GDAT622\_Invest6\_Twitter\_James

James Stanfield

7/3/2020

# Access the Twitter API from inside Rstudio

Thankfully, I still had my old Twitter api setup saved on Github. I saved them into the new project folder and then removed them, as this is going into a public repository.

#app.name <- "app name"  
#consumer.key <- "consumer key"  
#consumer.secret <- "consumer secret"

#saveRDS(app.name, file = "app\_name.Rds")  
#saveRDS(consumer.key, file = "consumer\_key.Rds")  
#saveRDS(consumer.secret, file = "consumer\_secret.Rds")

Now we can create and use our token.

#token <- create\_token(app = app.name, consumer\_key = consumer.key, consumer\_secret = consumer.secret)

# Paradox Interactive

Paradox Interactive is one of my favorite video game companies. Crusader Kings II, a game all about building and maintaining a medieval dynasty has earned many hours of my time. Funnily enough, success in the game comes from how well you build an internal network of servants and vassals and an external network of leiges and allies. If both are built well, few things will shake you. Build one well and the other poorly, and you will deal with seasonal trouble. Build both poorly, and you will not rule for long.

Given this connection to our topic for this course, Paradox seemed like a fitting choice for this investigation.

#get history of tweets  
tl\_pi <- get\_timeline(c("PdxInteractive"), n = 400)  
  
# There are 900 columns, I'm only going to keep 2  
tl\_pi %>%  
 select(., screen\_name, reply\_to\_screen\_name) %>%   
 na.omit(.) %>%  
 distinct(.) -> tl\_pi  
  
str(tl\_pi)

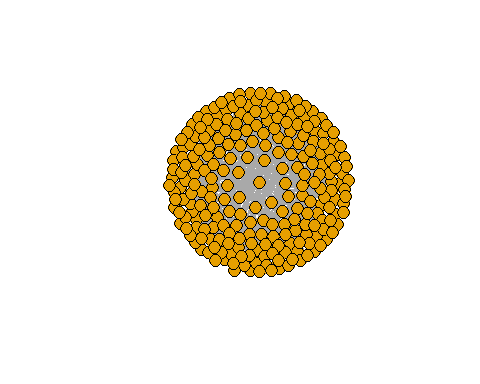
## Registered S3 method overwritten by 'cli':  
## method from   
## print.boxx spatstat

## tibble [241 x 2] (S3: tbl\_df/tbl/data.frame)  
## $ screen\_name : chr [1:241] "PdxInteractive" "PdxInteractive" "PdxInteractive" "PdxInteractive" ...  
## $ reply\_to\_screen\_name: chr [1:241] "LukasRoper" "Xales67" "EpicHoi4" "vieiraisanerd" ...

Well, that looks just like an edglist made out of Paradox Interactive’s reponses to tweets. Since these are responses, we can treat this as undirected, as each edge represents a tweet sent and reciprocated.

Now if we plotted this, we would just get Paradox Interactive connected to a bunch of otherwise independent nodes.

graph\_from\_data\_frame(tl\_pi) -> test1\_igraph  
plot(test1\_igraph, vertex.label = NA)



Yep, just a tennis ball with Paradox in the middle.

get\_friends("PdxInteractive")

## # A tibble: 512 x 2  
## user user\_id   
## <chr> <chr>   
## 1 PdxInteractive 110507842   
## 2 PdxInteractive 216368886   
## 3 PdxInteractive 1112988438343663616  
## 4 PdxInteractive 94226233   
## 5 PdxInteractive 1244939733475737600  
## 6 PdxInteractive 738246814193864704   
## 7 PdxInteractive 821726553797173248   
## 8 PdxInteractive 31746718   
## 9 PdxInteractive 320860323   
## 10 PdxInteractive 3299971   
## # ... with 502 more rows

get\_friends("16370407")

## # A tibble: 445 x 2  
## user user\_id   
## <chr> <chr>   
## 1 16370407 246717186   
## 2 16370407 1565575548  
## 3 16370407 2987541023  
## 4 16370407 2865669239  
## 5 16370407 388858301   
## 6 16370407 117714729   
## 7 16370407 40196571   
## 8 16370407 70925902   
## 9 16370407 19255177   
## 10 16370407 49616273   
## # ... with 435 more rows

#List who PdxInteractive follows  
get\_friends("PdxInteractive") -> f1  
  
#Pick 3 of those user and list who they follow  
get\_friends("16370407") -> f2  
get\_friends("216368886") -> f3  
get\_friends("1112988438343663616") -> f4  
  
#stack all the lists  
bind\_rows(f1, f2, f3, f4) -> friend\_list  
  
str(friend\_list)

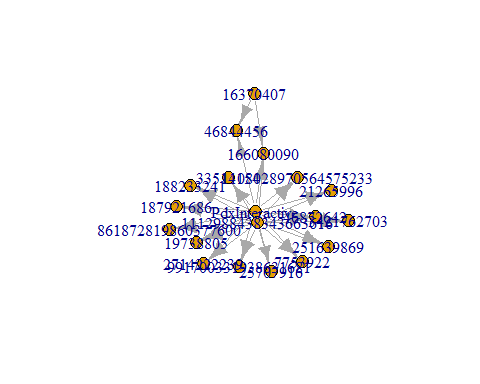
## tibble [1,027 x 2] (S3: tbl\_df/tbl/data.frame)  
## $ user : chr [1:1027] "PdxInteractive" "PdxInteractive" "PdxInteractive" "PdxInteractive" ...  
## $ user\_id: chr [1:1027] "110507842" "216368886" "1112988438343663616" "94226233" ...

#Remove anyone who doesn't appear at least twice in user\_id  
newd <- friend\_list %>% group\_by(user\_id) %>% dplyr::filter(n()>1) %>% arrange(., user\_id)  
newd

## # A tibble: 33 x 2  
## # Groups: user\_id [16]  
## user user\_id   
## <chr> <chr>   
## 1 PdxInteractive 1108028970564575233  
## 2 1112988438343663616 1108028970564575233  
## 3 PdxInteractive 166080090   
## 4 16370407 166080090   
## 5 1112988438343663616 166080090   
## 6 PdxInteractive 187921686   
## 7 1112988438343663616 187921686   
## 8 PdxInteractive 188233241   
## 9 1112988438343663616 188233241   
## 10 PdxInteractive 19738805   
## # ... with 23 more rows

This leaves us with 33 edges, much more manageable

graph\_from\_data\_frame(newd) -> Pdx\_igraph  
plot(Pdx\_igraph)



lookup\_users("1108028970564575233") -> u1  
u1$screen\_name

## [1] "VtM\_Bloodlines"

Interesting, it looks like the twitter account for “Vampire the Masquerade: Bloodlines” connects to a lot of the same accounts as PAradox Interactive does.

lookup\_users("166080090") -> u2  
u2$screen\_name

## [1] "TheWesterFront"

TheWesterFront happens to be “Executive Chairman of the Board at Paradox Interactive. Ventrue Clan.” (Ventrue is a Vampire the Masquerade reference)

lookup\_users("16370407") -> u3  
u3$screen\_name

## [1] "MrValue"

Mattias Vahlne aka @MrValue is an “Internet professional in general, Internet Video, Strategy, procurement, bizdev, CRM, Marketing Automation.”

So someone who works professionally in the kinds of fields that would interest Paradox as a business.

get\_timeline(c("PdxInteractive"), n = 400)

## # A tibble: 400 x 90  
## user\_id status\_id created\_at screen\_name text source  
## <chr> <chr> <dttm> <chr> <chr> <chr>   
## 1 193935~ 12795344~ 2020-07-04 21:56:02 PdxInterac~ "He ~ Twitt~  
## 2 193935~ 12791638~ 2020-07-03 21:23:11 PdxInterac~ "@Lu~ Twitt~  
## 3 193935~ 12790068~ 2020-07-03 10:59:40 PdxInterac~ "We'~ Twitt~  
## 4 193935~ 12790055~ 2020-07-03 10:54:31 PdxInterac~ "@Xa~ Twitt~  
## 5 193935~ 12789745~ 2020-07-03 08:51:15 PdxInterac~ "@Ep~ Twitt~  
## 6 193935~ 12789709~ 2020-07-03 08:36:53 PdxInterac~ "Our~ Twitt~  
## 7 193935~ 12789639~ 2020-07-03 08:09:10 PdxInterac~ "@vi~ Twitt~  
## 8 193935~ 12787078~ 2020-07-02 15:11:30 PdxInterac~ "\U0~ Twitt~  
## 9 193935~ 12787024~ 2020-07-02 14:50:04 PdxInterac~ "@ya~ Twitt~  
## 10 193935~ 12786992~ 2020-07-02 14:37:05 PdxInterac~ "@mi~ Twitt~  
## # ... with 390 more rows, and 84 more variables: display\_text\_width <dbl>,  
## # reply\_to\_status\_id <chr>, reply\_to\_user\_id <chr>,  
## # reply\_to\_screen\_name <chr>, is\_quote <lgl>, is\_retweet <lgl>,  
## # favorite\_count <int>, retweet\_count <int>, quote\_count <int>,  
## # reply\_count <int>, hashtags <list>, symbols <list>, urls\_url <list>,  
## # urls\_t.co <list>, urls\_expanded\_url <list>, media\_url <list>,  
## # media\_t.co <list>, media\_expanded\_url <list>, media\_type <list>,  
## # ext\_media\_url <list>, ext\_media\_t.co <list>,  
## # ext\_media\_expanded\_url <list>, ext\_media\_type <chr>,  
## # mentions\_user\_id <list>, mentions\_screen\_name <list>, lang <chr>,  
## # quoted\_status\_id <chr>, quoted\_text <chr>, quoted\_created\_at <dttm>,  
## # quoted\_source <chr>, quoted\_favorite\_count <int>,  
## # quoted\_retweet\_count <int>, quoted\_user\_id <chr>,  
## # quoted\_screen\_name <chr>, quoted\_name <chr>,  
## # quoted\_followers\_count <int>, quoted\_friends\_count <int>,  
## # quoted\_statuses\_count <int>, quoted\_location <chr>,  
## # quoted\_description <chr>, quoted\_verified <lgl>,  
## # retweet\_status\_id <chr>, retweet\_text <chr>,  
## # retweet\_created\_at <dttm>, retweet\_source <chr>,  
## # retweet\_favorite\_count <int>, retweet\_retweet\_count <int>,  
## # retweet\_user\_id <chr>, retweet\_screen\_name <chr>, retweet\_name <chr>,  
## # retweet\_followers\_count <int>, retweet\_friends\_count <int>,  
## # retweet\_statuses\_count <int>, retweet\_location <chr>,  
## # retweet\_description <chr>, retweet\_verified <lgl>, place\_url <chr>,  
## # place\_name <chr>, place\_full\_name <chr>, place\_type <chr>,  
## # country <chr>, country\_code <chr>, geo\_coords <list>,  
## # coords\_coords <list>, bbox\_coords <list>, status\_url <chr>,  
## # name <chr>, location <chr>, description <chr>, url <chr>,  
## # protected <lgl>, followers\_count <int>, friends\_count <int>,  
## # listed\_count <int>, statuses\_count <int>, favourites\_count <int>,  
## # account\_created\_at <dttm>, verified <lgl>, profile\_url <chr>,  
## # profile\_expanded\_url <chr>, account\_lang <lgl>,  
## # profile\_banner\_url <chr>, profile\_background\_url <chr>,  
## # profile\_image\_url <chr>

Let’s try a different approach.

I’ll make an edgelist by connecting everyone who appeared in the same tweet together, like a co-occurence.

# get history of tweets  
paradox\_el <- get\_timeline(c("PdxInteractive"), n = 400)  
  
# Let's grab our very 1st edgelist again  
paradox\_el %>%  
 select(., screen\_name, reply\_to\_screen\_name) %>%   
 na.omit(.) %>%  
 distinct(.) -> p\_el1  
names(p\_el1) <- c("V1", "V2")  
  
# Now we'll connect reply\_to\_screen\_name to quoted\_screen\_name  
paradox\_el %>%  
 select(., reply\_to\_screen\_name, quoted\_screen\_name) %>%   
 na.omit(.) %>%  
 distinct(.) -> p\_el2  
names(p\_el2) <- c("V1", "V2")  
  
# Now we'll connect quoted\_screen\_name to retweet\_screen\_name  
paradox\_el %>%  
 select(., quoted\_screen\_name, retweet\_screen\_name) %>%   
 na.omit(.) %>%  
 distinct(.) -> p\_el3  
names(p\_el3) <- c("V1", "V2")  
  
# Now we'll connect screen\_name to retweet\_screen\_name  
paradox\_el %>%  
 select(., screen\_name, retweet\_screen\_name) %>%   
 na.omit(.) %>%  
 distinct(.) -> p\_el4  
names(p\_el4) <- c("V1", "V2")  
  
# Now we'll connect reply\_to\_screen\_name to retweet\_screen\_name  
paradox\_el %>%  
 select(., reply\_to\_screen\_name, retweet\_screen\_name) %>%   
 na.omit(.) %>%  
 distinct(.) -> p\_el5  
names(p\_el5) <- c("V1", "V2")  
  
# Now we'll connect screen\_name to quoted\_screen\_name  
paradox\_el %>%  
 select(., screen\_name, quoted\_screen\_name) %>%   
 na.omit(.) %>%  
 distinct(.) -> p\_el6  
names(p\_el6) <- c("V1", "V2")  
  
#stack all the lists  
bind\_rows(p\_el1, p\_el2, p\_el3, p\_el4, p\_el5, p\_el6) %>% distinct(.)-> co\_el  
  
head(co\_el)

## # A tibble: 6 x 2  
## V1 V2   
## <chr> <chr>   
## 1 PdxInteractive LukasRoper   
## 2 PdxInteractive Xales67   
## 3 PdxInteractive EpicHoi4   
## 4 PdxInteractive vieiraisanerd  
## 5 PdxInteractive yaboipuncake   
## 6 PdxInteractive mitchgit

Well, turns out that in the 400 tweets we pulled, there were no co-occurences

At this point, I think I’m going to leave my network-building with just the one I made using friends.

Using followers might have provided more value, but would have been much more time intensive. The value we did get, is insight into who Paradox wants to reach, or at least who the social media manager and his bosses wnat to reach.

If I was Paradox, I would likely study who follows the accounts that I want to reach and see how I can best interact those communities.

The importance of a follower vs a friend is that, in theory, a follower is already predisposed to hear what you have to say. By choosing to follow you, they have signed up to receive anything you post. Friends are those you chosen to follow, and you have much less built-in influence on them, if you have any at all.

Retweets and hashtags are powerful as they involve jumping onto something that is already moving. If someone expresses something that you want to spread, retweeting it can expand the reach of the original author to your audience. In reverse, you hope that your followers will retweet your posts so that they spread to their followers and so on throughout the network.

Hashtags posses the same power, allowing accounts to quickly call out a specific topic with a quick string, and spread that topic.

The data that could be valuable in this kind of marketing is the buying habits of your target audience. Companies like Amazon and Google make a great deal of their money by selling customer data to interested parties.