Récupération de fonds de carte

**Warning**

**Il est impératif de connaître le système de coordonnées utilisé pour le calcul des coordonnées géographiques, sans quoi la représentation sera fausse. Le plus souvent, c’est le système WGS84 qui sera utilisé. Si on souhaite fusionner ces données à un objet spatial, il faudra s’assurer que les systèmes de coordonnées et la projection est la même.**

**Voir la note XX**

# Qu’est-ce qu’un fonds de carte ?

## Définition

Un fond de carte permet de représenter graphiquement les caractéristiques d’une région géographique d’intérêt. Le fond de carte permet de “planter le décor” et d’apporter les éléments contextuels utiles. C’est sur ce fond de carte que vont venir se greffer nos données géolocalisées.

Il existe une quantité infinie de fonds de cartes différents, qui peuvent grossièrement être rangés dans deux catégories :

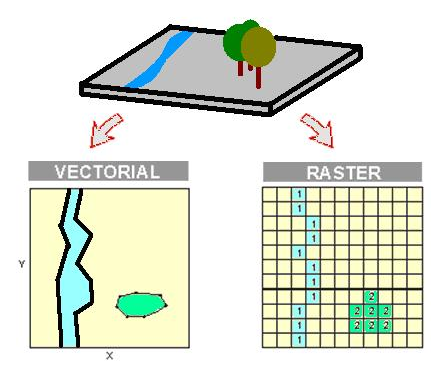


Figure 1 : A droite, une représentation matricielle de la zone. A gauche, une représentation vectorielle.

* les **représentations vectorielles (ou *vector layer*).**

Dans ce cas, les caractéristiques de la région d’intérêt sont représentées sous la forme d’un ensemble de formes (ou *features*) comme des points ou des tracés (lignes, polygones…). Ce format est particulièrement adapté à la représentation de zones géographiques bien délimitées, telles que les limites administratives (pays, régions, départements, communes), les limites géologiques, l’emplacement de villes, etc…

* les **représentations matricielles (ou *raster layer*).**

Dans ce cas, les caractéristiques de la région d’intérêt sont stockées sous la forme d’une image (ensemble de pixel rangés en lignes/colonnes, d’où le nom). A chaque pixel est associé une valeur, permettant de décrire les caractéristiques. Par exemple, l’altitude peut être représentée sous le forme de pixels allant du vert au rouge, les plans d’eau peuvent être représentés par des pixels de couleur bleu, etc…

## Le format « Shapefile »

Le shapefile, ou « fichier de formes » est un format de fichier pour les systèmes d'informations géographiques (SIG). Initialement développé par ESRI pour ses logiciels commerciaux, ce format est désormais devenu un standard de facto, et est utilisé par un grand nombre de logiciels.

Il contient toute l'information liée à la géométrie des objets décrits, qui peuvent être :

* des points
* des lignes
* des polygones

Son extension est classiquement SHP, et il est toujours accompagné de deux autres fichiers de même nom, et d'extensions :

* un fichier DBF, qui contient les données attributaires relatives aux objets contenus dans le shapefile
* un fichier SHX, qui stocke l'index de la géométrie

D'autres fichiers peuvent être également fournis :

* .sbn et .sbx - index spatial des formes.
* .fbn et .fbx - index spatial des formes pour les shapefile en lecture seule
* .ain et .aih - index des attributs des champs actifs dans une table ou dans une table d'attributs du thème.
* .prj - information sur le système de coordonnées, utilisant le format WKT (Well Known Text).
* .shp.xml - métadonnées du shapefile.
* .atx - fichier d'index des attributs pour le fichier dbf, sous la forme <shapefile>.<nom\_de\_la\_colonne>.atx (ArcGIS 8 et suivants)
* .qix

# Ou trouver des fonds de carte gratuitement ?

## Cas France : Les données IGN

Pour obtenir des fonds de cartes reprenant les limites des régions / départements / communes de France (sous forme de vector layer), le plus simple est d’utiliser les données GEOFLA mises à disposition par l’IGN. Ces données sont distribuées sont au format ESRI Shapefile (SHP), standard très largement répandu et quasi-universel. A noter que le système cartographique utilisé dans ces fichiers est le Lambert-93, le format légal en France.

## Cas France : Les données OSM

La plupart des jeux de données OSM directement réutilisables (fond de carte et cie) sont publiées sur le site de [l’Open Data](http://www.data.gouv.fr/). A titre d’exemple, les données de tracé administratifs issus d’OSM sont disponibles [ici](https://www.data.gouv.fr/fr/dataset/limites-administratives-francaises-issues-d-openstreetmap). Attention, les données issues d’OSM sont diffusées dans le système WGS84.

## Cas Monde : le site GADM

# Comment les intégrer dans R ?

Il existe plusieurs façons de charger des données spatiales dans R, faisant intervenir différents packages. Ici, nous utilserons uniquement les fonctions de chargement issus du packages rgdal. Ce package est une interface entre R et la “Geospatial Abstraction Library” (GDAL), une bibliothèque de fonctions dédiées aux manipulations de données spatiales très largement utilisée.

**Remarque** : *Les données utilisées ont été téléchargées depuis le site* [*http://professionnels.ign.fr/contoursiris*](http://professionnels.ign.fr/contoursiris) *depuis l'espace "telechargement". Sauvegarder ces donnees dans un répertoire, dont vous récupèrerez le chemin pour pouvoir l’enregistrer dans la variable « root » des exemples ci-dessous.*

## Ex1 : Avec la librairie MapTools

root <- votre\_repertoire

library(maptools)

IR1 <- readShapePoly(file.path(paste0(root,"/CONTOURS-IRIS\_FE.shp"))

,proj4string = CRS("+init=epsg:2154"))

**WARNING : quand on utilise ReadShapePoly, il faut préciser le CRS. En effet, quand on ne le renseigne pas (comme c’est le cas dans les lignes de commandes précédentes), alors on obtient les différences suivantes :**

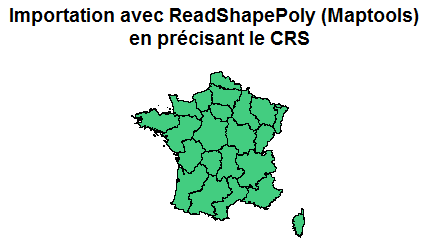


Figure 2 : Résultat d'importation avec MapTools (en précisant le CRS)

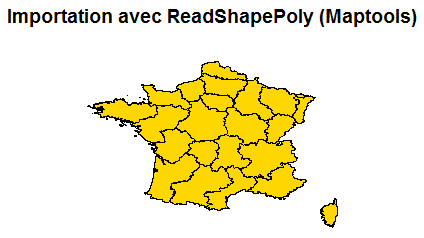


Figure 3 : Résultat d'importation avec MapTools (sans préciser le CRS)

L’avantage de Rgdal est que ce package reconnait automatiquement le CRS du fichier Shapefile importé. L’utilisation de Maptools peut donc s’avérer plus complexe car on a pas toujours connaissance du système de projection utilisé pour la carte téléchargée.

## Ex2 : Avec la librairie Rgdal

library(rgdal)

IR2 <- readOGR(dsn = root,

layer="CONTOURS-IRIS\_FE",

stringsAsFactors=FALSE)

# La structure des données spatiale

Les objets créés via *readOGR* (notamment) ont des structures particulières. Ils sont divisés en plusieurs sous-objets, ou slots, préfixé par le symbole @. Chaque sous-objet possède sa propre structure, et ses propres informations. Pour y accéder, la commande suit le même motif que pour accéder à une variable dans un dataframe, mais via le symbole @ au lieu du symbole $.

Exemple :

> slotNames(IR2)

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

# SI on souhaite accéder au slot « proj4string »

> IR1@proj4string

CRS arguments:

+init=epsg:2154 +proj=lcc +lat\_1=49 +lat\_2=44 +lat\_0=46.5 +lon\_0=3 +x\_0=700000

+y\_0=6600000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no\_defs

# Si on souhaite afficher quelques donnees

> head(IR1@data$DEPCOM)

[1] 95580 95258 95116 95308 95055 95395

Les informations de géométries, relatives au fichier .SHP, sont stockées dans les slots @polygons, @plotOrder et @bbox.

Les données associées, issues du fichier .DBF, sont dans le slot @data.

Enfin, les données de projection, issues du fichier .PROJ, sont dans un slot à part, @proj4string.

**Remarque :** *Pour plus d’informations, vous pouvez regarder la documentation accompagnant cette classe particulière avec la commande class?SpatialPolygons.*