Ajouter des données attributaires dans un fichier spatial

Traitement avec R

# Les problématiques

## L’ID des polygones dans les objets spatiaux

**Warning**

**L’ID des polygones correspond aux numéros de ligne de la table attributaire**

## Absence de clés de jointure entre Open Data et les données spatiales : Comment procéder ?

Passer par une table tierce permettant la jointure : typiquement, il peut s’agir d’une table ID1 / ID2 qui permettra de faire le lien entre les Open Data et les données spatiales.

# La parfaite illustrations de ces problématiques avec le cas Pologne

## Source des données

* Fond de carte téléchargé sur le site : [Lien](http://codgik.gov.pl/index.php/darmowe-dane/prg.html)
* Données Open Data téléchargées sur le site : [Lien](http://swaid.stat.gov.pl/en/Edukacja_dashboards/Raporty_predefiniowane/RAP_DBD_EDU_12.aspx)

## Les grandes étapes

Après avoir importé les fonds de carte et les données Open Data (en ayant éventuellement et si nécessaire retraités), on peut associer ces dernières à l’objet spatial.

Pour ce faire, il « suffit » de joindre les données attributaire du SpatialPolygonsDataFrame avec les données Open Data.

L’idée est la suivante :

* On « sort » le dataframe contenu dans l’objet spatial
* On effectue tous les retraitements souhaités (dont la jointure fait partie)
* On réintègre le nouveau dataframe dans l’objet spatial.

## Explication du code

Dans le cas de la Pologne, on ne dispose pas de clé de jointure entre l’Open Data téléchargé et le fonds de carte. On a donc crée un petit fichier qui attribue à chaque Voivodeship le numéro correspondant.

### Importation de la carte

**# Importation de la carte au niveau "wojewodztwa"**

wojewodztwa <- readOGR(dsn = root\_carte, layer = "wojewodztwa",

verbose = F,stringsAsFactors=FALSE)

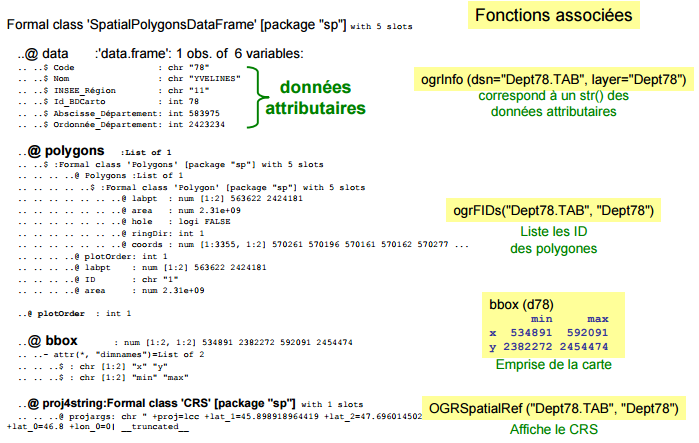
> slotNames(wojewodztwa)

[1] "data" "polygons" "plotOrder" "bbox" "proj4string"

Un SpatialPolygonsDataFrame est compose des classes suivantes :

|  |  |  |
| --- | --- | --- |
| Slots | Classe | Descriptions |
| Data | « data.frame » | Les données attributaires |
| Polygons | « list » | ***Voir ci dessous*** |
| PlotOrder | « integer » | Identifie l’ordre dans lequel les différents polygones doivent etre affiché |
| Bbox | « matrix » | Renvoie les coordonnées du rectangle contenant l’ensemble des données (autrement dit, le « x » min/max et le « y » min/max) |
| Proj4string | « CRS » | Définis le système de projection |

**Visualisation des slots d’un SpatialPolygonDataFrame :**



Source : [Lien](http://rug.mnhn.fr/semin-r/PDF/semin-R_carto_RBaudoin_170311.pdf)

**Structure du slot « polygons » : un polygone contient lui-même différent slot :**

# Slot du premier polygone

> slotNames(wojewodztwa@polygons[[1]])

[1] "Polygons" "plotOrder" "labpt" "ID" "area"

|  |  |  |
| --- | --- | --- |
| Slots | Classe | Descriptions |
| Polygons | List | Contient :   * ***Labpt*** : Paire de coordonnées x, y donnant un point d'étiquette, le point d'étiquette de la plus grande composante polygone * ***Area*** : Object of class "logical"; does the polygon seem to be a hole * ***Hole*** : Object of class "logical"; does the polygon seem to be a hole * ***ringDir*** : Object of class "integer"; the ring direction of the ring (polygon) coordinates, holes are expected to be anti-clockwise * ***coords*** : les coordonnées du polygones, c'est-à-dire un vecteur (plus ou moins grand) qui permet de dessiner le contour de la figure géométrique |
| plotOrder | Integer | Identifie l’ordre dans lequel les différents polygones doivent être affiché |
| Labpt | Numeric | Paire de coordonnées x, y donnant un point d'étiquette, le point d'étiquette de la plus grande composante polygone |
| ID | Character | Le numéro identifiant du polygone |
| Area | Numeric | L’aire du polygone |

Affichage des types :

> class(wojewodztwa@polygons[[1]]@Polygons) # list

[1] "list"

> class(wojewodztwa@polygons[[1]]@plotOrder) # integer

[1] "integer"

> class(wojewodztwa@polygons[[1]]@labpt) # numeric

[1] "numeric"

> class(wojewodztwa@polygons[[1]]@ID) # character

[1] "character"

> class(wojewodztwa@polygons[[1]]@area) # numeric

[1] "numeric"

Affichage Contenu du slot « Polygons »

> str(wojewodztwa@polygons[[1]]@Polygons)

List of 1

$ :Formal class 'Polygon' [package "sp"] with 5 slots

.. ..@ labpt : num [1:2] 422175 309358

.. ..@ area : num 9.4e+09

.. ..@ hole : logi FALSE

.. ..@ ringDir: int 1

.. ..@ coords : num [1:23869, 1:2] 351531 351561 351581 351612 351639 ...

Après importation, on regarde attentivement la table attributaire qui dans ce cas précis correspond au tableau ci-dessous :

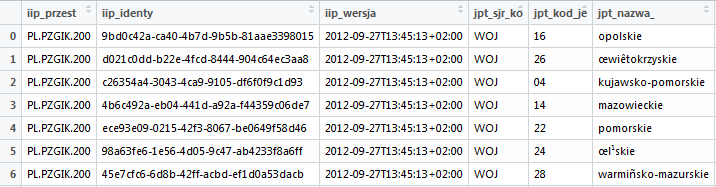


Tableau : Tableau attributaire des cartes Shapefile Pologne

L’objectif est d’ajouter les données suivantes :

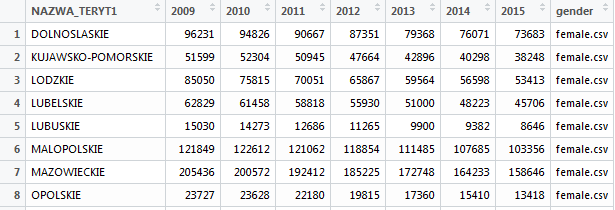


Tableau : Open Data Pologne à cartographier

Problème de clé de jointure, donc on va se servir d’une table (construite sur Excel, a partir des informations trouvée sur Wikipedia) :



Tableau : Table permettant de réaliser la jointure entre le fonds de carte et les Open Data Pologne

### Ajout de données attributaires

Les étapes sont les suivantes :

1. Ajouter les codes des voivodeships dans la base Open Data
2. Ajouter les codes des voivodeships dans l’objet spatial
3. Fusionner la base Open Data avec le dataframe spatial
4. Réintégrer les nouvelles données dans l’objet spatial

Le code :

**# Etape 0 : Récupération des ID des polygones (qui correspondent aux noms des lignes) et enregistrement dans le dataframe**

# Creation d’une variable « POLYID »

wojewodztwaWGS@data$POLYID <- rownames(wojewodztwaWGS@data)

# Affichage pour verification

View(wojewodztwaWGS@data)

**# Etape 1 : Ajout des code\_voivod dans le dataframe des donnees Open Data**

m1a <- merge(res, # Correspond aux "x"

code\_voivod, # Correspond aux "y"

by.x = "NAZWA\_TERYT1",

by.y = "Polish.name")

**# Etape 2 : Ajout des code\_voivod dans le dataframe des donnees spatiales**

**# Méthode 1 : en « sortant » le dataframe de l’objet spatial**

m1b <- merge(wojewodztwaWGS@data, # Correspond aux "x"

code\_voivod, # Correspond aux "y"

by.x = "jpt\_kod\_je",

by.y = "Teryt.")

**# Méthode 2 : cette étape peut etre « optimisée » en procédant de la manière suivante :**

library(dplyr) # Bibliothèque utile pour réaliser la jointure

*# Rappel : les jointures ne peuvent s'effectuer que si les variables sont de même format*

class(wojewodztwaWGS@data$jpt\_kod\_je) # Character

class(code\_voivod$Teryt.) # Factor

*# Modification du format de la variable Teryt.*

code\_voivod$Teryt. <- as.character(code\_voivod$Teryt.)

# On remplace le slot @data par un nouveau tableau de données, contenant les anciennes données + les nouvelles

wojewodztwaWGS@data <- left\_join(wojewodztwaWGS@data, code\_voivod, by = c("jpt\_kod\_je" = "Teryt."))

***# Remarque : le risque avec la méthode 2 c’est lorsque l’on a de grande base de données,***

***# avec beaucoup de retraitement à effectuer. Mais pour de petit retraitement, elle est plus***

***# avantageuse. Il faut donc savoir faire les deux pour pouvoir s’adapter facilement aux***

***# problématiques rencontrées.***

**# Etape 3 (facultative) : Fusionner la base Open Data avec le dataframe spatial**

***# Remarque : cette étape n’est pas à faire si l’on a choisi la seconde méthode lors de l’étape précédente***

m2 <- merge(m1a, # Correspond aux "x"

m1b, # Correspond aux "y"

by.x = "Teryt.",

by.y = "jpt\_kod\_je")

**# Etape 4 : Ajout dans l'objet spatial**

# On réordonne les data en fonction du polyID

wojewodztwaWGS@data <- m2\_bis[order(as.numeric(m2\_bis$POLYID)),]

# Comme c'est le nom de la ligne qui fait le lien avec l'ID des polygones, on renomme les lignes

rownames(wojewodztwaWGS@data) <- wojewodztwaWGS@data$POLYID

# Check :

all(lapply(wojewodztwaWGS@polygons, slot, "ID") == rownames(wojewodztwaWGS@data)) # TRUE

# True