

Création de fonds de cartes statiques et dynamiques avec R et Google Maps

Jérôme SUEUR



MNHN
UMR CNRS 7205 - OSEB

Mardi 7 février 2012

Plan

Pourquoi faire ?

Cartes statiques
RgoogleMaps
dismo

Cartes dynamiques
googleVis
plotGoogleMaps

Autres possibilités

Références

Plan

Pourquoi faire ?

Cartes statiques
RgoogleMaps
dismo

Cartes dynamiques
googleVis
plotGoogleMaps

Autres possibilités

Références

Pourquoi faire ?

- ▶ cartes de distribution d'espèces,
- ▶ cartes de densité par pays,
- ▶ suivi de déplacement (tracking, télédétection),
- ▶ cartes de territoires, de zones végétales, etc.

Plan

Pourquoi faire ?

Cartes statiques
RgoogleMaps
dismo

Cartes dynamiques
googleVis
plotGoogleMaps

Autres possibilités

Références

Cartes statiques

RgoogleMaps

- ▶ Auteur : Markus Loecher,
- ▶ Nécessite les packages : png, ReadImages,
- ▶ Interroge Google Maps et récupère les fonds de carte,
- ▶ Ajoute des points et lignes sur les fonds de carte.

Cartes statiques

RgoogleMaps

Coordonnées et noms des principaux sites du MNHN en IdF :

```
lat <- c("48.84471", "48.862", "48.83873",  
         "48.83248", "48.69335", "48.83047")  
long <- c("2.36", "2.2876", "2.3591",  
          "2.41724", "2.49626", "2.1117")  
nom <- c("Jardin des Plantes", "Musée de l'Homme",  
         "IPH", "Zoo de Vincennes",  
         "Brunoy", "Arboretum de Chèvreloup")
```

Cartes statiques

RgoogleMaps

Carte définie à partir d'un point central et d'un niveau de zoom, sauvee dans un fichier .png :

```
library(RgoogleMaps)

paris1 <- GetMap(center=c(48.857, 2.348),
                 zoom=12,
                 destfile="paris1.png",
                 mptype="terrain")
```


Cartes statiques

RgoogleMaps

Pour afficher la carte dans x11, utiliser `PlotOnStaticMap()` :

`PlotOnStaticMap(paris1)`

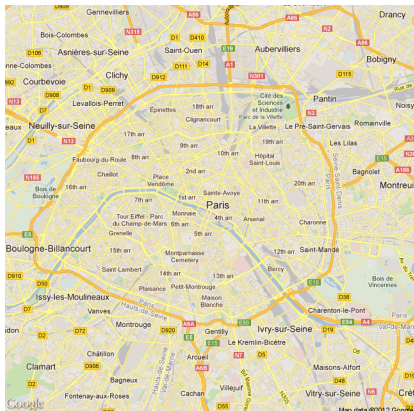


Cartes statiques

RgoogleMaps

La même carte avec un autre fond :

```
paris2 <- GetMap(center=c(48.857, 2.348), zoom=12,  
                 destfile="paris2.png", maptype="normal")  
PlotOnStaticMap(paris2)
```



Cartes statiques

RgoogleMaps

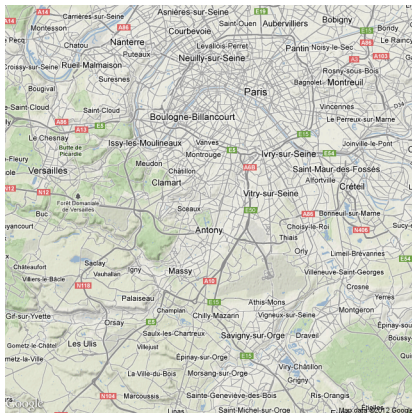
La fonction `qbbox()` [quantiles bounding box] permet de cadrer la carte en fonction des points à afficher :

```
bb <- qbbox(lat = as.numeric(lat),  
            lon = as.numeric(long))  
paris3 <- GetMap.bbox(latR = bb$latR,  
                      lonR = bb$lonR,  
                      destfile = "paris3.png",  
                      maptype="terrain"  
                      )
```

Cartes statiques

RgoogleMaps

PlotOnStaticMap(paris3)

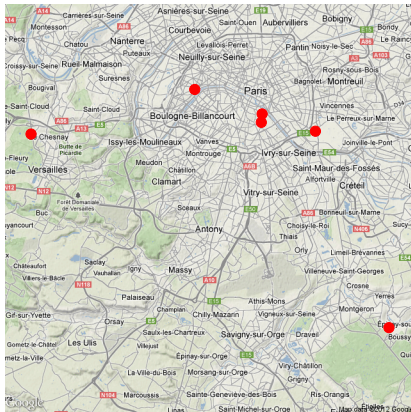


Cartes statiques

RgoogleMaps

Pour illustrer la carte avec des points ou des lignes, utiliser la fonction `PlotOnStaticMap()` :

```
PlotOnStaticMap(paris3,  
                 lat = as.numeric(lat), lon = as.numeric(long),  
                 cex = 2, pch = 19, col = "red", add = FALSE)
```



Cartes statiques

dismo

- ▶ Auteurs : Robert J. Hijmans, Steven Phillips, John Leathwick and Jane Elith
- ▶ Nécessite les packages : raster, XML, RGDAL, et sp,
- ▶ Fonctions utiles en cartographies :
 - ▶ geocode pour la localisation
 - ▶ gmap pour la projection
- ▶ Wrapper de RGoogleMaps pour des deux fonctions.

Cartes statiques

dismo

A partir d'un mot-clé :

```
library(dismo)

(x <- geocode("Paris, France"))

  ID      lon      lat  lonmin  lonmax  latmin  latmax
1  1  2.352222  48.85661  2.224162  2.480281  48.79785  48.91531 Paris

(e <- extent(unlist(x[4:7])))

class      : Extent
xmin       : 2.224162
xmax       : 2.480281
ymin       : 48.79785
ymax       : 48.91531

g <- gmap(e, type = "satellite")
```

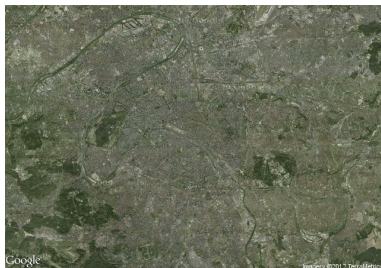
La fonction `extent()`, qui vient du package `raster`, récupère les coordonnées de la 'bounding box'.

Cartes statiques

dismo

On affiche dans x11 avec `plot()`

```
plot(g)
```



Cartes statiques

dismo

A partir d'un ensemble de points :

```
xy <- cbind(as.numeric(long), as.numeric(lat))  
g <- gmap(xy, type='terrain')
```

Cartes statiques

dismo

La projection est de type Mercator.

Pour ajouter des annotations, il faut donc transformer les données avec la fonction `Mercator()` :

```
Mxy <- Mercator(xy)
head(Mxy)
```

```
           x           y
[1,] 262714.0 6248553
[2,] 254654.5 6251478
[3,] 262613.8 6247541
[4,] 269085.9 6246484
[5,] 277882.4 6222989
[6,] 235073.4 6246145
```

Cartes statiques

dismo

```
plot(g)  
points(Mxy, col=2, pch=20)  
text(Mxy, labels=nom, col=2, pos=4)
```



Plan

Pourquoi faire ?

Cartes statiques
RgoogleMaps
dismo

Cartes dynamiques
googleVis
plotGoogleMaps

Autres possibilités

Références

Cartes dynamiques

googleVis

Google Visualization API, Google Chart Tools : application google pour produire des graphiques dynamiques :

<http://code.google.com/apis/chart/>

- ▶ Interface R – Google Chart Tools,
- ▶ Auteurs : Markus Gesmann, Diego de Castillo,
- ▶ <http://code.google.com/p/google-motion-charts-with-r/>
- ▶ Trois fonctions pour faire des cartes :
 - ▶ `gvisMap()`,
 - ▶ `gvisGeoMap()`,
 - ▶ `gvisIntensityMap()`.

Cartes dynamiques

googleVis

Principe :

1. création d'un `data.frame` avec les données de localisation par des :
 - ▶ mots-clé
 - ▶ données latitude / longitude en décimales
2. création du code `html / javascript` avec `gvisMap()`
3. projection avec `plot()`

Cartes dynamiques

googleVis

Les arguments pour les données :

- ▶ `data` ⇒ `data.frame`,
- ▶ `locationvar` ⇒ chaîne de caractères, le nom de la colonne de `data` contenant les données de localisation,
- ▶ `tipvar` ⇒ chaîne de caractères, le nom de la colonne de `data` contenant le texte des étiquettes, accepte du code `html`.

Cartes dynamiques

googleVis

Localisation par mots-clé :

```
library(googleVis)
df <- data.frame(
  adresses = c("Paris", "45 rue Buffon 75005 Paris"),
  tip = c("Centre de Paris",
         "UMR 7205 OSEB<BR>C'est ici!")
)
head(df)
```

	adresses	tip
1	Paris	Centre de Paris
2	45 rue Buffon 75005 Paris	UMR 7205 OSEB C'est ici!

```
mymap1 <- gvisMap(df,
  locationvar = "adresses",
  tipvar = "tip",
  options = list(showTip = TRUE))
plot(mymap1)
```

[voir le résultat](#)

Cartes dynamiques

googleVis

Localisation par donnée longitude / latitude :

```
latlong <- paste(lat, long, sep=":")  
df <- data.frame(latlong, nom, stringsAsFactors = FALSE)  
head(df)
```

	latlong	nom
1	48.84471:2.36	Jardin des Plantes
2	48.862:2.2876	Musée de l'Homme
3	48.83873:2.3591	IPH
4	48.83248:2.41724	Zoo de Vincennes
5	48.69335:2.49626	Brunoy
6	48.83047:2.1117	Arboretum de Chèvreloup

```
mymap2 <- gvisMap(df, locationvar="latlong",  
                  tipvar="nom",  
                  options = list(showTip = TRUE))  
plot(mymap2)
```

[voir le résultat](#)

Cartes dynamiques

googleVis

L'argument graphique option est une list avec les différentes options :

- ▶ `gvis.editor = TRUE` ⇒ modification en ligne,
- ▶ `enableScrollWheel = TRUE` ⇒ zoom,
- ▶ `showTip = TRUE` ⇒ étiquettes pour chaque point,
- ▶ `showLine = TRUE` ⇒ ligne entre les points,
- ▶ `lineColor` ⇒ couleur des lignes,
- ▶ `lineWidth` ⇒ épaisseur des lignes,
- ▶ `mapType` ⇒ type de carte (`normal`, `terrain`, `satellite`, `hybrid`)
- ▶ `useMapTypeControl = TRUE` ⇒ changement de type de fond de carte en ligne,
- ▶ `zoomLevel` ⇒ niveau de zoom,
- ▶ `width`, `height` ⇒ largeur et hauteur en pixels.

Cartes dynamiques

googleVis

```
mymap3 <- gvisMap(  
  df,  
  locationvar="latlong",  
  tipvar="nom",  
  options = list(  
    gvis.editor = TRUE,  
    enableScrollWheel = TRUE,  
    showTip = TRUE,  
    mapType = "normal",  
    useMapTypeControl = TRUE,  
    width=800, height=800  
  )  
)  
plot(mymap3)
```

[voir le résultat](#)

Cartes dynamiques

googleVis

Exemple de distribution d'une cigale méditerranéenne : *Dimissalna dimissa* :

<http://sueur.jerome.perso.neuf.fr/pubmore.html>

Cartes dynamiques

googleVis

Exemple de tracking : données sur le trajet d'un individu de Bernache nonette (*Branta leucopsis*)

Source : Larry Griffin et David Cabot <http://www.movebank.org>

Les données :

```
head(goose)
```

```
      location.long location.lat
8553      -10.22300      54.11583
8554      -10.22300      54.11583
8555      -10.22350      54.11533
8556      -10.20200      54.11916
8557      -10.20183      54.11933
8558      -10.20100      54.13066
```

Cartes dynamiques

googleVis

	latlong	nom
1	54.11583:-10.223	Barnacle goose
2	54.11583:-10.223	Barnacle goose
3	54.11533:-10.2235	Barnacle goose
4	54.11916:-10.202	Barnacle goose
5	54.11933:-10.20183	Barnacle goose
6	54.13066:-10.201	Barnacle goose

Cartes dynamiques

googleVis

La carte :

```
mymap4 <- gvisMap(df, locationvar="latlong",  
                 options = list(  
                   showTip = FALSE,  
                   showLine = TRUE,  
                   mapType = "satellite",  
                   lineColor="red"  
                 )  
               )  
plot(mymap4)
```

[voir le résultat](#)

Cartes dynamiques

googleVis

Il y a deux autres fonctions pour des cartes de densité par pays :

- ▶ `gvisIntensityMap()`,
- ▶ `gvisGeoMap()`.

Cartes dynamiques

plotGoogleMaps

- ▶ Auteur : Milan Kilibarda,
- ▶ Deux fonctions :
 - ▶ `plotGoogleMaps()` cartes dynamique avec projection de données spatiales (polygones, points),
 - ▶ `bubbleGoogleMaps()` carte dynamique avec projection des points modifiables (bubbles).

Cartes dynamiques

plotGoogleMaps

Object de classe `Spatial` du package `sp` qui contient :

- ▶ la 'bounding box',
- ▶ le CRS (coordinate reference system)
- ▶ des sous-classes selon le type de données spatiales
 - ▶ `SpatialPoints`, `SpatialPointsDataFrame`,
 - ▶ `SpatialLines`, `SpatialLinesDataFrame`,
 - ▶ `SpatialPolygons`, `SpatialPolygonsDataFrame`,
 - ▶ `SpatialPixels`,
 - ▶ `SpatialGrid`

Cartes dynamiques

plotGoogleMaps

Données meuse et meuse.grid du package sp.

Préparation des données :-(

Données de différentes mesures de pollution

```
library(plotGoogleMaps)
data(meuse)
coordinates(meuse) <- ~x+y
proj4string(meuse) <- CRS('+init=epsg:28992')
```

Grille 40 m × 40 m (localisation des points)

```
data(meuse.grid)
coordinates(meuse.grid) <- c('x', 'y')
meuse.grid <- as(meuse.grid, 'SpatialPixelsDataFrame')
im <- as.image.SpatialGridDataFrame(meuse.grid['dist'])
cl <- ContourLines2SLDF(contourLines(im))
proj4string(cl) <- CRS('+init=epsg:28992')
```

Cartes dynamiques

plotGoogleMaps

Projection simple des points avec étiquettes complètes, écriture dans un fichier `.html` :

```
m <- plotGoogleMaps(meuse,filename='mymap1.htm')
```

[voir le résultat](#)

Cartes dynamiques

plotGoogleMaps

Création d'un carte avec plusieurs calques (projections). Premier calque avec les points de mesure de pollution :

```
mapMeusePoints <- plotGoogleMaps(meuse,  
                                add=TRUE,  
                                mapTypeId='TERRAIN')
```

Deuxième calque avec les lignes de contours (c1) de niveau d'inondation :

```
mapMeuseC1 <- plotGoogleMaps(c1,  
                             previousMap=mapMeusePoints,  
                             filename='mymap2.htm')
```

[voir le résultat](#)

Cartes dynamiques

plotGoogleMaps

Project du taux de pollution au zinc par des points de taille proportionnelle :

```
m <- bubbleGoogleMaps(meuse,  
                        zcol='zinc',  
                        filename='mymap3.htm')
```

⇒ création d'un fichier .html et d'un fichier .png pour la légende.

[voir le résultat](#)

Plan

Pourquoi faire ?

Cartes statiques
RgoogleMaps
dismo

Cartes dynamiques
googleVis
plotGoogleMaps

Autres possibilités

Références

Autres possibilités

- ▶ Écriture de fichiers `.kml` (Google Earth) avec la fonction `writeOGR()` du package `rgdal`,
- ▶ package `DeducerSpatial` : un GUI pour la projection spatiale fonctionnant avec `JGR`. Vidéo [ici](#).
- ▶ package `ggmap` fonctionnant sur le principe de `ggplot2`.

Plan

Pourquoi faire ?

Cartes statiques
RgoogleMaps
dismo

Cartes dynamiques
googleVis
plotGoogleMaps

Autres possibilités

Références

Références

- ▶ Bivand *et al.* 2008 – Applied spatial data analysis with R. Use R! Springer.
- ▶ `dismo` : <http://cran.r-project.org/web/packages/dismo/index.html>
- ▶ `plotGoogleMaps` : <http://cran.at.r-project.org/web/packages/plotGoogleMaps/index.html>
- ▶ `RgoogleMaps` :
<http://cran.r-project.org/web/packages/RgoogleMaps/index.html>
- ▶ `googleVis` :
<http://cran.r-project.org/web/packages/googleVis/index.html>