables (user_id, status_id, created_at, screen_name, text, ...)
> colnames(duque_auto) <- c("nodos","followers","friends","verified")
>
> duque_hub <- duque_rts %>% # y luego solos hubs y la info que quiero
+   distinct(screen_name,.keep_all=T) %>%
+   select(screen_name,followers_count,friends_count,verified)
distinct: removed 174,721 rows (69%), 77,761 rows remaining
select: dropped 31 variables (user_id, status_id, created_at, text, source, ...)
> colnames(duque_hub) <- c("nodos","followers","friends","verified")
>
> duque_nodos <- rbind.data.frame(duque_auto,duque_hub) # Los uno
> duque_nodos <- duque_nodos[!duplicated(duque_nodos$nodos),] # y borro duplicados
>
> length(nodos$nodos) == length(duque_nodos$nodos) # Mismo número!
[1] TRUE
```

```
> ## Ahora, para mantener el orden de los nodos, utilizo join y
> ## ¡¡SIEMPRE PONGO A MI DF NODOS PRIMERO!!
>
> nodos <- left_join(nodos,duque_nodos)
Joining, by = "nodos"
left_join: added 3 columns (followers, friends, verified)
  > rows only in x      0
  > rows only in y  (  0)
  > matched rows     82,817
  >                   =====
  > rows total        82,817
>
> nodos$nodos[23123] == V(duque_net)$name[23123] # Mismo nombre!
[1] TRUE
```

- Es importantísimo verificar que el orden de nuestra información de los nodos coincide con la información de nuestro objeto *igraph*.

```
> ## Recien ahora podemos agregar info a nuestro nodos:  
>  
> V(duque_net)$followers <- nodos$followers  
> V(duque_net)$friend <- nodos$friends  
> V(duque_net)$verified <- nodos$verified
```

- ¡Listo!
- No hay muchas más información que agregar. El resto de información importante acerca de los nodos la vamos a sacar de la propia red.
- Y para eso debemos analizar las características de la red...

¿PREGUNTAS?

Pregunta de Investigación

- No lo vamos a cubrir
- ¡Importantísimo!
- Nuestra ventaja comparativa

Data

- Obtener (TW)
- Limpieza de datos
- Organizar la data
- Explorar la data

Crear la red

- Introducción a igraph
- V(net) y E(net) en igraph

Características de la red

- Betweenness
- Centrality
- Paths

Detección de comunidades

- Greedy algorithms / Randomwalk
- Características de las comunidades.

Visualizar la red

- Layout
 - Force-directed, random.
- Centro / Periferia
- Baja y alta resolución

Análisis de la red I

- Activación de la red
- La paradoja de la amistad
- Rising stars / rising tides

Análisis de la red II

- Disonancias y resonancias cognitivas
- Viralidad
- Análisis de texto
- Experimentos