Récupérer automatiquement des coordonnées en R à partir de nom de lieux

Julie Gravier

27/01/2020

Que ce soit pour une présentation dans un colloque, la réalisation d’un cours ou plus largement dans un projet de reprise de données existantes, nous avons souvent le souhait de cartographier la répartition des objets archéologiques – au sens large – mis au jour et analysés. Toutefois, il est assez fréquent que la documentation à partir de laquelle nous puissions créer ces cartes soit présentée dans le cadre de publications sous forme de tableaux ou de listes énumératives, sans précision des latitudes et longitudes des objets étudiés. Leurs géolocalisations est généralement faite manuellement, point par point, lieu par lieu, dans un SIG. Le problème principal de cette procédure tient au fait que sa probabilité de réalisation est quasi inversement proportionnelle au volume de données à traiter et au temps que l’on détient… La boîte à outils a pour objectif de présenter comment peut-on récupérer des coordonnées de manière automatique afin de réduire le temps et l’ingratitude de ce travail.

### Packages nécessaires

library(tidyverse)  
# devtools::install\_github(repo = 'rCarto/photon') # package développé pour faire le lien entre R et l'API Photon  
library(photon)  
library(sf) # package spatial  
library(tmap) # package de visualisation spatiale dynamique

## Importer et adapter les données

À partir du bilan sur la Protohistoire récente en Picardie publié par François Malrain, Stéphane Godefroy et Frédéric Gransar et du tableau recensant les études carpologiques réalisées, nous allons récupérer les coordonnées des lieux d’où proviennent les échantillonnages caropologiques. Plus précisément, le tableau de la publication recense les auteur.e.s ayant réalisé l’étude, le département d’appartenance du site archéologique, la commune, le nom du site, le.s responsable.s de l’opération archéo, la datation associée à l’échantillon caropologique. Hélas, compte tenu de la manière dont a été encodé le .pdf, il est relativement peu aisé d’exploiter le .pdf via R. Il est ainsi plus rapide de copier-coller préalablement le tableau dans un .txt (intitulé “RAP.txt” dans le dossier “data”).

rap\_picardie <- read\_table("data/RAP.txt") # lire le tableau de données  
head(rap\_picardie)

## # A tibble: 6 x 1  
## `Étude par Dépt. Commune Site Responsable Datation`   
## <chr>   
## 1 Matterne Aisne Bazoches-sur-Vesle La Foulerie S. Dessenne La Tène D   
## 2 Matterne Aisne Bazoches-sur-Vesle Les Chantraines Cl. Pommepuy La Tène  
## 3 Matterne Aisne Beaurieux Les Grèves C. Haselgrove toutes périodes   
## 4 Bakels Aisne Berry-au-Bac Le Chemin de la pêcherie Néolithique   
## 5 Matterne Aisne Bohain Le Moulin Mayeux Ph. Lemaire Ier siècle   
## 6 Matterne Aisne Bucy-Le-Long Le Grand Marais Cl. Pommepuy Hallstatt

Le tableau contient une seule colonne de type *character*. On va ainsi séparer les éléments en différentes colonnes, sachant que l’on souhaite récupérer les informations sur la commune (et le département) dans laquelle a eu lieu l’opération archéologique.

rap\_picardie\_df <- rap\_picardie %>%  
 rename(variable = `Étude par Dépt. Commune Site Responsable Datation`) %>%  
 separate(col = variable, c("Etude", "Departement", "Commune"), sep = " ", remove = FALSE) %>%  
 # séparation des premiers éléments en 3 colonnes afin de récupérer le nom des communes et des départements  
 mutate(hdf = "Hauts-de-France, France", # création d'une colonne avec informations complémentaires spatiales  
 lieux = str\_c(Commune, Departement, hdf, sep = ", ")) # concaténation des colonnes liées aux noms des communes, dép, région et pays

Deux choses sont importantes à noter dans le code ci-dessus :  
1. la fontion *rename()* n’est pas nécessaire ici. Elle permet uniquement d’avoir un nom de variable court dans la 2e ligne de code pour que ce dernier soit plus lisible lorsque l’on travaille sur petit écran ;  
2. nous avons créé une colonne **lieux** contenant le nom de la commune, le département, la région actuelle (Hauts-de-France) et le pays (France). C’est cette colonne qui nous servira à faire le géocodage des lieux d’où proviennent les échantillons caropologiques. L’ajout d’information (région & pays) sert en tant que contrainte spatiale dans la recherche des cordonnées lors de l’utilisation de l’API Photon. En effet, par exemple, plusieurs communes ont le même nom en France. Sans contrainte de recherche dans la région Hauts-de-France, il est possible que le résultat de la recherche automatique soit relatif à une commune située en dehors de la région.

## Géocodage et vérification de sa validité

rap\_geoloca <- geocode(location = rap\_picardie\_df$lieux, key = "place", limit = 1) # récupération des coordonnées avec photon  
rap\_geoloca\_sf <- st\_as\_sf(x = rap\_geoloca, coords = c("lon", "lat")) %>% # construction d'un objet "sf"  
 st\_set\_crs(4326) %>% # les coordonnées récupérées sont el lon/lat, WGS 84  
 st\_transform(crs = 2154) # transformation du système de coordonnées en Lambert 93

Une fois que l’on a récupérer les coordonnées et créé un objet *sf*, soit un objet spatial dans R (voir en particulier Pebesma, E., 2018. Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10 (1), 439-446, <https://doi.org/10.32614/RJ-2018-009>), on va pouvoir vérifier la validité des coordonnées de manière dynamique avec *tmap*

## tmap\_mode("view") # mode dynamique  
## ici le mode dynamique n'est pas actionné du fait que l'on écrit un Rmarkdown  
tm\_shape(rap\_geoloca\_sf) + # le "shape" (objet sf)  
 tm\_bubbles(size = 0.4, col = "darkorange") # type de visualisation

Si l’on est satisfait, on peut exporter l’objet *sf* en .shp :

st\_write(rap\_geoloca\_sf, "data/geoloca\_etudes\_carpo\_RAP25.shp")

## Writing layer `geoloca\_etudes\_carpo\_RAP25' to data source `data/geoloca\_etudes\_carpo\_RAP25.shp' using driver `ESRI Shapefile'  
## features: 88  
## fields: 13  
## geometry type: Point

## Cartographie

Enfin, on peut cartographier les résultats pour notre futur conférence, cours, publication, etc.

#### Imports des fonds de carte

sf\_fond\_europe <- read\_sf("data/geom\_eu34\_nuts13.shp") %>% # fond Europe  
 st\_transform(crs = 2154) # en Lambert 93  
france <- read\_sf("data/REGION.shp") %>% # fond des régions françaises actuelles  
 st\_transform(crs = 2154) # en Lambert 93  
  
## zoom sur notre zone d'étude  
bbox\_haut\_de\_france <- st\_bbox(france %>% filter(INSEE\_REG == "32")) # 32 étant le code des Hauts-de-France

#### Carte

ggplot() +   
 geom\_sf(data = sf\_fond\_europe, fill = "grey98", color = "grey98") +  
 geom\_sf(data = france, fill = "grey98", color = "grey40") +  
 geom\_sf(data = rap\_geoloca\_sf, color = "grey30", fill = "#e66101", shape = 21, size = 2) +  
 coord\_sf(xlim = bbox\_haut\_de\_france[c(1,3)], ylim = bbox\_haut\_de\_france[c(2,4)]) +  
 ggthemes::theme\_igray() +  
 labs(subtitle = "Les études carpologiques réalisées en Picardie en 2005",  
 caption = "J. Gravier 2020 | LabEx DynamiTe, UMR Géographie-cités.\nSources : Malrain, Gaudefroy, Gransar 2005") +  
 theme(axis.text.x = element\_blank(),  
 axis.text.y = element\_blank(),  
 axis.ticks = element\_blank()) +  
 ggspatial::annotation\_scale(location = "tr", width\_hint = 0.5)

