# Tutoriel prevR

Version du 30 novembre 2007 pour prevR 1.11

prevR a été développé par Joseph LARMARANGE dans le cadre d'un projet de recherche de l'IRD et du Centre Muraz financé par l'ANRS (Agence Nationale de Recherche sur le VIH/Sida). Ce projet, numéroté ANRS 12114, porte sur la mesure et les estimations des prévalences nationales du VIH en Afrique subsaharienne.

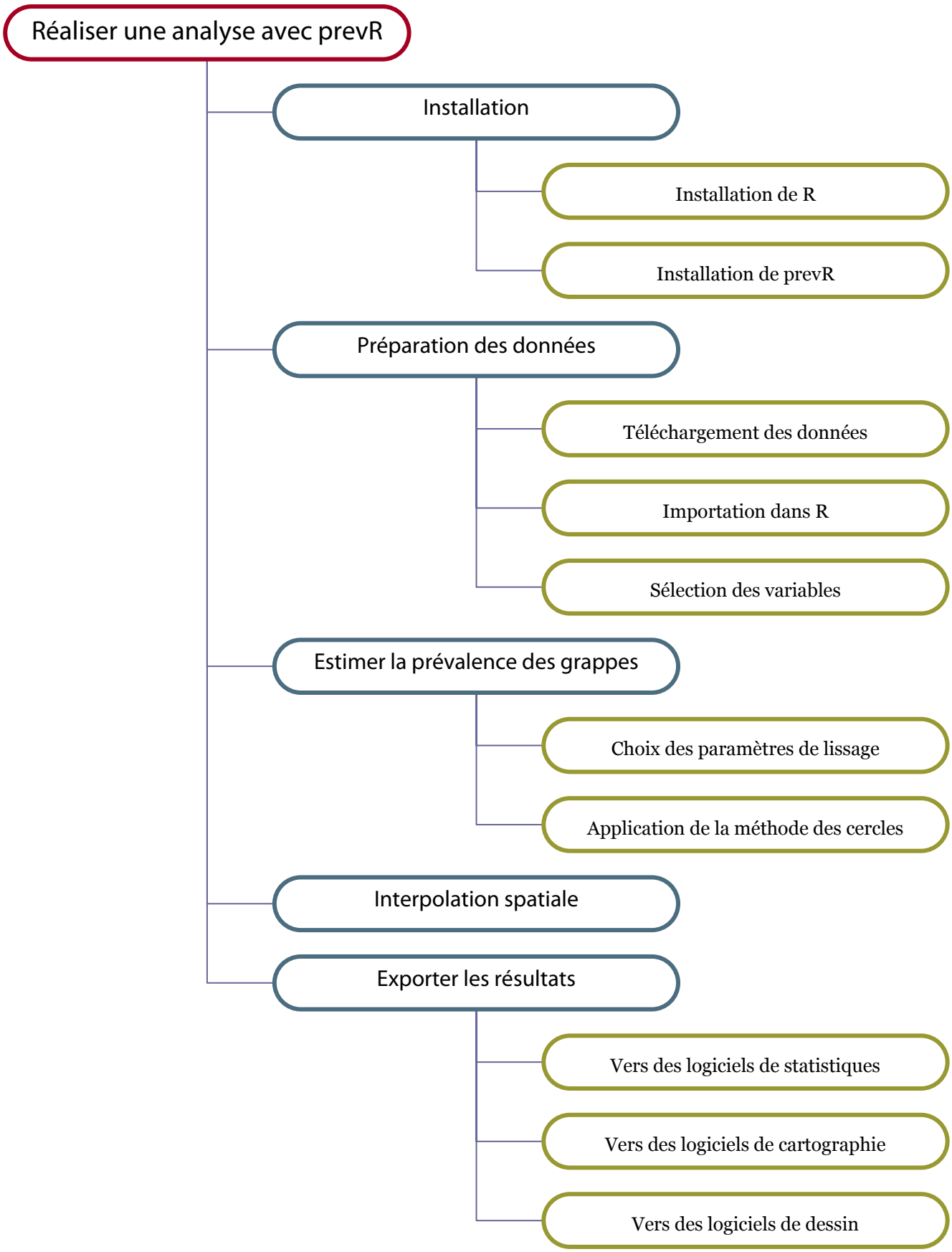
prevR a été conçu initialement pour représenter les variations spatiales de la prévalence du VIH à partir des données des Enquêtes Démographiques et de Santé (EDS ou DHS). Les exemples présentés ici portent donc sur cette problématique. Cependant, prevR peut être utilisé pour représenter tout type d'indicateur correspondant à une proportion dans le cadre d'enquêtes présentant un échantillonnage comparable à celui des EDS, c'est-à-dire un sondage en grappes.

prevR permet d'importer des données de type EDS, de les formater, puis de cartographier la prévalence d'un phénomène par estimation de la prévalence de chaque zone enquêtée (méthode des cercles) et interpolation spatiale (krigeage ordinaire). Les résultats peuvent être ensuite exportés vers d'autres logiciels de statistiques ou de cartographie (SIG). Des exemples d'exportation seront fournis à la fin de ce document.



Le site officiel de distribution de prevR est http://www.ceped.org/prevR/. prevR est également disponible sur http://joseph.larmarange.net/prevR/. Ce second site propose également un forum de discussion utilisateur ainsi qu'une mailing liste pour être tenu informé des mises à jour de prevR.

# Plan d'ensemble du tutoriel



# Table des Matières

**Note :**

Ce tutoriel a été conçu en priorité pour des utilisateurs ayant une connaissance minimum de R. La lecture du document **R pour les débutants** d'Emmanuel Paradis est donc fortement conseillée. Ce dernier est téléchargeable sur http://cran.r-project.org/other-docs.html.

Pour des utilisateurs expérimentés de R, nous leur conseillons de lire la documentation des fonctions de prevR.

Pour plus de détails techniques sur les méthodologies employées, nous vous renvoyons aux documents suivants :

> J. Larmarange, S. Yaro, R. Vallo, P. Msellati, N. Méda et B. Ferry, « Cartographier les données des enquêtes démographiques et de santé à partir des coordonnées des zones d'enquêtes », *Chaire Quételet 2006*, 29 novembre au 1er décembre 2006, Université Catholique de Louvain, Louvain-la-Neuve, Belgique[1].
>
> J. Larmarange, *Prévalences du VIH : validité d'une mesure*, chapitre 4, thèse de doctorat en démographie sous la direction de Benoît Ferry, Université Paris Descartes, 2007, disponible en ligne sur http://joseph.larmarange.net/.

# 1. Installation de R

## 1.1 Installation sous Windows

Il vous faut d'abord télécharger la dernière version de l'installateur pour Windows sur CRAN (Comprehensive R Archive Network) à cette adresse : http://cran.r-project.org/.

Cliquez sur *Windows (95 or later)*, puis sur *base*, et télécharger le fichier d'installation *R-2.6.0-win32.exe* (le numéro de version peut évoluer).

Lors de l'installation :

- Choisissez le dossier de destination (par défaut R sera installé dans *C:\Program Files\R*.
- Composants à installer : choisissez *Installation utilisateur complète*.
- Dans les options de démarrage, nous vous conseillons de choisir *Démarrage personnalisé*.
- Mode d'affichage : sélectionnez le mode *SDI* (ce dernier mode est nécessaire si vous vous voulez exploiter pleinement Tinn-R, voir 1.3).
- Style d'aide : si vous ne savez pas quoi choisir, sélectionnez le mode *CHM*.
- Accès internet : si vous avez un doute, sélectionnez *standard*.

[1] Ce document est disponible en ligne sur le site de l'UCL à http://www.uclouvain.be/13881.html ou sur le site du premier auteur à http://joseph.larmarange.net.

- Choisissez où vous souhaitez faire apparaître des raccourcis.

Pour information, il est possible de faire fonctionner R sur une clé USB. Pour cela, copier le répertoire *c:\Program Files\R* sur votre clé USB. Pour démarrer R, cliquez sur *Clé:\R\R-2.6.0\bin\Rgui.exe*.

### 1.2 Installation sous MacOS ou Linux

Pour une installation sur une autre plateforme que Windows, nous vous renvoyons à la documentation de R, en particulier au document intitulé **R Installation and Administration** disponible à cette adresse http://cran.r-project.org/manuals.html.

### 1.3 Optionnel : installer quelques projets d'amélioration de l'interface de R : Tinn-R et SciViews R Gui

Pour éditer du code R, nous vous recommandons le logiciel Tinn-R disponible gratuitement en ligne à cette adresse http://www.sciviews.org/Tinn-R/.

Vous pouvez également améliorer l'interface de R à l'aide du projet SciViews R Gui disponible en ligne à cette adresse http://www.sciviews.org/SciViews-R/index.html.

Il existe plusieurs projets d'amélioration de l'interface de R. Tous les renseignements sont disponibles sur http://www.sciviews.org/_rgui/.

## 2. Installation de prevR

### 2.1 Installation sous Windows.

Il faut d'abord installer les packages suivants qui sont nécessaires au fonctionnement de prevR :

- *sp*,
- *gstat*,
- *fields* et
- *maptools*.

Par ailleurs, nous vous recommandons vivement d'installer également les packages :

- *lattice* (fonctions graphiques supplémentaires) et
- *RSvgDevice* (permet d'exporter au format *SVG*).

L'ensemble de ces packages sont disponible sur CRAN. Ils peuvent être installés facilement dans R si vous disposez d'une connexion internet :

- Lancez R.
- Cliquez sur *Packages* > *Installer le(s) package(s)*.
- Sélectionnez un site miroir proche de vous.
- À l'aide de la touche *CTRL*, sélectionnez les packages suivants : *sp, gstat, fields, maptools, lattice* et *RSvgDevice*.

R téléchargera automatiquement et installera automatiquement ces différents packages.

Si vous ne disposez pas d'une connexion internet, vous pouvez télécharger manuellement ces différents package sur CRAN (choisissez le format *zip*) et les installer à partir de la commande *Packages* > *Installer le(s) package(s) des fichiers zip* disponible dans les menus de la console R (voir ci-dessous).

Il reste à installer prevR à partir du fichier *zip* disponible sur les sites du CEPED (http://www.ceped.org/prevR/) et de Joseph Larmarange (http://joseph.larmarange.net/prevR/) :

- Téléchargez prevR au format *zip* et enregistrez le sur votre disque dur.
- Dans R, cliquez sur *Packages* > *Installer le(s) package(s) des fichiers zip*.
- Sélectionnez le fichier adéquat.

## 2.2 Installation sous MacOS ou Linux

Nous vous renvoyons à la section 6 du document **R Installation and Administration**, disponible à cette adresse http://cran.r-project.org/manuals.html, qui détaille la procédure à suivre pour installer des packages à partir des fichiers sources (distribués sous la forme d'archives *tar.gz*).

Les fichiers sources de prevR sont disponibles sur les site du CEPED (http://www.ceped.org/prevR/) et de Joseph Larmarange (http://joseph.larmarange.net/prevR/). Téléchargez l'archive au format *tar.gz*.

Les packages nécessaires au fonctionnement de prevR sont disponibles sur CRAN (http://cran.r-project.org). Avant d'installer prevR, vous devrez installer les packages suivants : *sp*, *gstat*, *fields* et *maptools*. Nous vous recommandons d'installer également *lattice*, *RcolorBrewer*, *Rarcinfo* et *RSvgDevice*.

# 3. Remarques générales sur l'utilisation de R et de prevR

R est un langage, orienté objets, de programmation statistique basé sur le langage S. R est sensible à la casse des caractères. Ainsi, l'objet AbcD sera différent de ABCD ou encore de abcd. Nous vous déconseillons fortement d'utiliser des caractères accentués dans le nom des objets que vous manipulez avec R.

> De nombreuses ressources sont disponibles sur internet pour vous initier à R. Outre une recherche avec Google, nous vous recommandons de consulter cette page http://cran.r-project.org/other-docs.html et en particulier le document **R pour les débutants** d'Emmanuel Paradis.
>
> **La suite de ce tutoriel présuppose que vous ayez lu au minimum ce document.**

Lorsqu'un package est installé, les fichiers du package sont copiés dans le répertoire d'installation de R. Cependant, il n'est pas directement utilisable. En effet, lorsque que vous démarrez R, seules les fonctionnalités de base de R sont chargées en mémoire. Pour pouvoir utiliser les fonctions d'un package particulier, il faut au préalable le charger en mémoire, et ce à chaque fois que vous démarrez R.

## 3.1 Charger prevR, obtenir de l'aide, citer prevR et démonstration

Pour utiliser un package, vous devez le charger en mémoire. Deux solutions possibles :

- Utilisez le menu *Packages* > *Charger le package*.
- Utilisez la commande library ou la commande require.

Ainsi, pour charger prevR, il suffit de taper :

```
> library(prevR)
Le chargement a nécessité le package : fields
fields is loaded use help(fields) for an overview of this library
Le chargement a nécessité le package : sp
Le chargement a nécessité le package : gstat
Le chargement a nécessité le package : maptools
Le chargement a nécessité le package : foreign
```

Les autres packages nécessaires à l'utilisation de prevR sont chargés automatiquement.

Pour obtenir de l'aide sur une fonction dans R, il suffit de taper ? suivi du nom de la fonction ou du package, ou bien la fonction help. Ainsi, pour obtenir une aide générale sur prevR ou sur la fonction krige.prev, il suffit de taper :

```
> ?prevR
> help('prevR')
> ?krige.prev
```

Chaque fonction documentée fournit des exemples d'utilisation. Il est possible d'exécuter les exemples d'une fonction à l'aide de example(). La fonction mean() de R permet de calculer une moyenne. Pour voir des exemples d'utilisation de cette fonction il suffit de taper :

```
> example(mean)

mean> x <- c(0:10, 50)

mean> xm <- mean(x)

mean> c(xm, mean(x, trim = 0.1))
[1] 8.75 5.50

mean> mean(USArrests, trim = 0.2)
  Murder  Assault UrbanPop     Rape
    7.42   167.60    66.20    20.16
```

Pour rechercher un texte dans la documentation de R et de ses packages, il suffit d'avoir recours à la fonction help.search :

```
> help.search('proportion') # Pour chercher les fonctions travaillant sur des
proportions
```

**NB :** On notera que des commentaires peuvent être mis dans du code R à l'aide du caractère #. Le texte qui suit ce caractère n'est alors pas interprété.

prevR est livré avec un jeu de données permettant d'essayer ses différentes fonctions. Ce jeu de données est appelé alicante. Ces données sont issues de la simulation d'une Enquête Démographique et de Santé sur un pays fictif présentant une prévalence nationale de 10 pour cent. 8 000 personnes ont été enquêtées, réparties en 401 clusters.

Pour charger les données en mémoire, il suffit d'avoir recours à la fonction data. La fonction ls permet à tout moment de savoir quels objets sont présents en mémoire. Pour plus de détails sur les données fournies par alicante, il suffit de consulter l'aide associée.

```
> data(alicante)

> ls()
[1] "alicante.bounds" "alicante.cities" "alicante.clust"  "alicante.krige"  "alicante.prev"

> ?alicante
```

Pour savoir comment citer prevR dans un article, vous pouvez avoir recours à la fonction citation().

```
> citation('prevR')
To cite package prevR in publications use:
 Joseph Larmarange et al., 2006, 'Cartographier les données des enquêtes démographiques et de santé
à partir des coordonnées des zones  d'enquête', Chaire Quételet, 29 novembre au 1er décembre 2006,
Université Catholique de Louvain, Louvain-la-Neuve, Belgique (http://www.uclouvain.be/13881.html).
```

Pour une démonstration des possibilités du package prevR, utilisez la fonction demo().

```
> demo(prevR)
```

**ATTENTION :** la démonstration peut prendre plusieurs minutes, selon la puissance de votre machine, certaines fonctions ayant des temps de calcul relativement longs.

## 3.2 Quelques fonctions de base

Voici une liste, non exhaustive, de quelques fonctions de base particulièrement utiles. Pour plus de détails, voir l'aide de chaque fonction.

| Nom | Description |
|---|---|
| as.character | Pour passer un objet en mode texte. |
| as.factor | Pour passer un objet en mode facteurs. |
| as.numeric | Pour passer un objet en mode numérique. |
| c | Permet de créer un vecteur. |
| data | Charge des données contenues dans un package. |
| demo | Permet d'exécuter une démonstration. |
| dev.copy | Copie une sortie graphique vers une autre sortie graphique. |
| dev.off | Ferme la sortie graphique courante. |
| dev.set | Active une sortie graphique. |
| edit | Édite un objet. |
| example | Exécute les exemples fournis dans la documentation d'une fonction. |
| function | Pour écrire ses propres fonctions. |
| getwd | Affiche le répertoire de travail courant. |
| graphics.off | Ferme toutes les fenêtres graphiques. |
| help | Fournit de l'aide sur une fonction. |
| help.search | Effectue une recherche dans l'aide. |
| levels | Affiche les étiquettes de valeur d'un objet de type facteurs. |
| library | Charge un package en mémoire. |
| list | Pour créer des listes. |
| load | Charge un fichier de données. |
| ls | Liste l'ensemble des objets en mémoire. |
| order | Pour trier des données. |
| rm | Supprime un objet. |
| save | Sauve un ou plusieurs objets dans un fichier de données. |
| save.image | Sauve l'ensemble des objets en mémoire. |
| seq | Permet de générer une liste de nombres. |
| setwd | Définit le répertoire de travail. |
| str | Détaille la structure d'un objet. |
| summary | Fournit un résumé détaillé du contenu d'un objet. |
| write.table | Exporte des données sous la forme d'un fichier texte. |
| x11 | Ouvre une nouvelle fenêtre graphique. |

Pour des manipulations avancées des tableaux de données, nous vous conseillons la lecture de l'aide de l'opérateur [ :

```
> help('[.data.frame')
```

# 4. Préparation des données

Nous illustrerons l'utilisation de prevR à travers l'estimation des variations spatiales de la prévalence du VIH au Cameroun à partir de l'Enquête Démographique et de Santé de 2004.

## 4.1 Téléchargement des données

Afin de réaliser cette analyse, il est nécessaire de récupérer les données suivantes :

- localisation des zones d'enquêtes de l'EDS ou clusters,
- résultats au test VIH des personnes enquêtées,
- frontières du pays (sous la forme d'un polygone géoréférencé),
- localisation des principales villes du pays (coordonnées longitude/latitude).

### 4.1.1 Bases de données des EDS

Les données des enquêtes EDS peuvent être obtenues gratuitement en ligne sur le site de Measure DHS : http://www.measuredhs.com/. Si vous n'êtes pas inscrit, il vous faudra créer un compte, décrire votre projet de recherche et demander l'accès aux données du pays intéressé. Il faut réaliser une demande spécifique pour l'accès aux données VIH et une autre pour l'accès aux données GPS. Pour l'obtention des données GPS, un formulaire d'engagement éthique devra être imprimé, signé et à retourné à Measure DHS. Il faut compter parfois quelques jours avant que l'accès aux données ne vous soit notifié.

Les données d'enquêtes ou les résultats au test VIH doivent être téléchargées au format SPSS (fichiers avec le suffixe *sv.zip*) ou au format rectangulaire (fichiers avec le suffixe *rt.zip*). Décompressez les archives et copiez le fichier portant l'extension *.sav* dans un répertoire de travail. Dans notre exemple, nous avons renommé le fichier des résultats du test VIH de l'EDS 2004 du Cameroun en *cm.hiv.sav* pour plus de commodités.

Les données GPS sont téléchargeables directement au format *dbf*. Par commodité, nous avons renommé ce fichier en *cm.gps.dbf*.

### 4.1.2 Frontières du pays via le DCW

Il existe plusieurs bases de données cartographiques fournissant les frontières nationales des différents pays du monde. L'une des plus connues est le *Digital Chart of the World*, dont les données sont téléchargeables gratuitement en ligne sur http://www.maproom.psu.edu/dcw/. prevR fournit une fonction permettant d'importer facilement un fichier de points téléchargé depuis ce site.

Cependant, il faut noter que les données du DCW datent de 1992 et n'ont pas été actualisées. Elles ne seront donc plus valables si les frontières du pays étudié ont subi des modifications depuis cette date.

Vous pouvez néanmoins utiliser toute autre source pour définir les limites de votre zone d'études. Il vous suffit de les importer et de les mettre en forme de manière à obtenir un *data.frame* (tableau de données) avec deux colonnes nommées *x* et *y* dont chaque ligne correspond à un point d'un polygone fermé[2].

Dans le présent exemple, nous utiliserons les données du DCW pour le Cameroun. Après avoir sélectionné votre continent et votre pays, choisissez l'option *Download Points*. Faites un clic droit sur *download data* et choisissez *Enregistrez la cible du lien sous*. Vous obtiendrez alors un fichier de la forme *pays2pts.txt* et soit *cameroon2pts.txt* dans notre exemple.

### 4.1.3 Position des principales villes via le GRUMP

Pour prendre en compte les principales agglomérations urbaines dans l'analyse, il est nécessaire de connaître les coordonnées des principales villes du pays. Celles-ci peuvent être obtenues gratuitement en ligne à partir du projet Global Rural-Urban Mapping Project (GRUMP). Les données de ce projet sont accessibles à cette adresse : http://sedac.ciesin.org/gpw/.

Arrivé sur leur site, choisissez *downloadable data*. Sélectionner le pays qui vous intéresse dans la liste située en fin de page. Dans la partie *Select a product*, choisissez *Get GRUMP > Settlement Points*. Dans *Select Options*, sélectionnez le format *csv* et l'année *circa 2000*.

Vous devrez vous enregistrer pour pouvoir télécharger le fichier désiré. Certains utilisateurs peuvent rencontrer des soucis de téléchargement avec le logiciel *Internet Explorer*. Préférez dans ce cas là le navigateur libre *Firefox*, téléchargeable gratuitement sur http://www.mozilla-europe.org/fr/products/firefox/.

Après décompression de l'archive, placez le fichier *csv* dans votre répertoire de travail. Par commodité, nous avons renommé le fichier obtenu dans notre exemple en *cm.cities.csv*.

**NB :** si vous comptez exporter ultérieurement vos résultats vers un logiciel de cartographie, vous pouvez également télécharger la position des villes au format *shapefile* pour pouvoir habiller vos cartes.

## 4.2 Importation et mise en forme des données

Nous vous recommandons de placer les différents fichiers obtenus dans un même répertoire que nous appellerons répertoire de travail.

Au lancement de R, utilisez la commande *Fichier > Changer le répertoire courant…* pour spécifier votre répertoire de travail.

Puis, utilisez la commande *Packages > Charger le package…* pour charger prevR en mémoire.

[2] Une aire géographique est définie d'un point de vue informatique par une série de points, chacun défini par une latitude et une longitude, formant un polygone. Voir 4.2.4 pour plus de détails.

### 4.2.1 Création du fichier individu

La fonction make.ind.spss a été spécialement conçue pour lire et mettre en forme des données individuelles, au format SPSS, fournies par Measure DHS. Elle prend deux paramètres : le nom du fichier SPSS à importer et le code de langue pour l'interface utilisateur ('*en*' pour l'anglais, '*fr*' pour le français).

```
> cm.ind <- make.ind.spss('cm.hiv.sav',lang='fr')
```

La fonction commence par lire les différentes variables contenues dans le fichier SPSS, puis vous demande de spécifier un certain nombre d'entre elles :

```
Veuillez indiquer les variables suivantes :

Identifiant (0 s'il n'existe pas) :

 1: ACASEID - space dummy                          2: HIVCLUST - Cluster number
 3: HIVHHN - HH structure                          4: HIVMENA - HH menage
 5: HIV60 - Line number of respondent              6: HIV62 - Sex of household member
 7: HIV63 - Age of household members               8: HIV64 - Age 15-17, 18+
 9: HIV65 - Line number of parent/responsible     10: HIV66 - Consent statement to parent
11: HIV67 - Consent to woman/man                  12: HIV68 - Sample result
13: HIVREG - Province                             14: HIVLOC - Locality
15: HIVTYPE - Urban/rural                         16: HDEFACTO - Slept last night
17: HIVEDUC - Level of education attending        18: HIVGRADE - Highest grade of education
19: HIVQNL - Sequence order in section            20: INDINT - Result of individual interview
21: TESTED - Found and tested in LAB file         22: HHDUP - Duplicate ID in HH
23: LABDUP - Duplicate ID in LAB file             24: HIVWT - Weight for HIV sample
25: RESULT.1 - Result of testing                  26: RESULT.2 - Result of testing
27: RESULT.3 - Result of testing                  28: FRESULT - Final result of testing
29: HIVCHILD - Had a child in last 5 years        30: HIVANC - Received ANC in last 5 years
31: HIVPET - Cluster in petrol line

Sélection : 1

Numéro du cluster (0 s'il n'existe pas. Le numéro de cluster sera alors calculé à partir des
identifiants.) :
Sélection : 0

Age (0 s'il n'existe pas) :
Sélection : 7

Sexe (0 s'il n'existe pas) :
Sélection : 6

Variable analysée (par exemple, résultat du test VIH) :
Sélection : 28

Poids statistique (0 s'il n'existe pas. Tous les individus auront un poids égal à 1.) :
Sélection : 24
```

Dans le cas présent, bien que la variable numéro du cluster était présente (variable 2), nous ne l'avons pas entrée de manière à montrer comment calculer le numéro de cluster à partir des identifiants.

Une fois les variables saisies, la fonction vous demande de confirmer votre choix. En cas d'erreur, sélectionnez *Non* et recommencez la saisie :

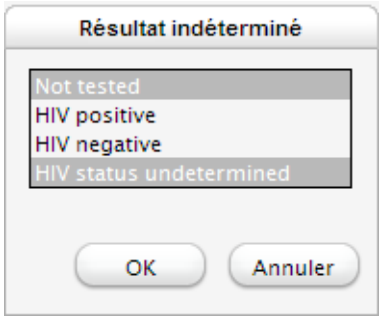
```
ATTENTION : veuillez vérifier les informations suivantes :

* Identifiant des individus :
ACASEID - space dummy
* Numéro de cluster :
Non renseigné - Il sera calculé à partir du numéro d'identifiants
* Age :
HIV63 - Age of household members
* Sexe :
HIV62 - Sex of household member
* Variable analysée :
FRESULT - Final result of testing
* Poids statistique :
HIVWT - Weight for HIV sample
-------------------------------------------------
Ces données sont-elles correctes ?
1: Oui
2: Non
```

**Sélection : 1**

Vous êtes ensuite invité à spécifier comment la variable analysée a été codée.

```
Trois fenêtres vont s'ouvrir pour recoder la variable analysée.
Vous devrez spécifier les modalités correspondant à un résultat positif (le phénomène étudié a eu
lieu),
négatif (n'a pas eu lieu) ou indéterminé (considéré alors comme valeur manquante).
Vous pouvez sélectionner plusieurs modalités à l'aide de la touche CTRL.
Êtes-vous-prêt ?

1: Oui
```

**Sélection : 1**




Exemple de fenêtre, pour la modalité *indéterminé :*

Les fenêtres pour les modalités *Résultat positif* et *Résultat négatif* sont identiques.

Par sécurité, il vous est demandé de confirmer à nouveau votre saisie :

```
ATTENTION : veuillez vérifier les informations suivantes :

* Résultat positif :
- HIV positive

* Résultat négatif :
- HIV negative

* Résultat indéterminé :
- Not tested
- HIV status undetermined

Ces données sont-elles correctes ?

1: Oui
2: Non
```

**Sélection : 1**

Si une variable de pondération a été saisie, il vous est demandé s'il est nécessaire de la diviser par un facteur donné. Cela dépend bien entendu de votre enquête. Dans les EDS, il faut en général les diviser par 1 000 000.

```
Souvent, dans les EDS, la variable poids doit être divisée par un facteur, usuellement 1 000 000.
Sa valeur moyenne est de :
999999.41

Si cette valeur est proche de 1, a priori la variable n'a pas à être modifiée. Si elle est proche
de 1 000 000, alors elle doit être divisée par ce facteur, sinon consultez la documentation de
l'enquête.
La variable doit-elle être gardée telle quelle ou divisée par un facteur ?

1: Pas de modification
2: Division par 1 000 000
3: Division par un autre facteur
```

**Sélection : 2**

Lorsque le numéro de cluster n'est pas renseigné, il peut être calculé à partir de l'identifiant des individus si celui-ci contient le numéro de cluster (ce qui est le cas dans les EDS). Typiquement, le numéro de cluster correspond aux premiers chiffres de l'identifiant.

```
La variable cluster n'est pas renseignée et doit donc être calculée à partir du numéro
d'identifiant. Usuellement, dans les EDS, le numéro de cluster correspond aux trois premiers
chiffres du numéro d'identifiant. Voici trois numéros d'identifiants, pris au début, au milieu et à
la fin du fichier :
Numéro d'identification 1 :       1 26 3  1
Numéro d'identification 2 :     236149 6  2
Numéro d'identification 3 :     466312 3  4
```

Dans le cas présent, des trois identifiants prélevés dans la base, on voit clairement que les numéros de cluster correspondant sont 1, 236 et 466. La découpe adéquate des identifiants est donc la découpe numéro 5.

```
Repérez pour chacun d'eux le numéro de cluster. Parmi les différentes propositions ci-dessous,
laquelle extrait les bons numéros de cluster ?
  1: Cluster 1:              Cluster 2:                  Cluster 3:
  2: Cluster 1:              Cluster 2:                  Cluster 3:
  3: Cluster 1:              Cluster 2:    2             Cluster 3:    4
  4: Cluster 1:              Cluster 2:   23             Cluster 3:   46
  5: Cluster 1:    1         Cluster 2:  236             Cluster 3:  466
  6: Cluster 1:   1          Cluster 2:  361             Cluster 3:  663          .
  7: Cluster 1:  1 2         Cluster 2:  614             Cluster 3:  631
  8: Cluster 1:   26         Cluster 2:  149             Cluster 3:  312
  9: Cluster 1:  26          Cluster 2:  49              Cluster 3:  12
 10: Cluster 1:  6 3         Cluster 2:  9 6             Cluster 3:  2 3      .
 11: Cluster 1:   3          Cluster 2:   6              Cluster 3:   3
 12: Cluster 1:  3           Cluster 2:  6               Cluster 3:  3
 13: Cluster 1:    1         Cluster 2:    2             Cluster 3:    4
```

**Sélection : 5**

Une fois la fonction exécutée, on peut vérifier son résultat à l'aide de str().

```
> str(cm.ind)
'data.frame':   12065 obs. of  7 variables:
 $ id             : chr  "     1 26 3  1" "     1 26 3  2" "     1137 1  1" "     1137 1  2" ...
 $ age            : int  24 25 36 29 47 48 44 18 16 15 ...
 $ sex            : Factor w/ 2 levels "Male","Female": 2 2 1 2 2 1 2 2 2 1 ...
 $ original.result: Factor w/ 4 levels "Not tested","HIV positive",..: 1 1 3 3 3 3 3 3 3 3 ...
 $ weight         : num  0.00 0.00 1.25 1.16 1.16 ...
 $ result         : Factor w/ 2 levels "Negative","Positive": NA NA 1 1 1 1 1 1 1 1 ...
 $ cluster        : int  1 1 1 1 1 1 1 1 1 1 ...
```

Pour le détail des différentes variables, voir l'aide de make.ind.spss() :

```
> ?make.ind.spss
```

### 4.2.2 Création du fichier cluster

Les analyses effectuées par prevR portent essentiellement sur un tableau de données (*data.frame*). Pour voir la structure de celui-ci, vous pouvez vous référer au fichier *alicante.clust* fourni avec prevR.

```
> data(alicante)
> str(alicante.clust)
'data.frame':   401 obs. of  11 variables:[3]
 $ cluster       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ x             : num  -1.21 -1.87 -1.04 -1.37 -2.21 ...
 $ y             : num  7.29 7.54 7.96 6.46 6.88 ...
 $ residence     : Factor w/ 2 levels "Rural","Urban": 1 1 1 1 1 1 1 1 1 1 ...
 $ region        : num  1 1 1 1 1 1 1 1 1 1 ...
 $ n             : num  23 21 24 16 26 22 21 22 17 22 ...
 $ nweight       : num  19.8 19.8 19.8 19.8 19.8 ...
 $ obs.prevalence: num  0.00 4.76 0.00 0.00 3.85 ...
 $ dist.city     : num   72.9  93.2 146.9  39.8  70.5 ...
 $ city.name     : chr  "D" "D" "D" "D" ...
 $ urban.area    : Factor w/ 2 levels "in urban area",..: 2 2 2 2 2 2 2 2 2 2 ...
```

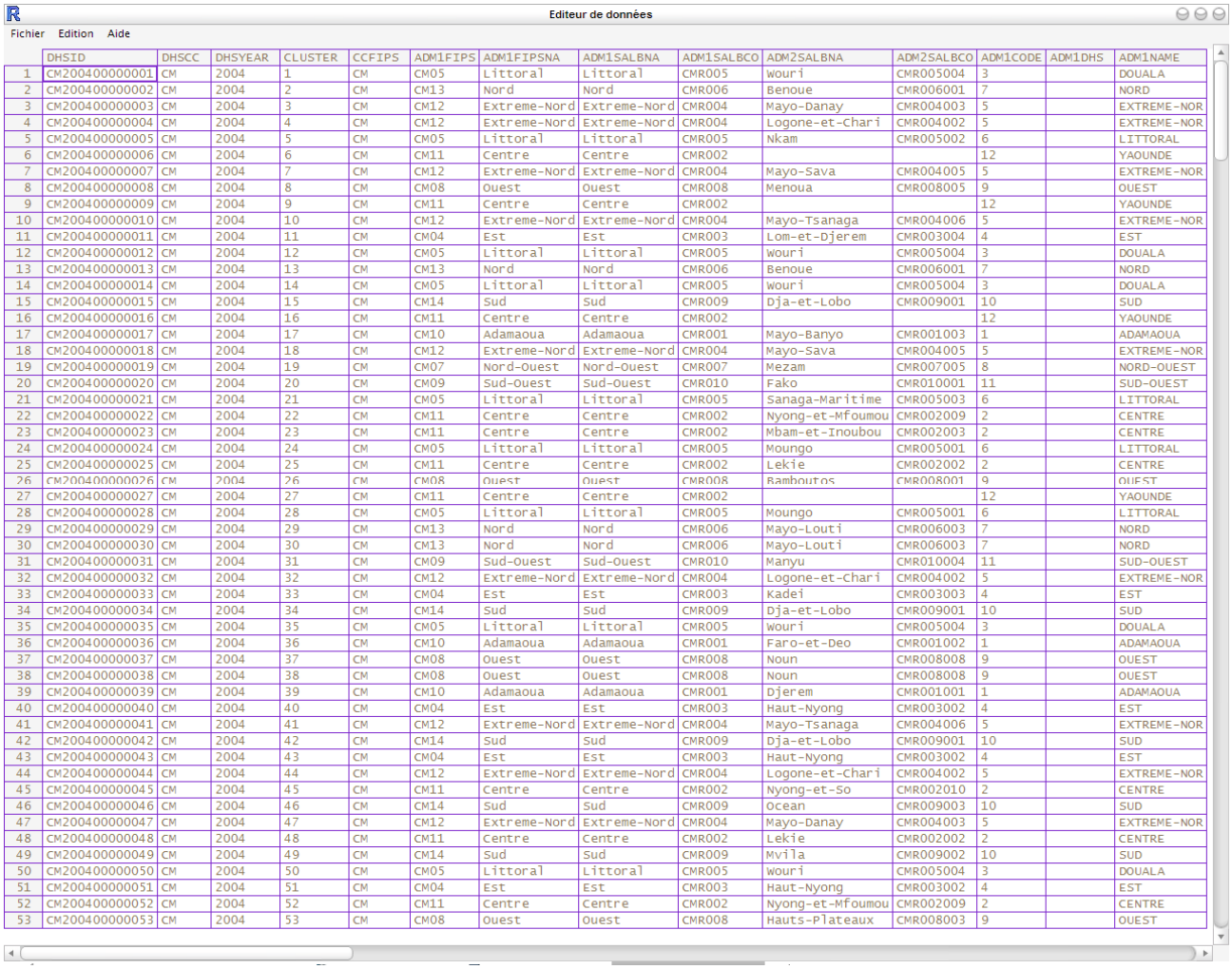
La fonction make.clust.dbf() permet de construire ce tableau de données à partir du fichier *dbf* fourni par Measure DHS et du fichier individu créé à l'étape précédente.

```
> cm.clust <- make.clust.dbf('cm.gps.dbf',cm.ind,lang='fr')
```

Dans un fichier *dbf*, on dispose seulement du nom des variables mais il n'y a pas d'étiquette spécifiant leur contenu comme dans un fichier SPSS. Une fenêtre présentant le contenu du fichier *dbf* s'affiche alors afin que vous puissiez identifier un certain nombre de variables. Une fois cela fait, fermez cette fenêtre pour pouvoir effectuer votre saisie.

```
Une fenêtre va s'ouvrir présentant les données contenues dans le fichier. Merci de repérer les
variables suivantes :
- Numéro du cluster (nécessaire)
- Longitude (en degrés au format décimal, nécessaire)
- Latitude (en degrés au format décimal, nécessaire)
- Milieu de résidence (urbain/rural, nécessaire si utilisation du paramètre U)
- Code numérique des régions (optionnel)
- Nom des régions (optionnel)
 Une fois les noms de ces variables identifiés, fermez la fenêtre pour que le programme puisse
continuer.

 Êtes-vous prêt ?
1: Oui

Sélection : 1
```

[3] Les variables *dist.city, city.name* et *urban.area* ne sont pas nécessaires au début de l'analyse. Elles seront rajoutées ultérieurement au tableau de données par la fonction calcul.dist.cities().



Editeur de données

Fichier Edition Aide

| | DHSID | DHSCC | DHSYEAR | CLUSTER | CCFIPS | ADM1FIPS | ADM1FIPSNA | ADM1SALBNA | ADM1SALBCO | ADM2SALBNA | ADM2SALBCO | ADM1CODE | ADM1DHS | ADM1NAME |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | CM200400000001 | CM | 2004 | 1 | CM | CM05 | Littoral | Littoral | CMR005 | Wouri | CMR005004 | 3 | | DOUALA |
| 2 | CM200400000002 | CM | 2004 | 2 | CM | CM13 | Nord | Nord | CMR006 | Benoue | CMR006001 | 7 | | NORD |
| 3 | CM200400000003 | CM | 2004 | 3 | CM | CM12 | Extreme-Nord | Extreme-Nord | CMR004 | Mayo-Danay | CMR004003 | 5 | | EXTREME-NOR |
| 4 | CM200400000004 | CM | 2004 | 4 | CM | CM12 | Extreme-Nord | Extreme-Nord | CMR004 | Logone-et-Chari | CMR004002 | 5 | | EXTREME-NOR |
| 5 | CM200400000005 | CM | 2004 | 5 | CM | CM05 | Littoral | Littoral | CMR005 | Nkam | CMR005002 | 6 | | LITTORAL |
| 6 | CM200400000006 | CM | 2004 | 6 | CM | CM11 | Centre | Centre | CMR002 | | | 12 | | YAOUNDE |
| 7 | CM200400000007 | CM | 2004 | 7 | CM | CM12 | Extreme-Nord | Extreme-Nord | CMR004 | Mayo-Sava | CMR004005 | 5 | | EXTREME-NOR |
| 8 | CM200400000008 | CM | 2004 | 8 | CM | CM08 | Ouest | Ouest | CMR008 | Menoua | CMR008005 | 9 | | OUEST |
| 9 | CM200400000009 | CM | 2004 | 9 | CM | CM11 | Centre | Centre | CMR002 | | | 12 | | YAOUNDE |
| 10 | CM200400000010 | CM | 2004 | 10 | CM | CM12 | Extreme-Nord | Extreme-Nord | CMR004 | Mayo-Tsanaga | CMR004006 | 5 | | EXTREME-NOR |
| 11 | CM200400000011 | CM | 2004 | 11 | CM | CM04 | Est | Est | CMR003 | Lom-et-Djerem | CMR003004 | 4 | | EST |
| 12 | CM200400000012 | CM | 2004 | 12 | CM | CM05 | Littoral | Littoral | CMR005 | Wouri | CMR005004 | 3 | | DOUALA |
| 13 | CM200400000013 | CM | 2004 | 13 | CM | CM13 | Nord | Nord | CMR006 | Benoue | CMR006001 | 7 | | NORD |
| 14 | CM200400000014 | CM | 2004 | 14 | CM | CM05 | Littoral | Littoral | CMR005 | Wouri | CMR005004 | 3 | | DOUALA |
| 15 | CM200400000015 | CM | 2004 | 15 | CM | CM14 | Sud | Sud | CMR009 | Dja-et-Lobo | CMR009001 | 10 | | SUD |
| 16 | CM200400000016 | CM | 2004 | 16 | CM | CM11 | Centre | Centre | CMR002 | | | 12 | | YAOUNDE |
| 17 | CM200400000017 | CM | 2004 | 17 | CM | CM10 | Adamaoua | Adamaoua | CMR001 | Mayo-Banyo | CMR001003 | 1 | | ADAMAOUA |
| 18 | CM200400000018 | CM | 2004 | 18 | CM | CM12 | Extreme-Nord | Extreme-Nord | CMR004 | Mayo-Sava | CMR004005 | 5 | | EXTREME-NOR |
| 19 | CM200400000019 | CM | 2004 | 19 | CM | CM07 | Nord-Ouest | Nord-Ouest | CMR007 | Mezam | CMR007005 | 8 | | NORD-OUEST |
| 20 | CM200400000020 | CM | 2004 | 20 | CM | CM09 | Sud-Ouest | Sud-Ouest | CMR010 | Fako | CMR010001 | 11 | | SUD-OUEST |
| 21 | CM200400000021 | CM | 2004 | 21 | CM | CM05 | Littoral | Littoral | CMR005 | Sanaga-Maritime | CMR005003 | 6 | | LITTORAL |
| 22 | CM200400000022 | CM | 2004 | 22 | CM | CM11 | Centre | Centre | CMR002 | Nyong-et-Mfoumou | CMR002009 | 2 | | CENTRE |
| 23 | CM200400000023 | CM | 2004 | 23 | CM | CM11 | Centre | Centre | CMR002 | Mbam-et-Inoubou | CMR002003 | 2 | | CENTRE |
| 24 | CM200400000024 | CM | 2004 | 24 | CM | CM05 | Littoral | Littoral | CMR005 | Moungo | CMR005001 | 6 | | LITTORAL |
| 25 | CM200400000025 | CM | 2004 | 25 | CM | CM11 | Centre | Centre | CMR002 | Lekie | CMR002002 | 2 | | CENTRE |
| 26 | CM200400000026 | CM | 2004 | 26 | CM | CM08 | Ouest | Ouest | CMR008 | Bamboutos | CMR008001 | 9 | | OUEST |
| 27 | CM200400000027 | CM | 2004 | 27 | CM | CM11 | Centre | Centre | CMR002 | | | 12 | | YAOUNDE |
| 28 | CM200400000028 | CM | 2004 | 28 | CM | CM05 | Littoral | Littoral | CMR005 | Moungo | CMR005001 | 6 | | LITTORAL |
| 29 | CM200400000029 | CM | 2004 | 29 | CM | CM13 | Nord | Nord | CMR006 | Mayo-Louti | CMR006003 | 7 | | NORD |
| 30 | CM200400000030 | CM | 2004 | 30 | CM | CM13 | Nord | Nord | CMR006 | Mayo-Louti | CMR006003 | 7 | | NORD |
| 31 | CM200400000031 | CM | 2004 | 31 | CM | CM09 | Sud-Ouest | Sud-Ouest | CMR010 | Manyu | CMR010004 | 11 | | SUD-OUEST |
| 32 | CM200400000032 | CM | 2004 | 32 | CM | CM12 | Extreme-Nord | Extreme-Nord | CMR004 | Logone-et-Chari | CMR004002 | 5 | | EXTREME-NOR |
| 33 | CM200400000033 | CM | 2004 | 33 | CM | CM04 | Est | Est | CMR003 | Kadei | CMR003003 | 4 | | EST |
| 34 | CM200400000034 | CM | 2004 | 34 | CM | CM14 | Sud | Sud | CMR009 | Dja-et-Lobo | CMR009001 | 10 | | SUD |
| 35 | CM200400000035 | CM | 2004 | 35 | CM | CM05 | Littoral | Littoral | CMR005 | Wouri | CMR005004 | 3 | | DOUALA |
| 36 | CM200400000036 | CM | 2004 | 36 | CM | CM10 | Adamaoua | Adamaoua | CMR001 | Faro-et-Deo | CMR001002 | 1 | | ADAMAOUA |
| 37 | CM200400000037 | CM | 2004 | 37 | CM | CM08 | Ouest | Ouest | CMR008 | Noun | CMR008008 | 9 | | OUEST |
| 38 | CM200400000038 | CM | 2004 | 38 | CM | CM08 | Ouest | Ouest | CMR008 | Noun | CMR008008 | 9 | | OUEST |
| 39 | CM200400000039 | CM | 2004 | 39 | CM | CM10 | Adamaoua | Adamaoua | CMR001 | Djerem | CMR001001 | 1 | | ADAMAOUA |
| 40 | CM200400000040 | CM | 2004 | 40 | CM | CM04 | Est | Est | CMR003 | Haut-Nyong | CMR003002 | 4 | | EST |
| 41 | CM200400000041 | CM | 2004 | 41 | CM | CM12 | Extreme-Nord | Extreme-Nord | CMR004 | Mayo-Tsanaga | CMR004006 | 5 | | EXTREME-NOR |
| 42 | CM200400000042 | CM | 2004 | 42 | CM | CM14 | Sud | Sud | CMR009 | Dja-et-Lobo | CMR009001 | 10 | | SUD |
| 43 | CM200400000043 | CM | 2004 | 43 | CM | CM04 | Est | Est | CMR003 | Haut-Nyong | CMR003002 | 4 | | EST |
| 44 | CM200400000044 | CM | 2004 | 44 | CM | CM12 | Extreme-Nord | Extreme-Nord | CMR004 | Logone-et-Chari | CMR004002 | 5 | | EXTREME-NOR |
| 45 | CM200400000045 | CM | 2004 | 45 | CM | CM11 | Centre | Centre | CMR002 | Nyong-et-So | CMR002010 | 2 | | CENTRE |
| 46 | CM200400000046 | CM | 2004 | 46 | CM | CM14 | Sud | Sud | CMR009 | Ocean | CMR009003 | 10 | | SUD |
| 47 | CM200400000047 | CM | 2004 | 47 | CM | CM12 | Extreme-Nord | Extreme-Nord | CMR004 | Mayo-Danay | CMR004003 | 5 | | EXTREME-NOR |
| 48 | CM200400000048 | CM | 2004 | 48 | CM | CM11 | Centre | Centre | CMR002 | Lekie | CMR002002 | 2 | | CENTRE |
| 49 | CM200400000049 | CM | 2004 | 49 | CM | CM14 | Sud | Sud | CMR009 | Mvila | CMR009002 | 10 | | SUD |
| 50 | CM200400000050 | CM | 2004 | 50 | CM | CM05 | Littoral | Littoral | CMR005 | Wouri | CMR005004 | 3 | | DOUALA |
| 51 | CM200400000051 | CM | 2004 | 51 | CM | CM04 | Est | Est | CMR003 | Haut-Nyong | CMR003002 | 4 | | EST |
| 52 | CM200400000052 | CM | 2004 | 52 | CM | CM11 | Centre | Centre | CMR002 | Nyong-et-Mfoumou | CMR002009 | 2 | | CENTRE |
| 53 | CM200400000053 | CM | 2004 | 53 | CM | CM08 | Ouest | Ouest | CMR008 | Hauts-Plateaux | CMR008003 | 9 | | OUEST |

Une fois les variables identifiées, fermez l'éditeur pour que R puisse reprendre la main et vous inviter à saisir les variables correspondantes. Faites attention, les noms des variables peuvent différer d'un pays à l'autre.

```
Veuillez indiquez les variables suivantes :

Numéro des clusters :

 1: DHSID          2: DHSCC          3: DHSYEAR        4: CLUSTER        5: CCFIPS         6: ADM1FIPS
 7: ADM1FIPSNA     8: ADM1SALBNA     9: ADM1SALBCO    10: ADM2SALBNA    11: ADM2SALBCO    12: ADM1CODE
13: ADM1DHS       14: ADM1NAME      15: ADM2CODE      16: ADM2DHS       17: ADM2NAME      18: ADM3CODE
19: ADM3DHS       20: ADM3NAME      21: ADM4CODE      22: ADM4DHS       23: ADM4NAME      24: PPLNAME
25: PPLCODE       26: REPAR1DHS     27: REPAR1NAME    28: REPAR2DHS     29: REPAR2NAME    30: SOURCE
31: LOCAL_LVL     32: U.R           33: LATNUM        34: LATDEG        35: LATMIN        36: LATSEC
37: LATTHOU       38: LATHEMI       39: LONGNUM       40: LONGDEG       41: LONGMIN       42: LONGSEC
43: LONGTHOU      44: LONGHEMI      45: UTMLAT        46: UTMLONG       47: UTMZONE       48: ALT_GPS
49: ALT_DEM       50: DATUM         51: SYMBOL        52: WAF_ID

Sélection : 4

Longitude (valeur décimale) :
Sélection : 39

Latitude (valeur décimale) :
Sélection : 33

Milieu de résidence :
Sélection : 32

Code numérique des régions (0 si non renseigné) :
Sélection : 12

Nom des régions (0 si non renseigné) :
```

```
Sélection : 14
```

Vérification des informations saisies :

```
ATTENTION : veuillez vérifier les informations suivantes :

* Numéro de cluster :
CLUSTER
* Longitude (valeur decimale) :
LONGNUM
* Latitude (valeur décimale) :
LATNUM
* Milieu de résidence :
U.R
* Code numérique des régions :
ADM1CODE
* Nom des régions :
ADM1NAME

--------------------------------------------------

Ces données sont-elles correctes ?

1: Oui
2: Non
```

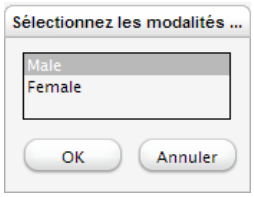
```
Sélection : 1
```

Si la variable *sexe* est présente dans le fichier individu, il est possible de restreindre l'analyse à l'une des modalités de cette variable.

```
La variable sexe a été détectée dans le fichier ind.
Voici ses modalités :
- Male
- Female

Voulez-vous restreindre l'analyse à l'une ou plusieurs de ces modalités ?

1: Oui
2: Non
```

```
Sélection : 2
```

Si vous choisissez de restreindre l'analyse à une ou plusieurs modalités de cette variable, une fenêtre telle que celle ci-dessous s'ouvrira vous permettant de restreindre l'analyse à une ou plusieurs modalités (utilisez la touche CTRL pour sélectionner plusieurs modalités.

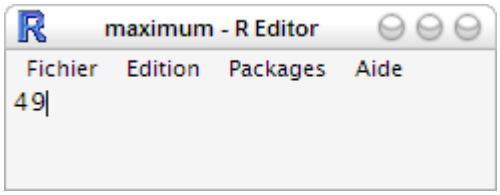


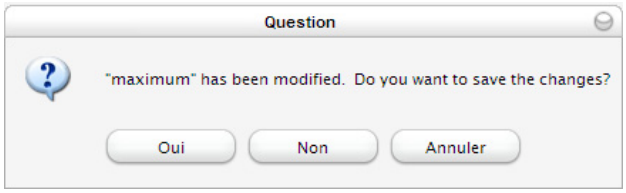

De même, si la variable *age* est présente dans le fichier individu, il est possible de restreindre l'analyse à une classe d'âges donnée. Dans le cas présent, comme de 50 à 59 ans seuls des hommes ont été testés, nous allons restreindre notre analyse aux 15-49 ans.

```
La variable âge a été détectée dans le fichier ind.
Voici ses caractéristiques :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  15.00   20.00   26.00   28.78   36.00   59.00

Voulez-vous restreindre l'analyse sur un intervalle de cette variable ?

1: Oui
2: Non
```

**Sélection : 1**

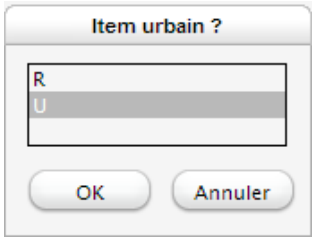
```
Deux fenêtre vont s'ouvrir. Modifiez la valeur minimum à garder, fermez la fenêtre, modifiez la
valeur maximum, fermez la fenêtre.
```




**Astuce :** si l'on souhaite restreindre l'analyse à d'autres variables que les variables sexe et âge, il suffit, au moment de la création du fichier individu, de désigner d'autres variables à la place du sexe et/ou de l'âge.

À l'étape suivante, vous serez invité à spécifier les items urbain et rural de la variable milieu de résidence.




```
Une fenêtre va s'ouvrir. Veuillez spécifier l'item urbain et l'item rural.
```

Quelques informations récapitulatives sont affichées à la fin de la création du tableau de données.

```
L'analyse a été restreinte aux personnes âgées de  15 - 49  ans.
Statistiques du fichier :
* 466 clusters.
* 9900 observations valides.
* Prévalence globale de 5.51%.
* Valeur de Noptimal proposée : 363
```

Vous pouvez avoir un aperçu du contenu du tableau de données obtenu à l'aide de str() et de summary().

```
> str(cm.clust)
'data.frame':   466 obs. of  9 variables:
 $ cluster       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ x             : num   9.72 13.53 15.23 14.58 10.30 ...
 $ y             : num   4.04  9.10 10.33 12.77  4.52 ...
 $ residence     : Factor w/ 2 levels "Rural","Urban": 2 1 2 1 2 2 2 1 2 1 ...
 $ region        : num  3 7 5 5 6 12 5 9 12 5 ...
 $ region.name   : Factor w/ 12 levels "ADAMAOUA","CENTRE",..: 3 7 5 5 6 12 5 9 12 5 ...
 $ n             : num  9 26 31 22 10 4 9 17 16 22 ...
 $ nweight       : num  10.79 24.15 62.14 34.87  6.59 ...
 $ obs.prevalence: num  0.00 0.00 3.13 0.00 0.00 ...
```

```
> summary(cm.clust)
    cluster           x               y          residence       region
 Min.   :  1.0   Min.   : 9.025   Min.   : 2.270   Rural:222   Min.   : 1.000
 1st Qu.:117.3   1st Qu.: 9.988   1st Qu.: 4.009   Urban:244   1st Qu.: 3.000
 Median :233.5   Median :11.333   Median : 4.734               Median : 6.000
 Mean   :233.5   Mean   :11.595   Mean   : 5.581               Mean   : 6.534
 3rd Qu.:349.8   3rd Qu.:13.367   3rd Qu.: 6.425               3rd Qu.: 9.750
 Max.   :466.0   Max.   :15.446   Max.   :12.771               Max.   :12.000

      region.name        n           nweight      obs.prevalence
 DOUALA      : 46   Min.   : 2.00   Min.   : 2.335   Min.   : 0.00
 YAOUNDE     : 45   1st Qu.:14.00   1st Qu.:11.710   1st Qu.: 0.00
 EXTREME-NOR: 41   Median :20.00   Median :19.074   Median : 4.26
 OUEST       : 40   Mean   :21.24   Mean   :21.234   Mean   : 5.79
 NORD-OUEST : 39   3rd Qu.:27.00   3rd Qu.:28.353   3rd Qu.: 8.87
 CENTRE      : 38   Max.   :57.00   Max.   :94.920   Max.   :45.53
 (Other)     :217
```

Pour information, les individus ayant un résultat indéterminé ou une pondération nulle sont considérés comme manquant et ne sont donc pas comptabilisés pour le calcul de *n*, *nweight* et *obs.prevalence*.[4]

Pour le détail des différentes variables, voir l'aide de make.clust.dbf :

```
> ?make.clust.dbf
```

### 4.2.3 Création du fichier villes

La fonction make.cities.csv() permet de générer un tableau de données des principales villes du pays à partir d'un fichier *csv* tel que celui récupéré sur le GRUMP.

Cette fonction agit de la même manière que la précédente. Tout d'abord le contenu du fichier *csv* est affiché dans un éditeur, puis vous êtes invité à saisir les variables demandées.

```
cm.cities <- make.cities.csv('cm.cities.csv',lang='fr')
```

```
Une fenêtre va s'ouvrir présentant les données contenues dans le fichier. Merci de repérer les
variables suivantes (toutes nécessaires) :
- Nom des villes
- Longitude (en degrés au format décimal)
- Latitude (en degrés au format décimal)
- Population (effectif)
 Une fois les noms de ces variables identifiés, fermez la fenêtre pour que le programme puisse
continuer.

 Êtes-vous prêt ?

1: Oui
```

```
Sélection : 1
```

```
-----------------------------------------------------

Veuillez indiquez les variables suivantes :

Nom des villes :

 1: CONTINENT    2: UNREGION     3: COUNTRY      4: UNSD         5: ISO3
 6: UQID         7: SCHNM        8: SCHADMNM     9: LATITUDE    10: LONGITUDE
11: TYPE        12: POP         13: YEAR        14: URBORRUR    15: ES90POP
16: ES95POP     17: ES00POP     18: POPSRC      19: SRCTYP      20: LOCNDATSRC
21: COORDSRCE
```

[4] Dans la suite de ce tutoriel nous appellerons observations valides les individus dont le résultat est déterminé (positif ou négatif) et dont la pondération est non nulle. Le nombre d'observations valides d'un cluster correspond donc à la variable *n*.

```
Sélection : 7

Longitude (valeur décimale) :
Sélection : 10

Latitude (valeur décimale) :
Sélection : 9

Population des villes :
Sélection : 17

ATTENTION : veuillez vérifier les informations suivantes :

* Nom des villes :
SCHNM
* Longitude (valeur decimale) :
LONGITUDE
* Latitude (valeur décimale) :
LATITUDE
* Population of cities :
ES00POP

-----------------------------------------------------

Ces données sont-elles correctes ?

1: Oui
2: Non

Sélection : 1
```

La structure du fichier créé est obtenue à l'aide de str().

```
> str(cm.cities)
'data.frame':   166 obs. of  4 variables:
 $ city.name : Factor w/ 166 levels "ABONGMBANG","AKO",..: 48 162 65 95 11 14 135 127 83 61 ...
 $ x         : num   9.7 11.5 13.4 14.3 10.4 ...
 $ y         : num   4.05  3.87  9.30 10.60  5.47 ...
 $ population: int  1512379 1213902 265294 230353 210707 206552 159663 145942 131145 107075 ...
```

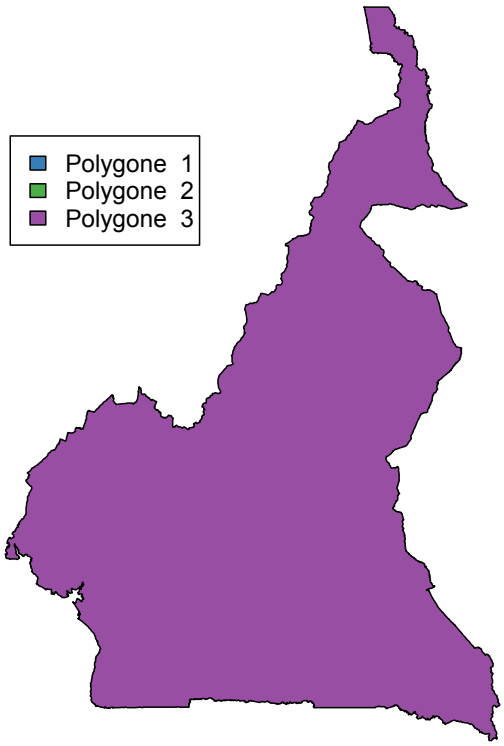
### 4.2.4 Création du fichier frontières du pays

Vous pouvez créer votre propre tableau de données à partir des fonctions d'importation et de manipulation des données de R. prevR utilise les frontières de la zone d'enquête sous la forme d'un *data.frame* (tableau de données) à deux colonnes ($x$ et $y$) représentant une liste de points formant un polygone fermé.

Il est possible d'importer un fichier de points au format texte fourni par le Digital Chart of the World (DCW) à l'aide de la fonction make.boundary.dcw(). Le temps d'exécution de cette fonction peut prendre quelques minutes selon la puissance de votre ordinateur. Un indicateur de progression s'affiche sous forme de messages dans R.

Certains pays sont décrits par plusieurs polygones. Par exemple, la France métropolitaine sera définie par un polygone dessinant son territoire continental, un polygone formant la Corse et plusieurs petits polygones correspondant aux petites îles situées le long de la côte. prevR n'est capable de tenir compte que d'un seul polygone. Si plusieurs polygones sont détectés dans le fichier du DCW, vous êtes invité à choisir quel polygone sera utilisé (les polygones détectés étant affichés sous forme graphique).

```
> cm.bounds <- make.boundary.dcw('cameroon2pts.txt',lang='fr')
100 sur 4411 points traités.
200 sur 4411 points traités.
....
4400 sur 4411 points traités.

-------------------------------------------------------
Longitude maximale observée dans le fichier : 16.192116
Longitude minimale observée dans le fichier : 8.494763
Latitude maximale observée dans le fichier : 13.078056
Latitude minimale observée dans le fichier : 1.652548
-------------------------------------------------------

Ce fichier contient 3 polygones.
Une nouvelle fenêtre vous montre les différents polygones
contenus dans le fichier et leur numéro.
ATTENTION : certains polygones sont invisibles à l'oeil nu
(petites îles par exemple).
Veuillez sélection le polygone principal qui sera utilisé
comme limites du pays.

1: Polygone  1
2: Polygone  2
3: Polygone  3

Sélection : 3


> str(cm.bounds)
'data.frame':   4380 obs. of  2 variables:
 $ x: num  14.2 14.2 14.2 14.2 14.2 ...
 $ y: num  12.5 12.5 12.5 12.5 12.5 ...
```




**NB :** nous vous conseillons de sauver vos données à l'aide de la commande *Fichier* > *Sauver environnement de travail*... ou bien à partir des fonctions save() et save.image().

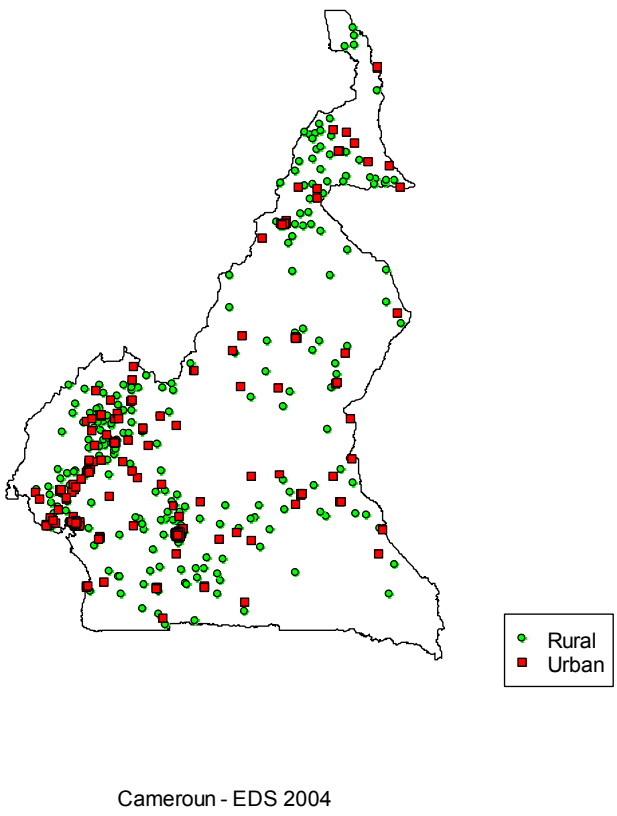
# 5. Cartographier le contenu du fichier cluster

Plusieurs représentations graphiques du fichier cluster peuvent être effectuées à l'aide de la fonction map.clust().

## 5.1 Afficher les clusters par milieu de résidence

Il suffit de préciser à la fonction map.clust() le nom du *data.frame* des clusters et celui correspondant aux frontières du pays. Un titre peut être précisé avec *main* et un sous-titre avec *sub*. D'autres options sont possibles (voir l'aide de map.clust() et celle de title()).

```
> map.clust(cm.clust,cm.bounds, lang='fr', main='Clusters par milieu de
résidence',sub='Cameroun - EDS 2004')
```




Le graphique obtenu peut être facilement exporter aux formats *emf, postscript, pdf, png, bmp* et *jpg* à l'aide du menu *Fichier* > *Sauver sous...* de la fenêtre graphique.

Pour un export au format *SVG* pour importation ultérieure dans un logiciel de dessin vectoriel (Inkscape ou Illustrator par exemple), voir l'annexe 1.

## 5.2 Afficher le nombre d'observations valides par cluster

Le nombre d'observations valides d'un cluster (variable *n*) correspond aux nombres d'observations avec un résultat déterminé (positif ou négatif) et une pondération non nulle (voir note 4 page 20).

Pour cela il faut modifier la variable *type* et lui attribuer la valeur *'count'*. Au passage, notez l'ajout du caractère \ devant le caractère ' dans le titre du graphique pour lui indiquer que cette apostrophe ne marque pas la fin du texte mais doit être affichée.

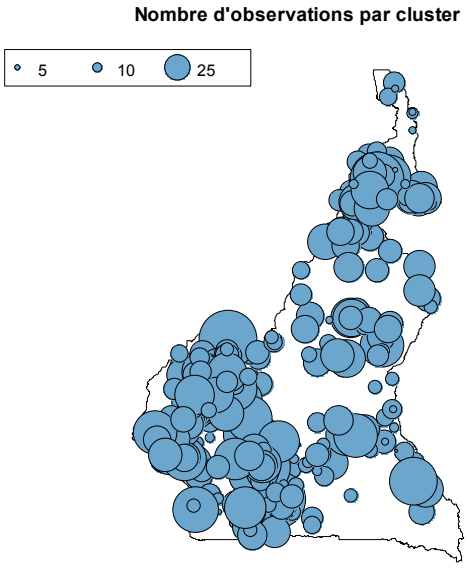
```
> map.clust(
      cm.clust,
      cm.bounds,
      type='count',
      lang='fr',
      main='Nombre d\'observations par cluster',
      sub='Cameroun - EDS 2004 - factor.size=0.2'
)
```


Cameroun - EDS 2004 - factor.size=0.2

Afin d'améliorer la lisibilité de la carte, il est possible de faire varier la taille des cercles à l'aide du paramètre *factor.size*. Sa valeur par défaut est de 0,2. Pour réduire la taille des cercles, on peut le passer à 0,15. La taille des cercles de la légende est également modifiée en conséquence.

La position de la légende peut être changée à l'aide du paramètre *legend.location*.

```
> map.clust(
      cm.clust,
      cm.bounds,
      type='count',
      lang='fr',
      factor.size=0.15,
      main='Nombre d\'observations par cluster',
      sub='Cameroun - EDS 2004 - factor.size=0.15',
      legend.location='topleft'
)
```




Cameroun - EDS 2004 - factor.size=0.15

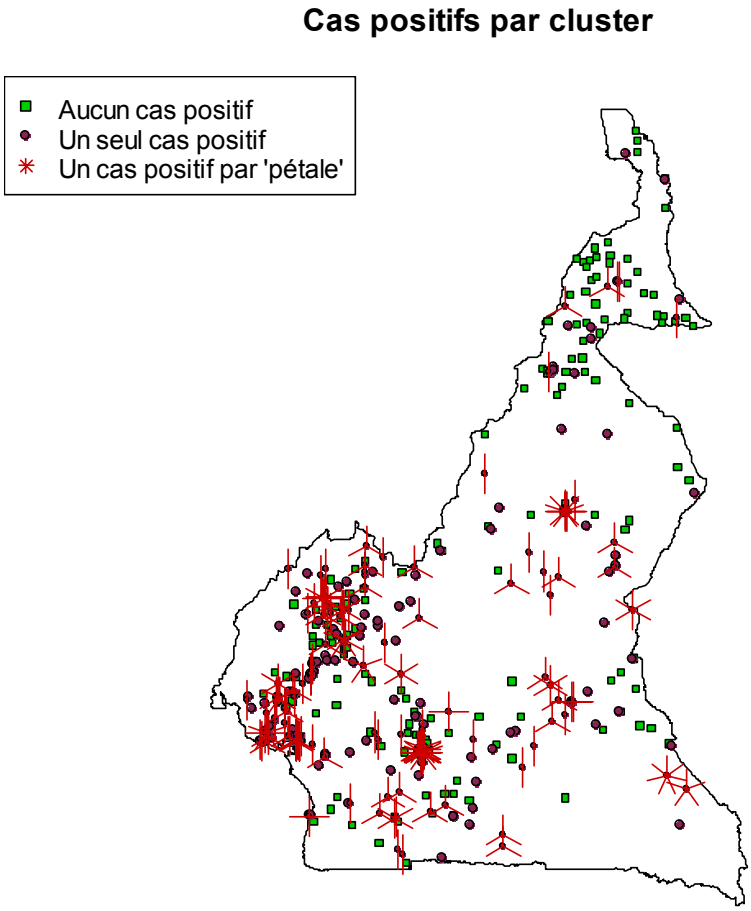
Pour exporter cette carte au format SVG puis lui appliquer une transparence avec le logiciel Inkscape, voir les annexes 1 et 2. Le fait d'appliquer une transparence aux cercles rend la lecture de la carte plus aisée et intuitive.

## 5.3 Représenter le nombre de cas positifs par cluster

Cette représentation graphique repose sur la fonction sunflowerplot(). Un cluster sans cas positif sera représenté par un point vert, un cluster avec un seul cas positif par un point mauve et un cluster avec plusieurs cas positifs par un point mauve et autant de « pétales » rouges que de cas positifs.

Pour cela, il suffit d'appeler map.clust() avec *type = 'flower'*.

```
> map.clust(
      cm.clust,
      cm.bounds,
      type='flower',
      lang='fr',
      main='Cas positifs par cluster',
      sub='Cameroun - EDS 2004',
      legend.location='topleft'
)
```



# 6. La méthode des cercles

Dans les enquêtes de type EDS, le nombre de personnes enquêtées par cluster est faible (entre 10 et 40 le plus souvent). Il en résulte que les prévalences observées dans chaque cluster (calculées à partir des personnes enquêtées de chaque cluster) varient fortement et reflètent les aléas de l'échantillonnage[5]. Pour être porteuse de sens, une prévalence nécessite d'être calculée sur un nombre suffisants d'individus, condition non remplie concernant les prévalences observées dans chaque cluster. Une interpolation spatiale réalisée à partir des prévalences observées n'apporte que peu ou pas d'information, les techniques d'interpolation spatiale classique présupposant une mesure relativement fine du phénomène étudié en chacun des points connus.

**Note sur les Interpolations spatiales**

Les méthodes d'interpolation spatiale permettent d'estimer les valeurs d'une variable en des points où elle n'est pas connue à partir des valeurs de cette variable aux points observés. Plus précisément, l'interpolation est la procédure qui consiste à estimer la valeur d'attribut pour des sites non échantillonnés situés à l'intérieur des limites définies par les positions de sites échantillonnés. L'extrapolation est la procédure qui consiste à estimer la valeur d'attribut pour des sites non échantillonnés situés à l'extérieur des limites définies par les positions des sites échantillonnés

**Pondération selon l'inverse de la distance :** il s'agit d'une méthode de moyenne pondérée ou chaque valeur de la grille à interpoler est calculée comme une moyenne pondérée des observations. Les facteurs de pondérations sont calculés proportionnellement à l'inverse de la distance élevée à une puissance. Cette méthode permet d'obtenir des grilles très rapidement mais crée des zones circulaires autour des valeurs observées (bull'eyes). Cet aspect peut être lissé en jouant sur la puissance et le voisinage. C'est un interpolateur exact (il passe par les valeurs observées).
Le **Krigeage** est une interpolation qui estime les valeurs aux points non échantillonnés par une combinaison des données. Les poids des échantillons sont pondérés par une fonction de structure qui est issue des données. On tient ainsi compte des distances, des valeurs et des corrélations. La fonction n'est pas fixée à priori mais suite à l'analyse du variogramme. On considère que la valeur estimée en un point est le produit d'un processus sous-jacent, il fournit une variance d'estimation contrairement aux autres approches. Elle permet d'appréhender la structure spatiale du phénomène étudié. Le Krigeage s'inscrit donc dans une démarche d'analyse des données géostatistique.

Les textes ci-dessus sont extraits des documents ci-dessous dont nous vous conseillons la lecture pour plus de détails :

- Cours sur l'*interpolation spatiale* de l'Université de Montréal (www.geog.umontreal.ca/donnees/geo2512/geo2512cours10.ppt).
- *Statistiques et Interpolations dans les SIG*, Laurent DRAPEAU, Centre I.RD Montpellier, Laboratoire HEA (www.faocopemed.org/vldocs/0000028/publi10.pdf).
- *Le Krigeage : la méthode optimale d'interpolation spatiale*, Yves Gratton, Institut d'Analyse Géographique (www.iag.asso.fr/pdf/krigeage_juillet2002.pdf).
- *Spatial analysis*, Wikipédia (http://en.wikipedia.org/wiki/Spatial_analysis).

Les techniques d'analyse spatiale décomposent les variations d'une variable *Z* en l'addition d'une tendance régionale *TR* et de résidus locaux *RL*. Dans le cadre des EDS, nous devons rajouter un troisième terme, l'erreur aléatoire *EA* due à l'échantillonnage.

[5] Dans la majorité des cas, les intervalles de confiance des prévalences observées sont tellement larges qu'il devient impossible de tirer la moindre conclusion.

La prévalence observée n'étant pas utilisable, l'approche développée dans prevR consiste donc à estimer une prévalence pour chaque cluster en ayant recours à une méthode dite *des cercles*. Cette dernière permet d'estimer une tendance régionale pour chaque grappe ou cluster. Dans un second temps, une fois la prévalence de chaque cluster correctement estimée, il devient possible d'appliquer des techniques classiques d'interpolation spatiale, notamment le krigeage ordinaire.

Outre une présentation succincte de la méthodologie des cercles dans le présent document, nous vous renvoyons au document suivant pour une présentation détaillée :

> Joseph Larmarange, *Prévalences du VIH en Afrique : validité d'une mesure*, chapitre 4, thèse de doctorat en démographie, sous la direction de Benoît Ferry, Université Paris Descartes, 2007, disponible sur http://joseph.larmarange.net.

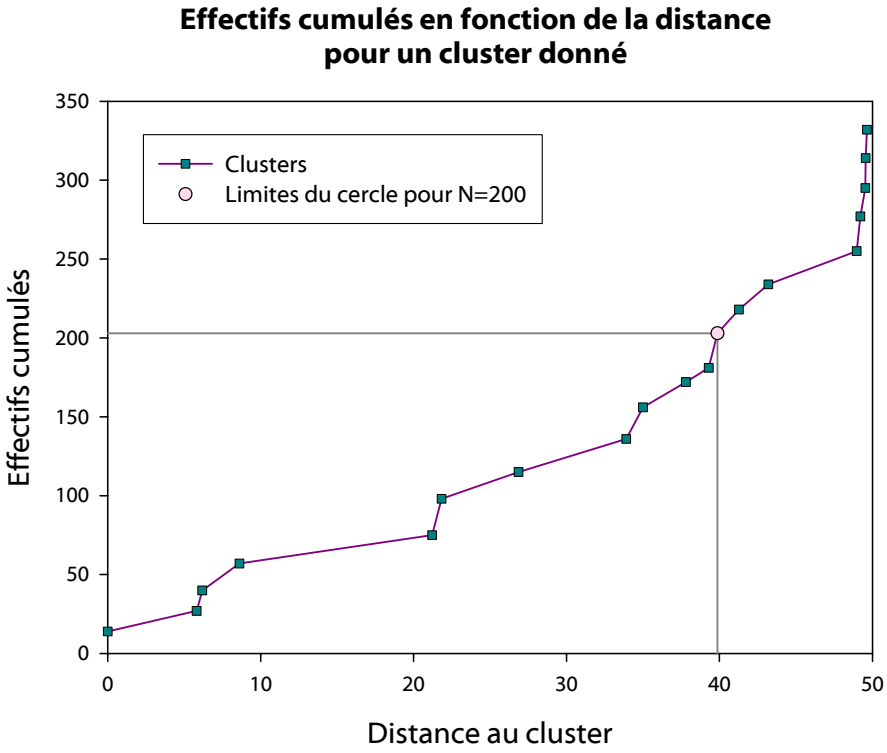
## 6.1 Recours à des cercles de même effectif : le paramètre *N*

S'inspirant de techniques de lissage utilisées pour le calcul de tendances régionales, l'approche de prevR consiste à tracer autour de chaque zone d'enquête un cercle puis à estimer la prévalence de cette zone à partir de l'ensemble des clusters situés à l'intérieur dudit cercle.

Plusieurs techniques de lissage utilisent des cercles de même rayon. Cependant, dans le cadre d'enquêtes de type EDS, il s'avère que les zones d'enquêtes ne sont pas uniformément réparties sur le territoire. Au contraire, leur maillage est dépendant de la densité de la population étudiée sur le territoire. Ainsi, le recours à des cercles de même rayon induirait que les prévalences seraient estimées sur un tout petit nombre d'observations dans les zones les moins peuplées et sur un très grand nombre dans les zones plus denses. Les prévalences estimées resteraient alors fortement aléatoires dans les zones peu denses, tandis que, dans les zones peuplées, il y aurait un effet d'uniformisation et une perte d'information.

Dans la mesure où c'est le nombre d'observations qui donne sens aux prévalences, l'approche de prevR privilégie le recours à des cercles de même effectif. Une fois posé un effectif *N* donné, le rayon des cercles pour chaque cluster est calculé de manière à ce que le nombre d'observations situées à l'intérieur du dit cercle soit au moins égal à *N*.

Le graphique ci-après montre, pour un cluster *x* donné, la répartition des clusters en fonction de leur distance au cluster *x* et du nombre cumulé d'observations. Si l'on a choisi une valeur de 200 pour le paramètre *N*, la prévalence du cluster *x* sera calculée sur les observations de l'ensemble des clusters situés à moins de 40 kilomètres de *x* (*x* inclus) soit sur 203 observations. Cette distance de 40 kilomètres correspond à la distance du premier cluster tel que l'effectif cumulé des observations soit supérieur à 200.




Le recours à un effectif minimum comme paramètre pour déterminer la taille des cercles permet à la fois de s'assurer que la prévalence de chaque cluster sera estimée sur un nombre d'observations suffisant et d'appliquer un niveau de lissage différent selon les zones d'enquêtes, le niveau de lissage étant déterminé par la superficie du cercle.

Lorsque l'on augmente progressivement la valeur du paramètre *N*, les prévalences estimées sont calculées sur un nombre d'observations plus important et sont lissées, atténuant ainsi les variations aléatoires de l'échantillon. Dans le même temps, elles tendent progressivement vers une valeur unique (effet d'uniformisation). Il s'agit alors de déterminer un compromis minimisant suffisamment les aléas de l'échantillonnage tout en conservant une précision locale.

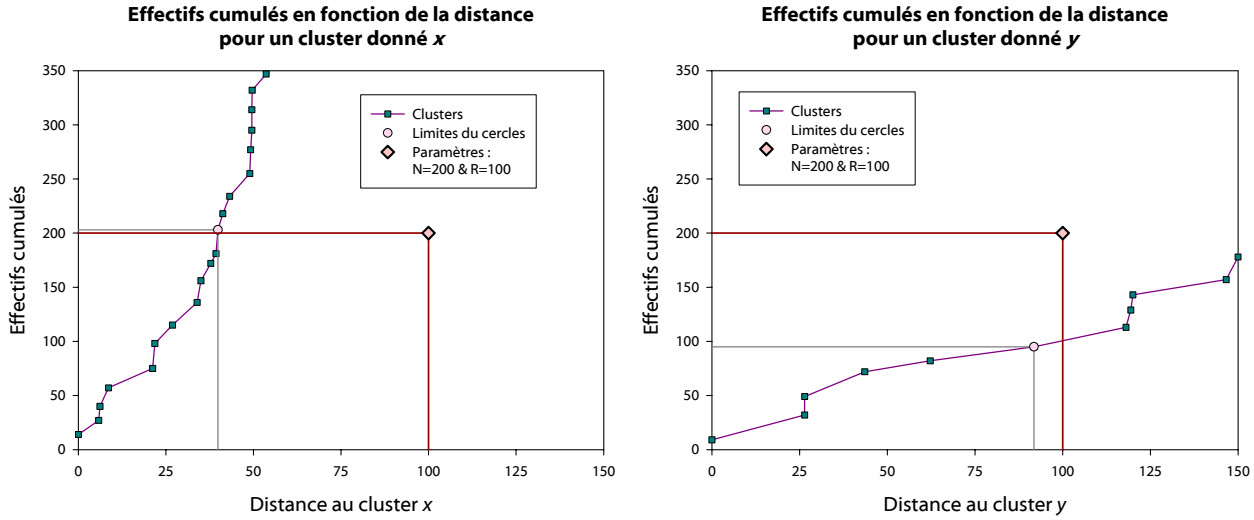
À cette fin, des modélisations et simulations d'enquêtes ont été réalisées en grand nombre. À partir de 24 500 simulations, nous avons montré qu'il était possible de déterminer une valeur optimale pour le paramètre *N* qui variait en fonction de la prévalence globale, du nombre total de personnes enquêtées et, dans une moindre mesure, du nombre de clusters.

Cette valeur peut être calculée à partir des fonctions infos.prev() et N.optim(). Elle est valable pour des enquêtes portant sur au moins 5 000 individus, 7 000 si la prévalence globale est inférieure à 2 %, 8 500 si la prévalence globale est inférieure à 1 %.

## 6.2 Ajout d'un rayon maximum : le paramètre *R*

Dans les zones faiblement enquêtées, notamment le long des frontières, le calcul des prévalences fait intervenir des clusters très éloignés les uns des autres. Il peut être alors préférable de limiter le lissage à des cercles plus petits, quitte à estimer, pour ces clusters là, les prévalences sur un nombre d'observations moindre.

prevR permet de rajouter au paramètre *N* un paramètre noté *R* correspondant à un rayon maximum des cercles de lissage. Deux situations peuvent alors se présenter, illustrées par les deux exemples ci-après correspondant à des valeurs de *N* et de *R* respectivement de 200 observations et de 100 kilomètres.




Pour le cluster *x*, l'effectif minimum de 200 observations est atteint pour une distance inférieure à 100 kilomètres. L'estimation de la prévalence restera donc inchangée par rapport à la situation sans le paramètre *R*. Pour le cluster *y* en revanche, très isolé, l'estimation ne sera réalisée que sur les 95 observations situées à moins de 100 kilomètres du cluster.

L'ajout du paramètre *R* servant le plus souvent à limiter la taille des cercles pour les clusters les plus isolés, une valeur adéquate de ce paramètre peut être obtenue en prenant le 9e décile des rayons des cercles de lissage lorsque seul le paramètre *N* est appliqué. Cela peut être calculé directement par la fonction infos.prev().

La technique des cercles de même effectif (paramètre *N* seulement) et celle des cercles de même rayon (paramètre *R* seulement) constituent deux cas particuliers de l'application conjointe des paramètres *N* et *R*, respectivement en attribuant la valeur *Infinie* à *R* ou à *N*.

**Conseil :** nous vous conseillons de produire une carte avec seulement le paramètre *N* puis avec l'utilisation conjointe de *N* et de *R*. Ensuite, aidez vous de la carte de type *flower* (voir 5.3) pour décider de retenir ou non le paramètre R.

L'ajout éventuel du paramètre R permet simplement de tester deux hypothèses concernant les zones faiblement enquêtées. En l'absence de ce paramètre, la prévalence de ces régions est fortement lissée à partir des valeurs des régions voisines. Lorsque R est appliquée, elle et plus fortement dépendante des quelques observations effectuées localement. En raison de la faible quantité de données dans ces régions, nous ne pourrons interpréter les résultats estimées dans ces zones de manière précise. Lorsque R sera appliqué, nous ne pourrons déterminer si les variations locales que nous observerons correspondent à une réalité des variations locales de l'épidémie dans ces régions ou bien s'il s'agit de variations aléatoires dues à l'échantillonnage. Cependant, cela pourra constituer des pistes de recherche à investiguer.

### 6.3 Prise en compte des agglomérations urbaines : le paramètre *U*

De nombreux phénomènes présentent des différentiels marqués selon le milieu de résidence. Il est possible d'observer des diffusions progressives d'une ville vers son voisinage ou bien encore la concentration d'un phénomène sur une agglomération donnée.

prevR peut prendre en compte des agglomérations urbaines pour le calcul des prévalences. Si c'est le cas, la prévalence d'un cluster situé hors agglomération ne sera calculée qu'à partir de clusters situés également hors agglomération. De manière générale, la taille des cercles sera un peu plus grande dans la mesure où les clusters appartenant à une agglomération urbaine n'auront pas été pris en compte pour calculer la taille du cercle de lissage.

Pour les clusters appartenant à une agglomération urbaine, seuls des clusters appartenant à la **même** agglomération seront pris en compte pour l'estimation des prévalences. Il importe donc que les agglomérations urbaines retenues pour l'analyse aient été suffisamment enquêtées pour que leur nombre d'observations ne soit pas trop faible. Différents critères pour retenir une agglomération urbaine dans l'analyse seront présentés dans la section 7.

Afin de pouvoir distinguer deux estimations réalisées avec des sélections différentes d'agglomérations urbaines, nous appelons *U* le nombre d'agglomérations urbaines retenues dans une estimation. La non prise en compte des agglomérations urbaines correspondra donc au cas *U*=0.

## 7. Estimer la prévalence de chaque cluster

La fonction permettant d'estimer la prévalence de chaque cluster par la méthode des cercles est estimate.prev(). La lecture de la documentation de cette fonction est fortement conseillée.

```
> ?estimate.prev
```

### 7.1 Choisir les paramètres *N* et *R*

Une valeur optimale du paramètre *N* peut être obtenue à partir d'une modélisation effectuée sur 14.000 simulations d'enquêtes type EDS (voir 6.1). Cette valeur est fournie par la fonction infos.prev().

```
> infos.prev(cm.clust, lang='fr')
Statistiques du fichier :
* 466 clusters.
* 9900 observations valides.
* Prévalence globale de 5.51%.
* Valeur de Noptimal proposée : 363
```

Vous pouvez utiliser la valeur proposée, ou bien choisir votre propre valeur.

**Approche complémentaire :**

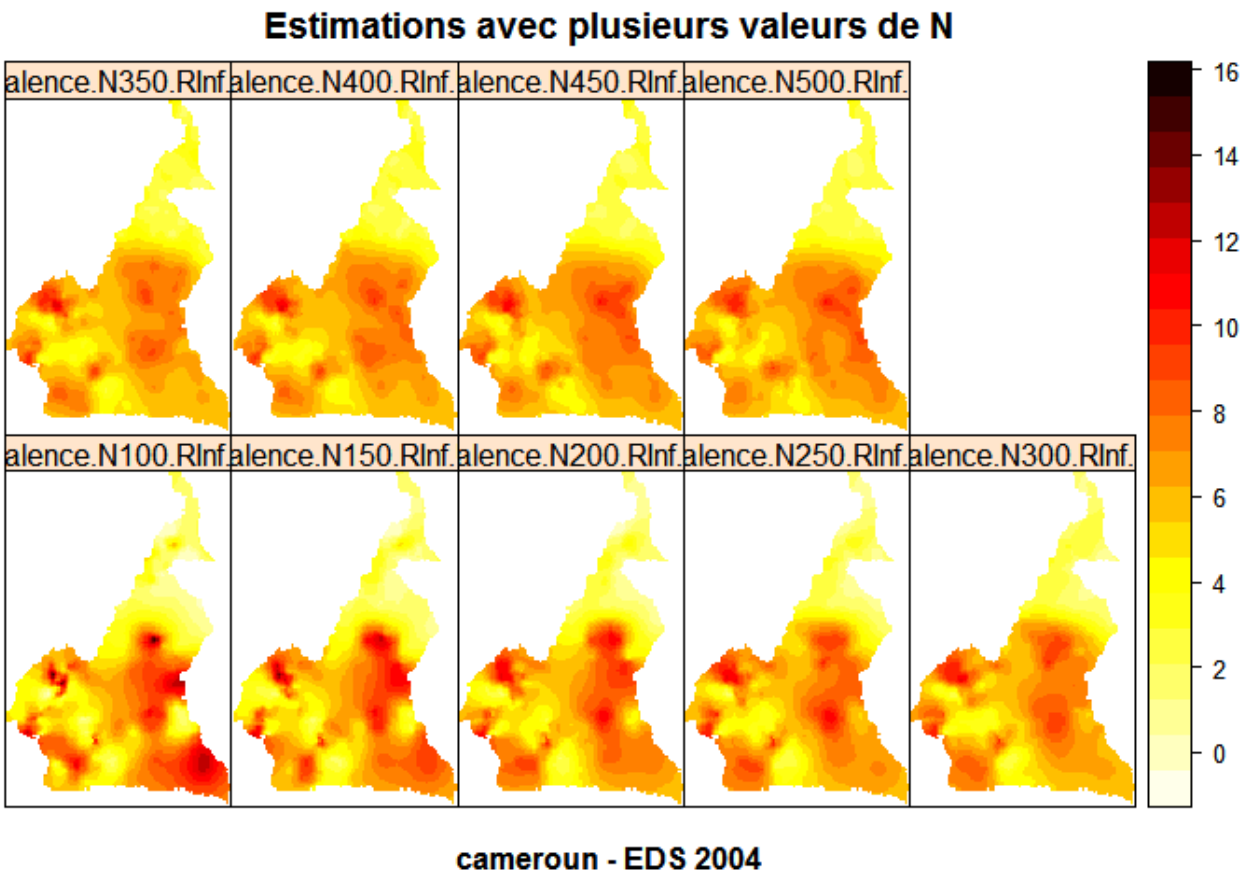
*Avant de lire ce qui suit, nous vous conseillons de lire au préalable la fin de la présente section ainsi que la section suivante consacrée à l'interpolation spatiale de la prévalence.*

Une possibilité consiste à réaliser plusieurs estimations avec des valeurs croissantes de *N* (par exemple faire varier *N* de 100 à 500 par pas de 50) puis à cartographier les différentes estimations réalisées pour faire son choix. La liste des valeurs de 100 à 500 par pas de 50 peut être obtenue à l'aide la fonction seq(). Par défaut, c'est-à-dire en l'absence de spécification par l'utilisateur, estimate.prev() ne tient pas compte des paramètres *R* et *U*.

```
> cm.prev.plusieursN <- estimate.prev(
      cm.clust,
      N=seq(100, 500, 50),
      lang='fr',
      merge.result = TRUE
)
> cm.krige.plusieursN <- krige.prev(
      cm.prev.plusieursN,
      c(
            est.prevalence.N100.RInf.U0~1,
            est.prevalence.N150.RInf.U0~1,
            est.prevalence.N200.RInf.U0~1,
            est.prevalence.N250.RInf.U0~1,
            est.prevalence.N300.RInf.U0~1,
            est.prevalence.N350.RInf.U0~1,
            est.prevalence.N400.RInf.U0~1,
            est.prevalence.N450.RInf.U0~1,
            est.prevalence.N500.RInf.U0~1),
      boundary = cm.bounds,
      lang = 'fr'
)
> spplot(
      cm.krige.plusieursN,
      zcol = c(
            'est.prevalence.N100.RInf.U0.pred',
            'est.prevalence.N150.RInf.U0.pred',
            'est.prevalence.N200.RInf.U0.pred',
            'est.prevalence.N250.RInf.U0.pred',
            'est.prevalence.N300.RInf.U0.pred',
            'est.prevalence.N350.RInf.U0.pred',
            'est.prevalence.N400.RInf.U0.pred',
            'est.prevalence.N450.RInf.U0.pred',
            'est.prevalence.N500.RInf.U0.pred'),
      col.regions = prevR.colors.red(21), cuts=20,
      main='Estimations avec plusieurs valeurs de N', sub='cameroun - EDS 2004'
)
```

Pour des informations sur le fonctionnement de krige.prev() et spplot(), veuillez consulter les sections 8 et 9.

On obtient ainsi la figure suivante :




Vous pouvez vous aidez de la carte de type *flower* (voir 5.3) pour vous aider à faire votre choix ainsi que des connaissances que vous avez sur l'épidémie du pays que vous étudié. Cette approche nécessite donc d'avoir une expertise préalable.

Une fois une valeur de *N* choisie (dans notre exemple nous avons choisi 363, la valeur optimale de N proposée par infos.prev()), la méthode des cercles s'applique de la manière suivante :

```
> cm.prev.N363 <- estimate.prev (cm.clust, N=363, lang='fr')

Début des calculs : 19:00:12
Cluster 25 sur 466 terminé. (19:00:13)
Cluster 50 sur 466 terminé. (19:00:14)
Cluster 75 sur 466 terminé. (19:00:14)
Cluster 100 sur 466 terminé. (19:00:14)
Cluster 125 sur 466 terminé. (19:00:15)
Cluster 150 sur 466 terminé. (19:00:15)
Cluster 175 sur 466 terminé. (19:00:16)
Cluster 200 sur 466 terminé. (19:00:16)
Cluster 225 sur 466 terminé. (19:00:17)
Cluster 250 sur 466 terminé. (19:00:17)
Cluster 275 sur 466 terminé. (19:00:18)
Cluster 300 sur 466 terminé. (19:00:18)
Cluster 325 sur 466 terminé. (19:00:19)
Cluster 350 sur 466 terminé. (19:00:19)
Cluster 375 sur 466 terminé. (19:00:20)
Cluster 400 sur 466 terminé. (19:00:20)
Cluster 425 sur 466 terminé. (19:00:21)
Cluster 450 sur 466 terminé. (19:00:21)

Fin des calculs : 19:00:22
Temp de calcul : 9.38 secs.
```

Lorsque plusieurs estimations sont réalisées simultanément, les temps de calcul peuvent devenir relativement longs. Un indicateur de progression est affiché à l'écran. Il peut être désactivé avec le paramètre progression = FALSE.

Le tableau de données renvoyé est de la forme :

```
> str(cm.prev.N363)
'data.frame':   466 obs. of  17 variables:
 $ cluster           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ x                 : num   9.72 13.53 15.23 14.58 10.30 ...
 $ y                 : num   4.04  9.10 10.33 12.77  4.52 ...
 $ residence         : Factor w/ 2 levels "Rural","Urban": 2 1 2 1 2 2 2 1 2 1 ...
 $ region            : num  3 7 5 5 6 12 5 9 12 5 ...
 $ region.name       : Factor w/ 12 levels "ADAMAOUA","CENTRE",..: 3 7 5 5 6 12 5 9 12 5 ...
 $ n                 : num  9 26 31 22 10 4 9 17 16 22 ...
 $ nweight           : num  10.79 24.15 62.14 34.87  6.59 ...
 $ obs.prevalence    : num  0.00 0.00 3.13 0.00 0.00 ...
 $ est.prevalence    : num  5.46 1.90 2.00 1.83 3.81 ...
 $ circle.count      : num  380 380 380 363 370 374 391 365 371 369 ...
 $ circle.radius     : num    3.11  59.01 102.51 242.89  69.52 ...
 $ circle.nb.clusters: int  17 18 16 21 21 20 17 17 19 16 ...
 $ quality.indicator : num     0.498  178.620  539.037 3096.468  251.253 ...
 $ N.parameter       : num  363 363 363 363 363 363 363 363 363 363 ...
 $ R.parameter       : num  Inf Inf Inf Inf Inf ...
 $ U.parameter       : num  0 0 0 0 0 0 0 0 0 0 ...
```

8 variables ont été ajoutées à cm.clust :

- *est.prevalence* qui correspond à la prévalence estimée par la méthode des cercles ;
- *circle.count* qui correspond au nombre d'observations valides sur lesquelles la prévalence a été calculée ;
- *circle.radius* qui correspond au rayon du cercle de lissage de chaque cluster ;
- *circle.nb.clust* qui correspond au nombre de clusters inclus dans le cercle de lissage ;
- *quality indicator :* il s'agit d'un indicateur de qualité, calculé pour chaque cluster comme le carré de *circle.radius* divisé par la racine de *circle.count*. Une valeur élevée de cet indicateur indique que la prévalence estimée a été calculée en prenant en compte des clusters éloignés les uns des autres et que le nombre d'observations était faible, d'où une estimation incertaine et peu précise. Au contraire, une valeur faible de cet indicateur sera obtenue pour des estimations calculées à partir d'observations suffisantes et proches les unes des autres, d'où une information précise et locale.
- *N.parameter,*
- *R.parameter* et
- *U.parameter.*

La fonction info.prev() renvoie des informations supplémentaires, notamment les quantiles des rayon des cercles de lissage. Si l'on décide d'utiliser la valeur du 9e décile pour choisir le paramètre *R*, on pourra retenir dans cet exemple la valeur de 118 kilomètres.

```
> infos.prev(cm.prev.N363, lang='fr')
Statistiques du fichier :
* 466 clusters.
* 9900 observations valides.
* Prévalence globale de 5.51%.
* Valeur de Noptimal proposée : 363
* Quantiles des rayons des cercles de lissage :
  50%   75%   80%   85%   90%   95%   99%
 50.6  81.3  94.7 102.5 117.8 137.3 205.7
```

Par défaut, les distances sont exprimées en kilomètres. Cependant, il est possible d'obtenir des miles en utilisant le paramètre miles = TRUE. Si les coordonnées des clusters ne sont pas exprimées en degrés décimaux mais dans un autre référentiel, utilisez le paramètre dist.fonction = 'rdist' pour calculer des distances euclidiennes. Les distances renvoyées par estimate.prev() seront alors exprimées dans l'unité du système de coordonnées.

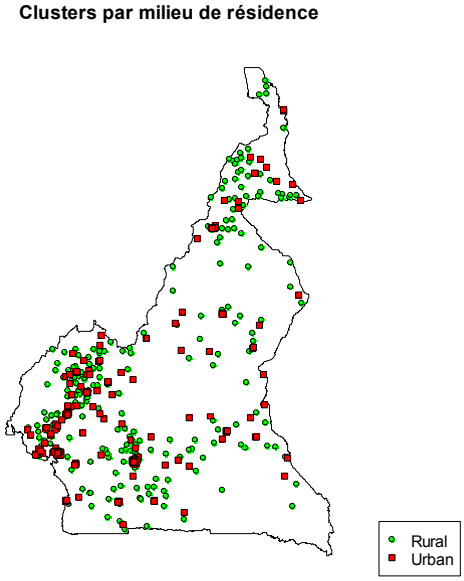
Si les noms des colonnes de votre tableau de données des clusters diffèrent des noms par défaut, vous devrez préciser les noms de vos variables avec var.clust et urban.area.code (voir la documentation de estimate.prev() pour plus de précisions).

## 7.2 Choix des agglomérations urbaines pour le paramètre *U*

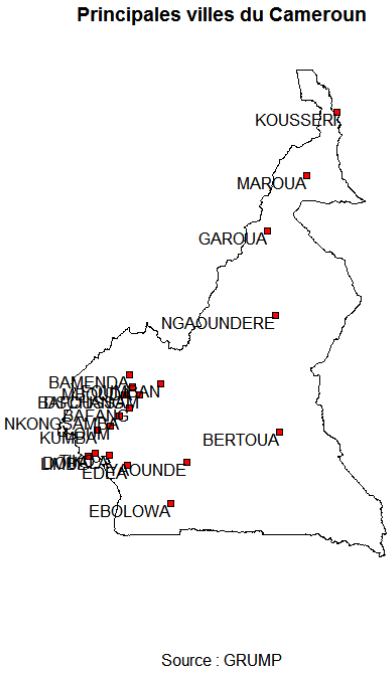
### 7.2.1 Carte des villes et carte des clusters par milieu de résidence

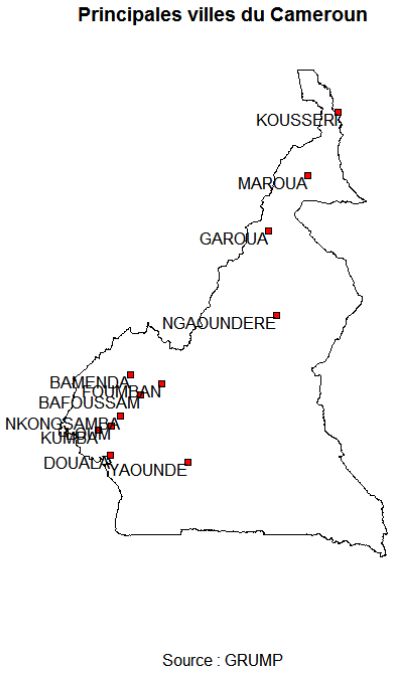
Les fonctions map.clust() et map.cities() permettent, dans un premier temps, de représenter la répartition des clusters par milieu de résidence et la position des principales villes du pays.

```
> map.clust(cm.clust,cm.bounds,main='Clusters par milieu de résidence',sub='Cameroun -
DHS 2004')
```



```
> map.cities(cm.cities, cm.bounds, min.population=65000,
      main='Principales villes du Cameroun', sub='Source : GRUMP')
> map.cities(cm.cities, cm.bounds, min.population=100000,
      main='Principales villes du Cameroun', sub='Source : GRUMP')
```




### 7.2.2 Recoder le milieu de résidence

Dans les variables disponibles dans les EDS, on ne peut savoir si un cluster appartient ou non à une ville donnée. Le milieu de résidence (urbain ou rural) est fourni mais ne peut être utilisé directement. En effet, cette dichotomie repose sur la définition en vigueur dans chaque pays et inclut le plus souvent des communes de taille moyenne et/ou chefs-lieux d'entité administrative.

Le projet GRUMP fournit les coordonnées longitude/latitude du centre des principales villes de chaque pays. prevR considère qu'une agglomération urbaine sera composée par les clusters urbains situés à une distance inférieure à 15 kilomètres du point central de la ville considérée. La recodification des clusters selon leur appartenance ou non à une agglomération urbaine est effectuée à l'aide de la fonction calcul.dist.cities(). La distance de 15 kilomètres ne représente pas la taille des agglomérations. Il s'agit plus précisément d'un critère de démarcation permettant de distinguer les clusters urbains appartenant à une agglomération des autres clusters urbains. Il peut bien sûr être modifié en fonction du contexte propre à chaque analyse, à travers le paramètre dist.

Dans un premier temps, nous conseillons de coder l'appartenance à une agglomération urbaine en prenant en compte un nombre important de villes. Lorsque la fonction calcul.dist.cities est appelée, une fenêtre s'ouvre et vous invite à choisir les villes pour lesquelles vous voulez calculer l'appartenance ou non à l'agglomération urbaine. Il est également possible de demander à la fonction de calculer cela pour l'ensemble des villes présentes dans le fichier villes avec type = 'all'.

```
> cm.clust <- calcul.dist.cities(cm.clust, cm.cities, lang='fr')
Une fenêtre va s'ouvrir. Veuillez sélectionner les villes retnues pour le paramètre U (utilisez les
touches CTRL et SHIFT).
```

```
> str(cm.clust)
'data.frame':   466 obs. of  12 variables:
 $ cluster       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ x             : num   9.72 13.53 15.23 14.58 10.30 ...
 $ y             : num   4.04  9.10 10.33 12.77  4.52 ...
 $ residence     : Factor w/ 2 levels "Rural","Urban": 2 1 2 1 2 2 2 1 2 1 ...
 $ region        : num  3 7 5 5 6 12 5 9 12 5 ...
 $ region.name   : Factor w/ 12 levels "ADAMAOUA","CENTRE",..: 3 7 5 5 6 12 5 9 ...
 $ n             : num  9 26 31 22 10 4 9 17 16 22 ...
 $ nweight       : num  10.79 24.15 62.14 34.87  6.59 ...
 $ obs.prevalence: num  0.00 0.00 3.13 0.00 0.00 ...
 $ dist.city     : num    1.87  26.17 102.64  90.79  62.26 ...
 $ city.name     : chr  "DOUALA" "GAROUA" "MAROUA" "KOUSSERI" ...
 $ urban.area    : Factor w/ 2 levels "in urban area",..: 1 2 2 2 2 1 2 2 1 2 ...
```

```
> infos.prev(cm.clust, lang='fr')
Statistiques du fichier :
* 466 clusters.
* 9900 observations valides.
* Prévalence globale de 5.51%.
* Valeur de Noptimal proposée : 363
* Nombre d'agglomérations urbaines trouvées : 22
* Noms des agglomérations urbaines trouvées :
BAFANG, BAFOUSSAM, BAMENDA, BERTOUA, BUEA, DOUALA, DSCHANG, EBOLOWA, EDEA, FOUMBAN,
GAROUA, KOUSSERI, KUMBA, KUMBO, LIMBE, LOUM, MAROUA, MBOUDA, NGAOUNDERE, NKONGSAMBA,
TIKO, YAOUNDE.
```


calcul.dist.cities() ajoute trois colonnes au tableau de données cluster qui lui est fourni en entrée :

- *dist.city* : la distance à la ville la plus proche ;
- *city.name* : nom de la ville la plus proche ;
- *urban.area :* l'appartenance ou nom à l'agglomération urbaine de la ville la plus proche.

infos.prev() détecte la présence ou non de ces trois variables et affiche le cas échéant le nombre d'agglomérations urbaines retenues ainsi que leurs noms.

### 7.2.3 Choix des agglomérations urbaines

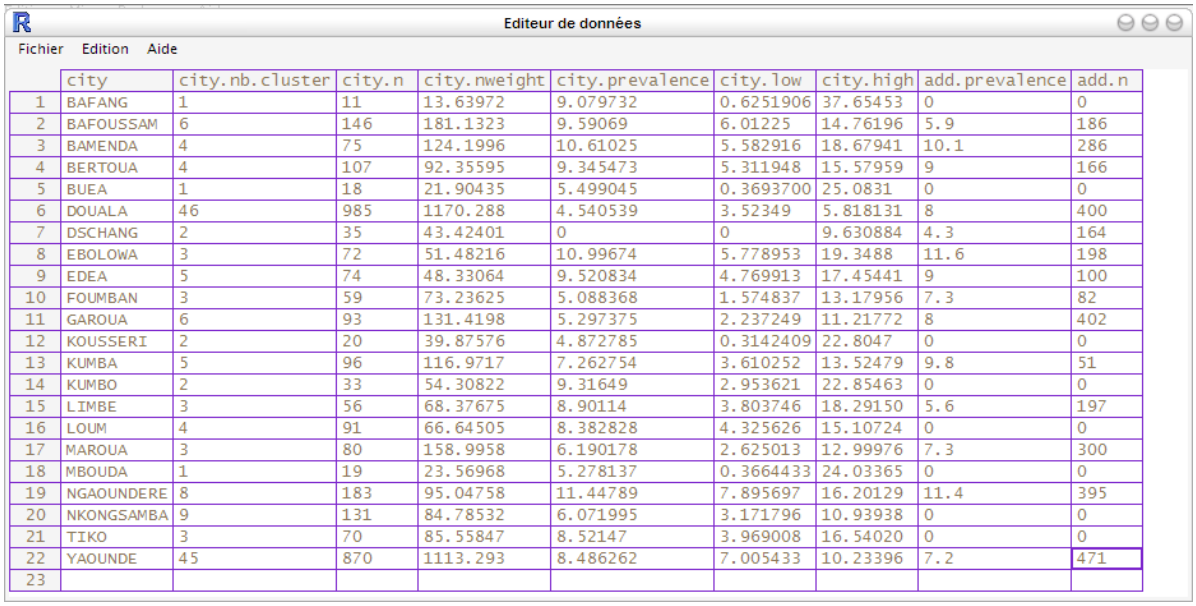
La fonction verif.urb() permet de calculer pour chaque agglomération urbaine le nombre de clusters concernés, le nombre d'observations valides, la prévalence observée sur l'agglomération et un intervalle de confiance de celle-ci. Le niveau de confiance de ce dernier peut être modifié avec le paramètre conf.level (0,9 soit 90 % par défaut).

```
> verif.urb(cm.clust)
         city city.nb.cluster city.n city.nweight city.prevalence  city.low city.high conf.level
1      BAFANG               1     11     13.63972        9.079732 0.6251906 37.654535        0.9
2   BAFOUSSAM               6    146    181.13233        9.590690 6.0122504 14.761959        0.9
3     BAMENDA               4     75    124.19958       10.610245 5.5829161 18.679413        0.9
4     BERTOUA               4    107     92.35595        9.345473 5.3119485 15.579592        0.9
5        BUEA               1     18     21.90435        5.499045 0.3693700 25.083099        0.9
6      DOUALA              46    985   1170.28768        4.540539 3.5234900  5.818131        0.9
7     DSCHANG               2     35     43.42401        0.000000 0.0000000  9.630884        0.9
8     EBOLOWA               3     72     51.48216       10.996741 5.7789535 19.348799        0.9
9        EDEA               5     74     48.33064        9.520834 4.7699133 17.454408        0.9
10    FOUMBAN               3     59     73.23625        5.088368 1.5748370 13.179564        0.9
11     GAROUA               6     93    131.41981        5.297375 2.2372492 11.217723        0.9
12   KOUSSERI               2     20     39.87576        4.872785 0.3142409 22.804702        0.9
13      KUMBA               5     96    116.97174        7.262754 3.6102521 13.524788        0.9
14      KUMBO               2     33     54.30822        9.316490 2.9536209 22.854626        0.9
```

```
15       LIMBE                3     56      68.37675         8.901140 3.8037463 18.291504         0.9
16        LOUM                4     91      66.64505         8.382828 4.3256258 15.107244         0.9
17      MAROUA                3     80     158.99576         6.190178 2.6250128 12.999756         0.9
18      MBOUDA                1     19      23.56968         5.278137 0.3664433 24.033647         0.9
19 NGAOUNDERE                8    183      95.04758        11.447887 7.8956974 16.201288         0.9
20 NKONGSAMBA                9    131      84.78532         6.071995 3.1717964 10.939380         0.9
21        TIKO                3     70      85.55847         8.521471 3.9690084 16.540203         0.9
22     YAOUNDE               45    870    1113.29302         8.486262 7.0054333 10.233962         0.9
```

Lorsque l'on dispose de données complémentaires provenant d'une autre source, il est possible de les ajouter afin de procéder à une comparaison, avec le paramètre add = TRUE. Une fenêtre s'ouvrira alors pour saisir la prévalence et l'effectif pour chaque agglomération de cette autre source. Dans notre exemple, nous avons ajouté, pour comparaison, des données provenant de la surveillance sentinelle des femmes enceintes[6].

```
> verif.urb(cm.clust, add=TRUE, lang='fr')
Une fenêtre va s'ouvrir. Compléter les colonnes 'add.prevalence' et 'add.n' pour chaque ville, puis
fermer. Pour 'add.prevalence', rentrer les données en %. Par exemple, pour 0,034=3,4%, saisir 3.4.
```



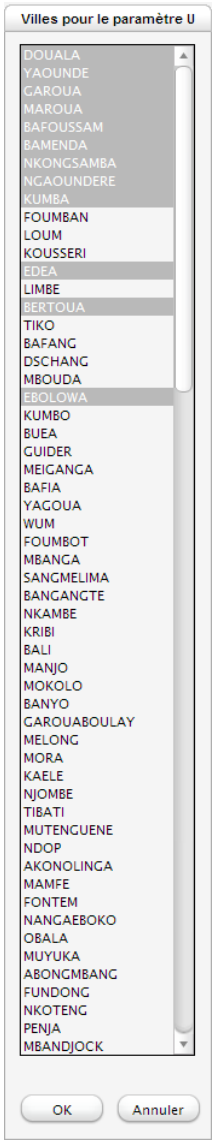

| | city | city.nb.cluster | city.n | city.nweight | city.prevalence | city.low | city.high | add.prevalence | add.n |
|---|---|---|---|---|---|---|---|---|---|
| 1 | BAFANG | 1 | 11 | 13.63972 | 9.079732 | 0.6251906 | 37.65453 | 0 | 0 |
| 2 | BAFOUSSAM | 6 | 146 | 181.1323 | 9.59069 | 6.01225 | 14.76196 | 5.9 | 186 |
| 3 | BAMENDA | 4 | 75 | 124.1996 | 10.61025 | 5.582916 | 18.67941 | 10.1 | 286 |
| 4 | BERTOUA | 4 | 107 | 92.35595 | 9.345473 | 5.311948 | 15.57959 | 9 | 166 |
| 5 | BUEA | 1 | 18 | 21.90435 | 5.499045 | 0.3693700 | 25.0831 | 0 | 0 |
| 6 | DOUALA | 46 | 985 | 1170.288 | 4.540539 | 3.52349 | 5.818131 | 8 | 400 |
| 7 | DSCHANG | 2 | 35 | 43.42401 | 0 | 0 | 9.630884 | 4.3 | 164 |
| 8 | EBOLOWA | 3 | 72 | 51.48216 | 10.99674 | 5.778953 | 19.3488 | 11.6 | 198 |
| 9 | EDEA | 5 | 74 | 48.33064 | 9.520834 | 4.769913 | 17.45441 | 9 | 100 |
| 10 | FOUMBAN | 3 | 59 | 73.23625 | 5.088368 | 1.574837 | 13.17956 | 7.3 | 82 |
| 11 | GAROUA | 6 | 93 | 131.4198 | 5.297375 | 2.237249 | 11.21772 | 8 | 402 |
| 12 | KOUSSERI | 2 | 20 | 39.87576 | 4.872785 | 0.3142409 | 22.8047 | 0 | 0 |
| 13 | KUMBA | 5 | 96 | 116.9717 | 7.262754 | 3.610252 | 13.52479 | 9.8 | 51 |
| 14 | KUMBO | 2 | 33 | 54.30822 | 9.31649 | 2.953621 | 22.85463 | 0 | 0 |
| 15 | LIMBE | 3 | 56 | 68.37675 | 8.90114 | 3.803746 | 18.29150 | 5.6 | 197 |
| 16 | LOUM | 4 | 91 | 66.64505 | 8.382828 | 4.325626 | 15.10724 | 0 | 0 |
| 17 | MAROUA | 3 | 80 | 158.9958 | 6.190178 | 2.625013 | 12.99976 | 7.3 | 300 |
| 18 | MBOUDA | 1 | 19 | 23.56968 | 5.278137 | 0.3664433 | 24.03365 | 0 | 0 |
| 19 | NGAOUNDERE | 8 | 183 | 95.04758 | 11.44789 | 7.895697 | 16.20129 | 11.4 | 395 |
| 20 | NKONGSAMBA | 9 | 131 | 84.78532 | 6.071995 | 3.171796 | 10.93938 | 0 | 0 |
| 21 | TIKO | 3 | 70 | 85.55847 | 8.52147 | 3.969008 | 16.54020 | 0 | 0 |
| 22 | YAOUNDE | 45 | 870 | 1113.293 | 8.486262 | 7.005433 | 10.23396 | 7.2 | 471 |
| 23 | | | | | | | | | |

```
         city city.nb.cluster city.n city.nweight city.prevalence  city.low city.high
1      BAFANG                1     11     13.63972        9.079732 0.6251906 37.654535
2   BAFOUSSAM                6    146    181.13233        9.590690 6.0122504 14.761959
3     BAMENDA                4     75    124.19958       10.610245 5.5829161 18.679413
4     BERTOUA                4    107     92.35595        9.345473 5.3119485 15.579592
5        BUEA                1     18     21.90435        5.499045 0.3693700 25.083099
6      DOUALA               46    985   1170.28768        4.540539 3.5234900  5.818131
7     DSCHANG                2     35     43.42401        0.000000 0.0000000  9.630884
8     EBOLOWA                3     72     51.48216       10.996741 5.7789535 19.348799
9        EDEA                5     74     48.33064        9.520834 4.7699133 17.454408
10    FOUMBAN                3     59     73.23625        5.088368 1.5748370 13.179564
11     GAROUA                6     93    131.41981        5.297375 2.2372492 11.217723
12   KOUSSERI                2     20     39.87576        4.872785 0.3142409 22.804702
13      KUMBA                5     96    116.97174        7.262754 3.6102521 13.524788
14      KUMBO                2     33     54.30822        9.316490 2.9536209 22.854626
15     LIMBE                 3     56     68.37675        8.901140 3.8037463 18.291504
16      LOUM                 4     91     66.64505        8.382828 4.3256258 15.107244
17     MAROUA                3     80    158.99576        6.190178 2.6250128 12.999756
18     MBOUDA                1     19     23.56968        5.278137 0.3664433 24.033647
19 NGAOUNDERE                8    183     95.04758       11.447887 7.8956974 16.201288
20 NKONGSAMBA                9    131     84.78532        6.071995 3.1717964 10.939380
21       TIKO                3     70     85.55847        8.521471 3.9690084 16.540203
22    YAOUNDE               45    870   1113.29302        8.486262 7.0054333 10.233962
```

[6] Suivant les agglomérations, des données datant de 2002 et lorsqu'elles n'étaient pas disponibles de 2000 : National AIDS Control Committe. **National HIV Sentinel surveillance Report (2002)**. Yaoundé. Ministry of Public Health, 2003, 42 pages. & National AIDS Control Committee. **Technical Report national Serosurvey on VIH/Syphilis (2000)**. Yaoundé. Ministry of Public Health, 2000, 219 pages.

```
   add.prevalence add.n  add.low  add.high p.value.comparison conf.level
1             0.0     0       NA        NA                NA        0.9
2             5.9   186 3.445394  9.762477         0.21670308        0.9
3            10.1   286 7.416399 13.654325         0.83359063        0.9
4             9.0   166 5.760419 13.731306         1.00000000        0.9
5             0.0     0       NA        NA                NA        0.9
6             8.0   400 5.932209 10.663398         0.01391705        0.9
7             4.3   164 2.115241  8.071971         0.60877976        0.9
8            11.6   198 8.163050 16.172756         1.00000000        0.9
9             9.0   100 4.936702 15.444009         1.00000000        0.9
10            7.3    82 3.410503 14.282425         0.73451911        0.9
11            8.0   402 5.902433 10.611184         0.51342936        0.9
12            0.0     0       NA        NA                NA        0.9
13            9.8    51 4.202260 19.996781         0.75275672        0.9
14            0.0     0       NA        NA                NA        0.9
15            5.6   197 3.251432  9.229574         0.35883764        0.9
16            0.0     0       NA        NA                NA        0.9
17            7.3   300 5.081109 10.398776         1.00000000        0.9
18            0.0     0       NA        NA                NA        0.9
19           11.4   395 8.908253 14.426918         1.00000000        0.9
20            0.0     0       NA        NA                NA        0.9
21            0.0     0       NA        NA                NA        0.9
22            7.2   471 5.399582  9.552642         0.46218264        0.9
```

Les informations renvoyées par verif.urb() peuvent guider le choix des agglomérations retenues pour le paramètre *U*. Il ne faut pas perdre de vue que, si une agglomération est retenue pour *U*, alors la prévalence d'un cluster de cette agglomération sera calculée uniquement à partir de clusters de cette même agglomération.




Plusieurs critères peuvent être pris en compte pour sélectionner les agglomérations urbaines :

- Tout d'abord, et c'est le plus évident, les villes pour lesquelles les nombres de clusters et d'observations valides sont importants. Dans le cas présent, nous pouvons décider de retenir les agglomérations comportant au moins 100 observations valides réparties sur au moins 6 clusters. Il s'agit de Douala, Yaoundé, Bafoussam, N'Gaoundéré et Nkongsamba.
- Il est également possible de retenir des agglomérations qui, bien que comportant un nombre d'observations valides plus faible, présentent une prévalence proche de celle calculée à partir d'une autre source de données. Dans notre exemple, nous pouvons retenir ainsi Bamenda, Bertoua, Kumba, Edéa et Ebolowa.
- Enfin, il est possible de retenir des agglomérations où la prévalence observée, bien qu'inférieure à celle mesurée par une autre source, reste supérieure à celle de son voisinage. La prise en compte de ces agglomérations permettra alors de les faire ressortir sur la carte produite. Nous retiendrons selon ce critère les agglomérations de Garoua et Maroua.

Pour les autres agglomérations, l'absence de données suffisantes dans l'EDS couplée à une absence de données complémentaires où à une contradiction avec la source complémentaire ne permet pas de se positionner. Il est alors préférable de ne pas les inclure pour le paramètre *U* et d'estimer leur prévalence à partir des clusters voisins. Au moment de l'interprétation des cartes produites, il conviendra de préciser que les tendances affichées sur la carte pour les zones

correspondant à ces agglomérations correspondront aux tendances régionales de la zone et non à la tendance limitée à l'agglomération elle-même.

La sélection des agglomérations urbaines retenues pour le paramètre U résulte d'un choix raisonné selon l'hypothèse que la prévalence observée au sein d'une agglomération rends compte de la réalité épidémique de cette agglomération, soit parce que la quantité d'information présente dans l'EDS est suffisante, soit parce que d'autres sources d'informations rendent cette hypothèse crédible. Il est possible de réaliser plusieurs cartes avec des sélections différentes pour le paramètre U afin de tester plusieurs hypothèses.

Une fois les agglomérations choisies pour la paramètre U, il reste à appliquer de nouveau la fonction calcul.dist.cities(), limitée cette fois-ci aux agglomérations retenues.

```
> cm.clust <- calcul.dist.cities(cm.clust, cm.cities, lang='fr')
Une fenêtre va s'ouvrir. Veuillez sélectionner les villes retnues pour le paramètre U (utilisez les
touches CTRL et SHIFT).

> infos.prev(cm.clust, lang='fr')
Statistiques du fichier :
* 466 clusters.
* 9900 observations valides.
* Prévalence globale de 5.51%.
* Valeur de Noptimal proposée : 363
* Nombre d'agglomérations urbaines trouvées : 12
* Noms des agglomérations urbaines trouvées :
BAFOUSSAM, BAMENDA, BERTOUA, DOUALA, EBOLOWA, EDEA, GAROUA, KUMBA, MAROUA, NGAOUNDERE, NKONGSAMBA,
YAOUNDE.
```

## 7.3 Réaliser plusieurs estimations simultanément

Ayant choisi les différents paramètres, il est possible d'appeler la fonction estimate.prev() avec plusieurs valeurs de ces derniers. Ainsi, il sera possible de cartographier par la suite les différentes estimations, lorsque seul le paramètre *N* est pris en compte, lorsque l'on tient compte de *N* et *R* et enfin lorsqu'on rajoute le paramètre *U*.

Les options N et R de estimate.prev peuvent prendre une valeur numérique ou une liste de valeurs numériques. L'estimation sera réalisée pour chaque combinaison des deux paramètres. Pour ne pas prendre en compte un de ces deux paramètres, il suffit de lui spécifier la valeur infinie Inf. Pour spécifier une liste, le plus simple consiste à utiliser la fonction c() (voir les exemples plus loin).

L'option U peut prendre trois valeurs : TRUE (le paramètre *U* est pris en compte), FALSE (il ne l'est pas), 2 (les estimations sont réalisées deux fois, une fois en tenant compte de *U*, une fois sans).

Lorsque plusieurs estimations sont réalisées simultanément, estimate.prev() renvoie un tableau de données avec une ligne par cluster **et** par estimation et fourni pour chaque cluster et chaque estimation les valeurs de la prévalence estimée (*est.prevalence*), le nombre d'observations retenues

(*circle.count*), le nombre de clusters inclus dans le cercle (*circle.nb.clusters*) et le rayon du cercle de lissage (*circle.radius*).

Avant de pouvoir réaliser une interpolation spatiale, il est nécessaire d'utiliser la fonction extract.data(), qui permet d'extraire une seule estimation du tableau de données, ou bien la fonction merge.prev(), qui réarrange les données de manière à obtenir un tableau de données avec une ligne par cluster. Les nouvelles colonnes créées par estimate.prev() sont alors dupliquées pour chaque estimation et leur nom prend alors un suffixe de la forme *.Nvaleur-de-n.Rvaleur-de-r.Uvaleur-de-U*. merge.prev() peut être directement appliquée aux résultats renvoyés par estimate.prev() en précisant merge.result = TRUE (voir les deux exemples ci-après).

```
> cm.prev <- estimate.prev(cm.clust,N=363, R=c(118, Inf), U=2, lang='fr')
> str(cm.prev)
'data.frame':   1864 obs. of  22 variables:
 $ cluster           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ x                 : num   9.72 13.53 15.23 14.58 10.30 ...
 $ y                 : num   4.04  9.10 10.33 12.77  4.52 ...
 $ residence         : Factor w/ 2 levels "Rural","Urban": 2 1 2 1 2 2 2 1 2 1 ...
 $ region            : num  3 7 5 5 6 12 5 9 12 5 ...
 $ region.name       : Factor w/ 12 levels "ADAMAOUA","CENTRE",..: 3 7 5 5 6 12 5 9 12 5 ...
 $ longitude         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ latitude          : num  NA NA NA NA NA NA NA NA NA NA ...
 $ n                 : num  9 26 31 22 10 4 9 17 16 22 ...
 $ nweight           : num  10.79 24.15 62.14 34.87  6.59 ...
 $ obs.prevalence    : num  0.00 0.00 3.13 0.00 0.00 ...
 $ dist.city         : num    1.87  26.17 102.64  90.79  62.26 ...
 $ city.name         : chr  "DOUALA" "GAROUA" "MAROUA" "KOUSSERI" ...
 $ urban.area        : Factor w/ 2 levels "in urban area",..: 1 2 2 2 2 1 2 2 1 2 ...
 $ est.prevalence    : num  5.46 1.90 2.00 3.01 3.81 ...
 $ circle.count      : num  380 380 380 69 370 374 391 365 371 369 ...
 $ circle.radius     : num    3.11  59.01 102.51  91.60  69.52 ...
 $ circle.nb.clusters: int  17 18 16 6 21 20 17 17 19 16 ...
 $ quality.indicator : num     0.498  178.620  539.037 1010.180  251.253 ...
 $ N.parameter       : num  363 363 363 363 363 363 363 363 363 363 ...
 $ R.parameter       : num  118 118 118 118 118 118 118 118 118 118 ...
 $ U.parameter       : num  0 0 0 0 0 0 0 0 0 0 ...

> cm.prev <- estimate.prev(cm.clust,N=363, R=c(118, Inf), U=2, lang='fr',
merge.result=TRUE)
> str(cm.prev)
'data.frame':   466 obs. of  32 variables:
 $ cluster                         : int  1 10 100 101 102 103 104 105 106 107 ...
 $ x                               : num   9.72 13.57 11.23 14.71 11.55 ...
 $ y                               : num   4.04 10.25  4.74 10.43  3.88 ...
 $ residence                       : Factor w/ 2 levels "Rural","Urban": 2 1 2 1 2 2 2 2 1 2 ...
 $ region                          : num  3 5 2 5 12 7 8 12 7 3 ...
 $ region.name                     : Factor w/ 12 levels "ADAMAOUA","CENTRE",..: 3 5 2 5 12 7 8 ...
 $ n                               : num  9 22 18 8 17 36 57 16 16 10 ...
 $ nweight                         : num  10.8 34.8 28.7 12.7 22.1 ...
 $ obs.prevalence                  : num   0.00 13.63  0.00  0.00  5.81 ...
 $ dist.city                       : num    1.87  91.90 102.20  45.20   3.59 ...
 $ city.name                       : chr  "DOUALA" "MAROUA" "YAOUNDE" "MAROUA" ...
 $ urban.area                      : Factor w/ 2 levels "in urban area",..: 1 2 2 2 1 2 2 1 2 1 ...
 $ est.prevalence.N363.R118.U0     : num  5.46 1.94 3.16 2.49 7.61 ...
 $ circle.count.N363.R118.U0       : num  380 369 375 379 367 376 387 370 380 386 ...
 $ circle.radius.N363.R118.U0      : num   3.11 70.93 69.52 61.67  3.73 ...
 $ circle.nb.clusters.N363.R118.U0 : int  17 16 15 15 17 18 18 16 18 18 ...
 $ quality.indicator.N363.R118.U0  : num    0.498 261.887 249.583 195.371   0.727 ...
 $ est.prevalence.N363.RInf.U0     : num  5.46 1.94 3.16 2.49 7.61 ...
 $ circle.count.N363.RInf.U0       : num  380 369 375 379 367 376 387 370 380 386 ...
 $ circle.radius.N363.RInf.U0      : num   3.11 70.93 69.52 61.67  3.73 ...
 $ circle.nb.clusters.N363.RInf.U0 : int  17 16 15 15 17 18 18 16 18 18 ...
 $ quality.indicator.N363.RInf.U0  : num    0.498 261.887 249.583 195.371   0.727 ...
 $ est.prevalence.N363.R118.U12    : num  5.46 1.94 3.16 1.54 7.61 ...
 $ circle.count.N363.R118.U12      : num  380 369 375 384 367 356 387 370 371 386 ...
 $ circle.radius.N363.R118.U12     : num   3.11 70.93 69.52 72.50  3.73 ...
```

```
 $ circle.nb.clusters.N363.R118.U12: int  17 16 15 15 17 15 18 16 15 18 ...
 $ quality.indicator.N363.R118.U12 : num    0.498 261.887 249.583 268.231   0.727 ...
 $ est.prevalence.N363.RInf.U12    : num  5.46 1.94 3.16 1.54 7.61 ...
 $ circle.count.N363.RInf.U12      : num  380 369 375 384 367 393 387 370 371 386 ...
 $ circle.radius.N363.RInf.U12     : num   3.11 70.93 69.52 72.50  3.73 ...
 $ circle.nb.clusters.N363.RInf.U12: int  17 16 15 15 17 16 18 16 15 18 ...
 $ quality.indicator.N363.RInf.U12 : num    0.498 261.887 249.583 268.231   0.727 ...
```

Dans ces exemples, quatre simulations ont été réalisées simultanément. Dans le premier cas, les données des 466 clusters ont été dupliquées 4 fois (une par estimation) et le tableau de données renvoyé comporte ainsi 1864 lignes.

Dans le second exemple, avec merge.result = TRUE, ce sont les variables *est.prevalence, circle.count, circle.radius* et *circle.nb.clusters* qui ont été dupliquées chacune 4 fois, et le nombre de lignes du tableau de données renvoyés correspond au nombre de clusters, soit 466. Les variables *N.parameter, R.parameter* et *U.parameter* ne sont plus présentes. Leur valeur a directement été intégrée dans le nom des nouvelles variables crées.

# 8. Interpolation spatiale de la prévalence

Une fois la prévalence de chaque cluster estimée, il est possible d'interpoler spatialement cette dernière pour obtenir une carte des variations spatiales du phénomène étudié. Deux techniques sont disponibles à partir de la fonction krige.prev() : l'interpolation linéaire selon l'inverse de la distance[7] et le krigeage (voir encadré page 26). Dans la suite de ce tutoriel nous parlerons essentiellement de la technique du krigeage ordinaire. Pour les personnes désirant utilisées la pondération inverse de la distance, les surfaces de tendances ou bien le krigeage universel, nous les renvoyons à la documentation des fonctions krige() et idw() du packages gstat puisque prevR a recours à ces deux fonctions pour la réalisation des interpolations spatiales.

Quelque soit la méthode utilisée, le principe de krige.prev() consiste à créer une grille plus ou moins fine composée de petites zones carrées ou pixels, puis à calculer pour chaque pixel la valeur correspondante de la variable interpolée, à partir de valeurs connues, en l'occurrence celles des zones enquêtées.

## 8.1 Principes généraux du krigeage ordinaire

Comme d'autres techniques d'interpolation, le krigeage calcule la valeur en un point de la variable interpolée à partir des valeurs connues en leur affectant à chacune une pondération. La technique de l'inverse de la distance utilise comme valeur de pondération l'inverse de la distance entre le point à estimée et le point enquêté. Le krigeage, quant à lui, calcule les pondérations à partir du degré de similarité de la variable. Statistiquement, ce degré de similarité correspond à la covariance entre les points enquêtés exprimée en fonction de la distance entre ces derniers.

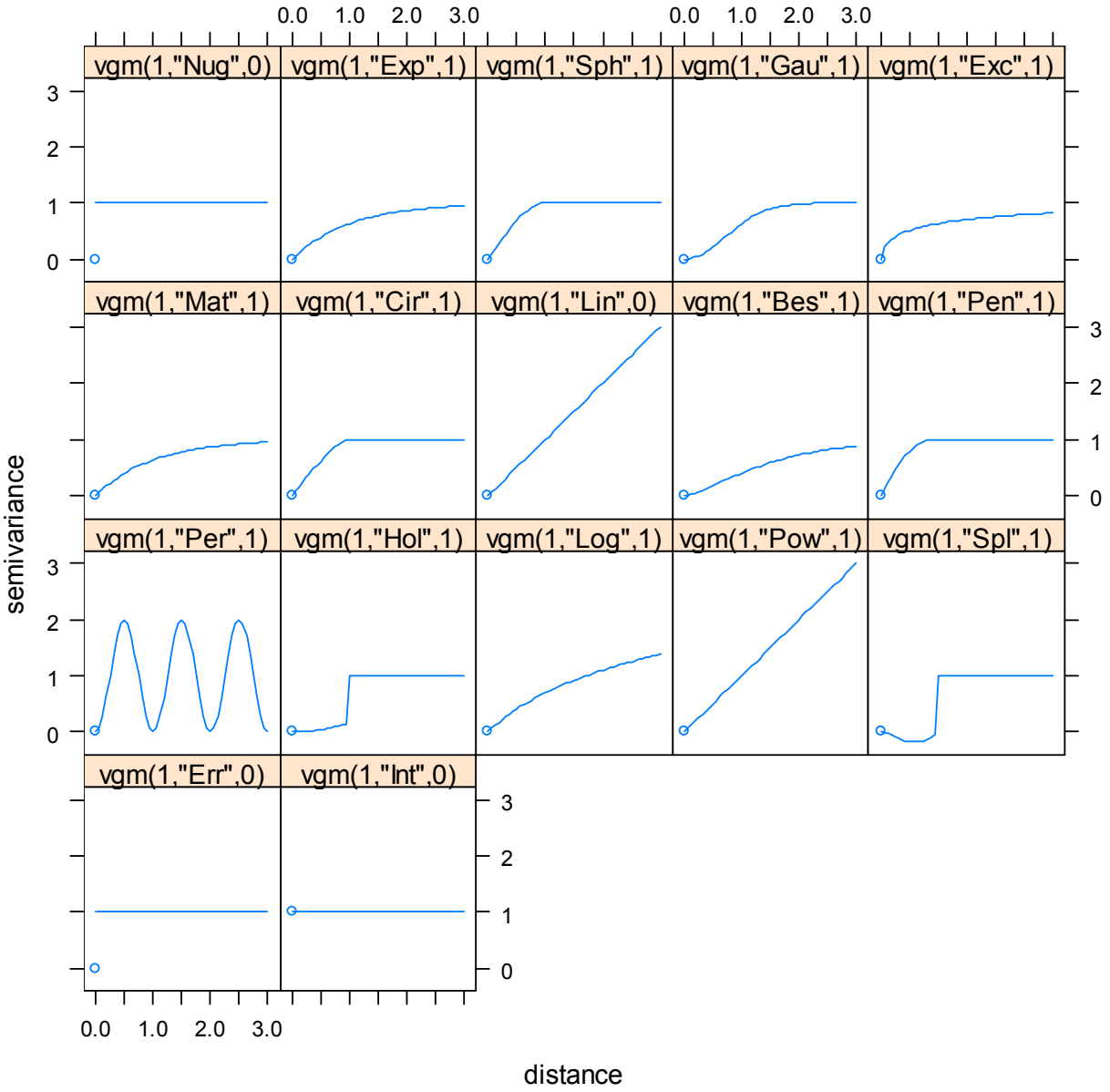
[7] En anglais, cette technique est appelée IDW pour Inverse Distance Weighting. Voir cet article en anglais de l'encyclopédie Wikipeia pour plus de renseignements : http://en.wikipedia.org/wiki/Inverse_distance_weighting.

Plus précisément, le krigeage n'utilise pas la covariance mais la moitié de celle-ci, encore appelée semi-variance. Est appelé semi-variogramme, l'expression de la semi-variance entre les points en fonction de leur distance. Il est représenté usuellement sous la forme d'un graphique avec la distance en abscisse et la semi-variance en ordonnée.

Pour réaliser une interpolation spatiale par krigeage, la première étape consiste à calculer sur l'échantillon des points enquêtés un semi-variogramme expérimental, à savoir les variations de la semi-variance en fonction de la distance entre les points.

Cependant, ce semi-variogramme expérimental n'est pas directement utilisable pour réaliser l'interpolation spatiale. Il faut le modéliser, c'est-à-dire trouver une fonction mathématique le décrivant au mieux. C'est cette fonction, ou modèle de semi-variogramme, qui sera utilisée pour le calcul effectif de l'interpolation.

Plusieurs types de courbes mathématiques existent pour modéliser le semi-variogramme. La fonction show.vgms() permet de visualiser celles qui sont prises en compte par le package gstat.



À l'usage, les modèles de semi-variogrammes de type *exponentiel* permettent d'obtenir des cartes lisibles et de qualité. La suite du tutoriel n'abordera que ce type de modèle de semi-variogrammes. Cependant, un utilisateur habitué à manier ce type d'outil géostatistique pourra avoir recours à d'autres types de semi-variogrammes.

L'ajustement d'un modèle de semi-variogramme aux données expérimentales sera abordé dans les parties suivantes au travers d'un exemple concret.

## 8.2 Les différents paramètres de krige.prev

Le tableau de données passé en paramètre de la fonction krige.prev() doit avoir été obtenu en spécifiant merge.result=TRUE à la fonction estimate.prev() ou bien en ayant eu recours à extract.data() de manière à ce qu'à chaque ligne du tableau de données correspondent à un cluster et à un seul. Si data comporte deux lignes ayant les mêmes coordonnées géographiques, alors vous obtiendrez une erreur dans krige.prev().

formula permet de spécifier la variable à interpoler sous la forme d'une formule. Pour interpoler, par exemple, la prévalence estimée pour *N=363*, *R=118* et *U=12*, nous appellerons krige.prev() avec formula = est.prevalence.N363.R118.U12 ~ 1. La formule employée est de la forme *variable ~ 1*. La partie *~1* indique qu'il s'agit d'un krigeage ordinaire. Des interpolations plus complexes peuvent être réalisés en spécifiant d'autres variables à la droite du ~. Pour cela nous vous renvoyons à la documentation de la fonction krige() du package gstat. Il est possible de réaliser plusieurs interpolations simultanément. Il suffit de passer une liste de formules au paramètre formula. Par exemple, pour interpoler à la fois la prévalence estimée et l'indicateur de qualité lorsque *N=363*, *R=118* et *U=12*, on entrera :

formula = c(est.prevalence.N363.R118.U12 ~ 1, quality.indicator.N363.R118.U12 ~ 1).

Le paramètre cell.size permet de définir la taille des pixels de la grille sur laquelle l'interpolation sera réalisée. Plus cell.size est petit, plus le nombre de pixels sera élevés, plus le temps de calcul sera long et plus la carte obtenue sera précise. Si ask.cell.size = TRUE (valeur par défaut), krige.prev calculera la taille de la grille obtenue avec la valeur de cell.size entrée et vous proposera de modifier cette valeur si besoin. L'unité de dimension de cell.size correspond à celle des coordonnées des clusters du tableau de données fourni en entrée.

Le paramètre type peut prendre plusieurs valeurs. Il détermine le type d'interpolation et la manière, en cas de krigeage, dont sera déterminé le modèle de semi-variogramme utilisé.

- Si type = 'idw', alors krige.prev réalisera une interpolation par inverse de la distance à l'aide la fonction idw() du package gstat. La puissance appliquée à l'inverse de la distance peut être précisée avec le paramètre idp.

- Si type = 'model', le modèle de semi-variogramme utilisé pour l'interpolation par krigeage sera celui passé au paramètre model. Si plusieurs interpolations sont réalisées simultanément, il est possible des spécifier plusieurs modèles à model. Voir la documentation de krige.prev() pour plus de détails.
- Si type = 'auto', krige.prev aura recours à la fonction fit.variogram() pour ajuster par la méthode des moindres carrés un modèle de semi-variogramme au semi-variogramme expérimental. C'est ce modèle qui sera ensuite utilisé pour l'interpolation spatiale. L'utilisateur doit rester vigilant dans la mesure où l'ajustement par la méthode des moindres carrés ne produit pas toujours un modèle de semi-variogramme adéquat.
- Par défaut, type = 'ask'. Dans le doute, utilisez de préférence ce mode de fonctionnement. Dans le cas présent, krige.prev() a toujours recours à fit.variogram() pour ajuster un modèle de semi-variogramme aux données expérimentales. Une fois celui-ci calculé, krige.prev() affiche le semi-variogramme expérimental ainsi que le modèle ajusté. L'utilisateur est alors invité à accepter le modèle ajusté par la méthode des moindres carrés ou à procéder à un ajustement manuel (voir l'exemple concret ci-après).

Le calcul d'interpolations spatiales sur des grilles fines est une opération longue et peut prendre plusieurs minutes. Soyez donc patient.

## 8.3 Exemple d'interpolation spatiale

Dans cet exemple, nous interpolerons la prévalence estimée au Cameroun avec les paramètres *N=363, R=118* et *U=12*. Le mode *ask* sera utilisé.

```
> cm.krige <- krige.prev(cm.prev, formula = est.prevalence.N363.R118.U12 ~ 1,
      boundary = cm.bounds, lang='fr')
```

La fonction commence par afficher la taille de la grille résultant de pixels carrés de 0,05 degrés de côté (valeur par défaut). Si l'on souhaite utiliser une grille plus fine (plus grand nombre de pixels), il faut réduire la longueur des côtés des pixels. (Une plus petite valeur de cell.size induit un nombre plus important de pixels sur une même zone géographique.)

```
Une taille de cellule de 0.05 induit une grille de 154x229 cellules.
Cela vous convient-il ?

1: Oui
2: Non

Sélection : 2

Entrez une nouvelle valeur de taille de cellule :
```

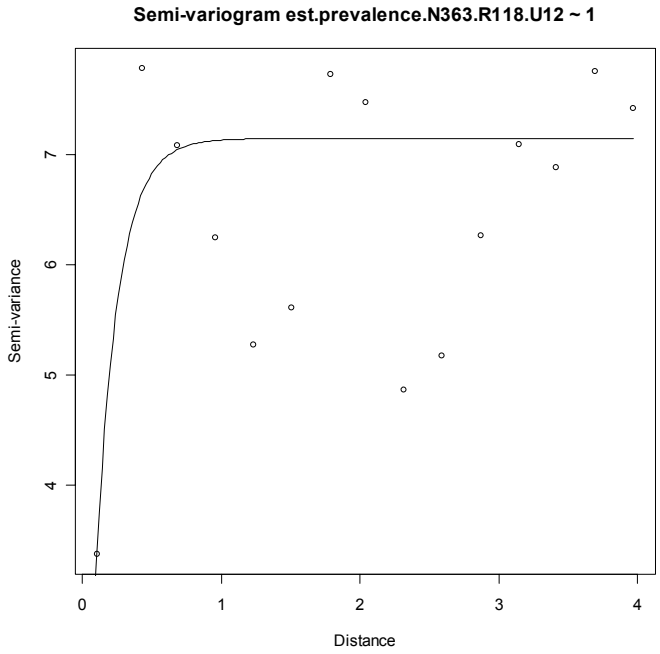
La nouvelle valeur est à saisir dans une fenêtre. Ici, nous avons choisi 0,025 degrés.

```
Une taille de cellule de 0.025 induit une grille de 308x458 cellules.
Cela vous convient-il ?

1: Oui
2: Non

Sélection : 1
```

Une fois la taille de la grille définie, la fonction calcule le semi-variogramme expérimental, ajuste un modèle de semi-variogramme aux données empiriques puis affiche les détails du modèle ajusté :

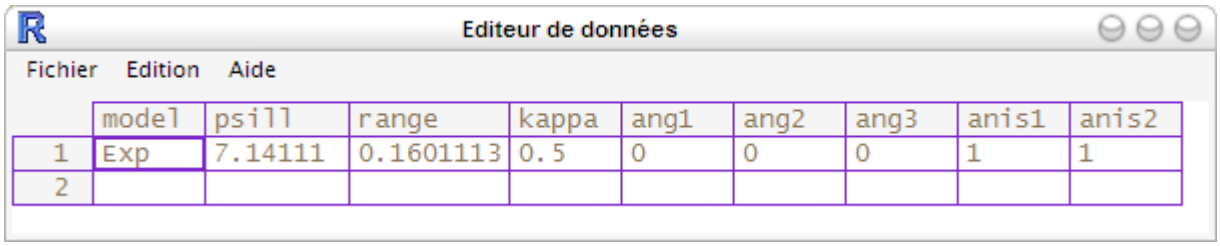
```
--- est.prevalence.N363.R118.U12 ~ 1 ---

  model   psill     range
1   Exp 7.14111 0.1601113
```

Le modèle en question est affiché dans une fenêtre graphique sous la forme d'une courbe. Les points représentent le semi-variogramme expérimental.




Le semi-variogramme expérimental étant plus ou moins régulier, l'ajustement n'a pas été optimal : le modèle ajusté croît trop rapidement vers une valeur plafond. Nous allons donc procéder à un ajustement manuel :

```
Ce modèle de variogramme convient-il ?

1: Oui
2: Non

Sélection : 2
Entrez un nouveau modèle.
```

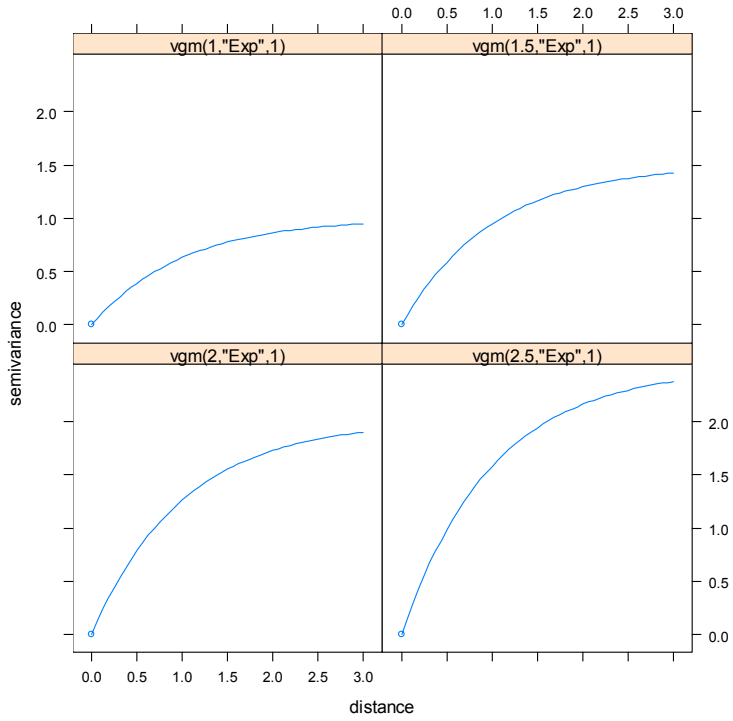



| | model | psill | range | kappa | ang1 | ang2 | ang3 | anis1 | anis2 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Exp | 7.14111 | 0.1601113 | 0.5 | 0 | 0 | 0 | 1 | 1 |
| 2 | | | | | | | | | |

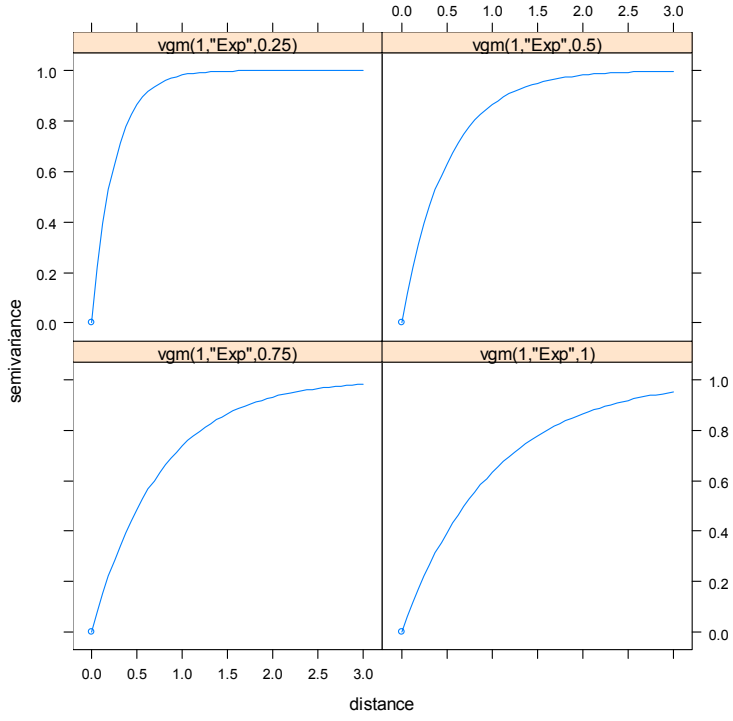
Une fenêtre s'ouvre permettant de modifier les paramètres du modèle. Pour plus de détails sur ces paramètres, consultez la documentation de la fonction vgm() du package gstat.

Pour un usage courant, seuls les paramètres *psill* et *range* seront à modifier. *psill* correspond à la valeur seuil vers laquelle le modèle tend et *range* à la rapidité avec laquelle le modèle s'approche de cette valeur seuil.

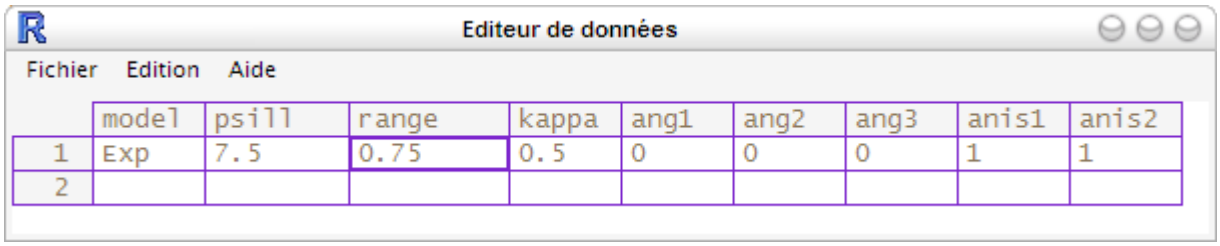
Ci-dessous, voici plusieurs modèles avec des valeurs de *psill* de 1, 1,5, 2 et 2,5. Plus la valeur de *psill* augmente, plus la courbe atteint une semi-variance élevée.




*range* influe sur la rapidité à laquelle le modèle s'approche de sa valeur seuil. Plus *range* est élevé, plus la croissance sera progressive. Ci-dessous, plusieurs modèles correspondant à des valeurs de *range* égales à 0,25, 0,5, 0,75 et 1.

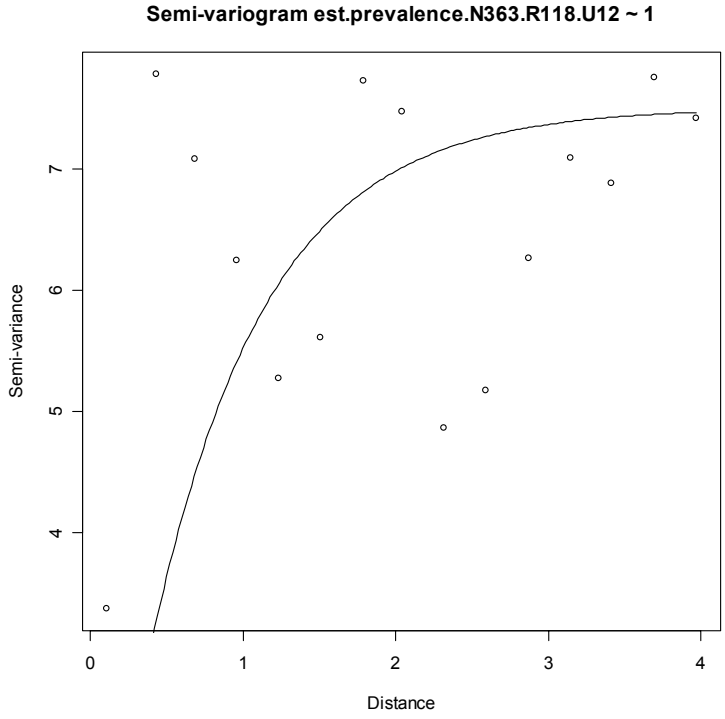


Dans le cas présent, nous allons modifier le modèle de manière à ce qu'il ait une croissance moins rapide (pour cela on augmente la valeur de *range*) et que sa valeur seuil soit un peu plus élevée (augmentation légère de *psill*). Nous allons donc entrer 7.5 pour *psill* et 0.75 pour *range*.




| | model | psill | range | kappa | ang1 | ang2 | ang3 | anis1 | anis2 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Exp | 7.5 | 0.75 | 0.5 | 0 | 0 | 0 | 1 | 1 |
| 2 | | | | | | | | | |

Le graphique est alors modifié pour afficher la courbe du nouveau modèle.




```
--- est.prevalence.N363.R118.U12 ~ 1 ---

  model psill range
1   Exp   7.5  0.75

Ce modèle de variogramme convient-il ?

1: Oui
2: Non
```

Si l'on désire affiner encore l'ajustement, il est possible de modifier à nouveau les valeurs des paramètres du modèle. On peut alors procéder à plusieurs essais et avancer par tâtonnement. Le calcul de l'interpolation spatiale se lance une fois le modèle accepté.

```
Sélection : 1

Modèle de variogramme utilisé :
  model psill range
1   Exp   7.5  0.75
[using ordinary kriging]
```

Pour effectuer plusieurs interpolations en même temps, il suffit de spécifier une liste de formules au paramètre formula à l'aide de la fonction c(). L'ensemble des résultats sera alors regroupé en un seul fichier, permettant de réaliser aisément des cartes comparatives. D'autres indicateurs peuvent être interpolés, tels que l'indicateur de qualité ou le rayon des cercles de lissage. Dans l'exemple ci-dessous, la prévalence estimée est interpolée selon trois cas de figure (utilisation de la méthode des cercles avec seul le paramètre *N*, avec les paramètres *N* et *R* et avec les trois paramètres *N*, *R* et *U*) ainsi que l'indicateur de qualité et le rayon des cercles de lissage de l'estimation avec *N*, *R* et *U*.

```
> cm.krige <- krige.prev(
      cm.prev,
      formula = c(
            est.prevalence.N363.RInf.U0 ~1,
            est.prevalence.N363.R118.U0 ~ 1,
            est.prevalence.N363.R118.U12 ~ 1,
            quality.indicator.N363.R118.U12 ~1,
            circle.radius.N363.R118.U12 ~1
      ),
      boundary = cm.bounds,
)
```

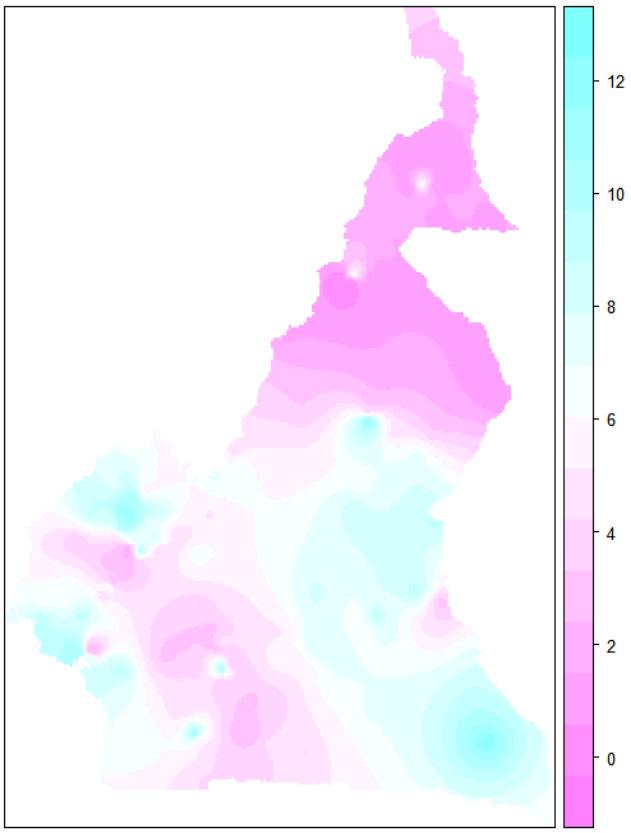
krige.prev() renvoie un objet de type *SpatialPixelsDataFrame*. Deux variables sont créées pour chacune des variables interpolées : l'une avec le suffixe *.pred* comportant pour chaque point de la grille la valeur de la prédiction ; l'autre avec le suffixe *.var* avec pour chaque point de la grille la variance de la prédiction (en cas de krigeage uniquement).

```
> str(cm.krige)
Formal class 'SpatialPixelsDataFrame' [package "sp"] with 7 slots
  ..@ data       :'data.frame': 71940 obs. of  12 variables:
  .. ..$ est.prevalence.N363.RInf.U0.pred    : num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  .. ..$ est.prevalence.N363.RInf.U0.var     : num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  .. ..$ est.prevalence.N363.R118.U0.pred    : num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  .. ..$ est.prevalence.N363.R118.U0.var     : num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  .. ..$ est.prevalence.N363.R118.U12.pred   : num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  .. ..$ est.prevalence.N363.R118.U12.var    : num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  .. ..$ quality.indicator.N363.R118.U12.pred: num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  .. ..$ quality.indicator.N363.R118.U12.var : num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  .. ..$ circle.radius.N363.R118.U12.pred    : num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  .. ..$ circle.radius.N363.R118.U12.var     : num [1:71940] NA NA NA NA NA NA NA NA NA NA ...
  ..@ coords.nrs : num(0)
  ..@ grid       :Formal class 'GridTopology' [package "sp"] with 3 slots
  .. .. ..@ cellcentre.offset: Named num [1:2] 8.49 1.65
  .. .. .. ..- attr(*, "names")= chr [1:2] "x" "y"
  .. .. ..@ cellsize         : Named num [1:2] 0.035 0.035
  .. .. .. ..- attr(*, "names")= chr [1:2] "x" "y"
  .. .. ..@ cells.dim        : Named int [1:2] 220 327
  .. .. .. ..- attr(*, "names")= chr [1:2] "x" "y"
  ..@ grid.index : int [1:71940] 1 2 3 4 5 6 7 8 9 10 ...
  ..@ coords     : num [1:71940, 1:2] 8.49 8.53 8.56 8.60 8.63 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : NULL
  .. .. ..$ : chr [1:2] "x" "y"
  ..@ bbox       : num [1:2, 1:2]  8.48  1.64 16.18 13.08
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:2] "x" "y"
  .. .. ..$ : chr [1:2] "min" "max"
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots
  .. .. ..@ projargs: chr NA
```

# 9. Cartographier les résultats

Les objets du type *SpatialPixelsDataFrame* peuvent être facilement représentés à l'aide de la fonction spplot() du package sp. Le premier paramètre spécifie l'ensemble de données, le second la variable représenter (ou plusieurs variables s'il s'agit d'une liste). Si le second paramètre est omis, spplot() représentera avec la même échelle colorimétrique l'ensemble des variables contenues dans l'ensemble de données.

```
> spplot(cm.krige,'est.prevalence.N363.R118.U12.pred')
```

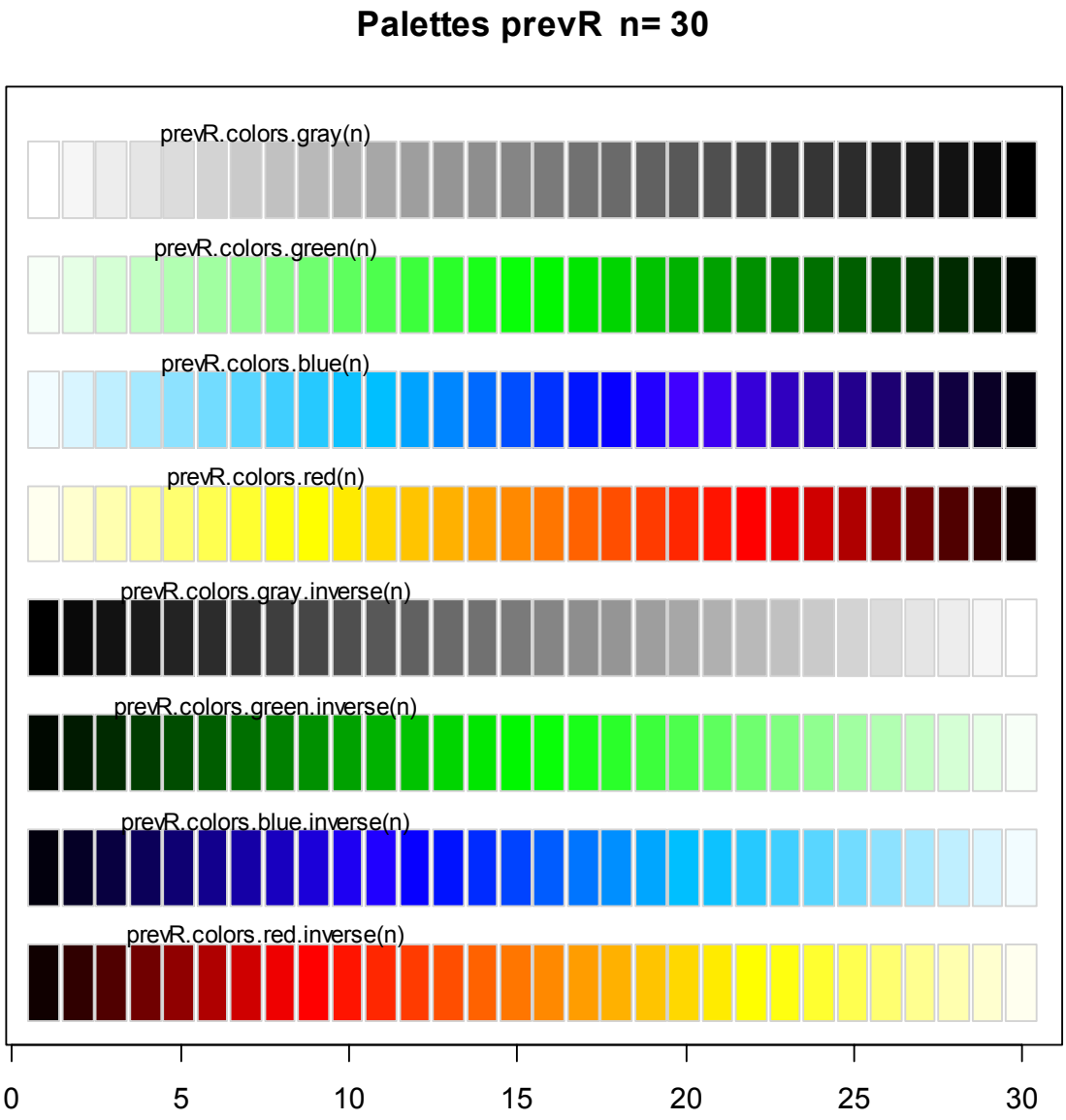



Il est possible de préciser un titre avec main et un sous-titre avec sub. Le paramètre cuts précise le nombre de plages de couleurs.

La fonction extract.col() peut être utile pour extraire certaines variables d'un ensemble de données. Voir la documentation de cette fonction.

La palette de couleurs à utiliser peut-être spécifiée avec col.regions. prevR fournit plusieurs palettes de couleurs, visible avec la fonction prevR.demo.pal().

```
> prevR.demo.pal(30)
```



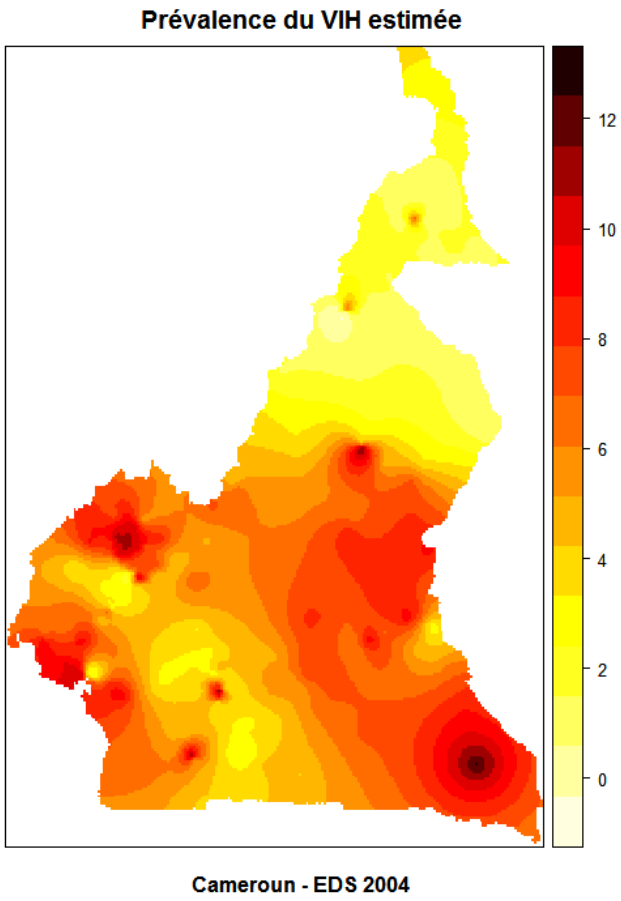

Ces palettes ont été conçues pour accentuer les contrastes en éclaircissant ou obscurcissant les valeurs extrêmes. D'autres palettes peuvent être utilisées. Consultez l'aide de la fonction rainbow() ou le package RColorBrewer. En appelant une fonction palette de couleurs avec spplot(), pensez à générer au moins une couleur de plus que de plages de niveaux.

Le nombre de plages de niveaux modifie le rendu global de la carte générée. Pour un rendu lisse, utilisez un nombre élevé de plages. Le nombre de plages est défini par le paramètre cuts.

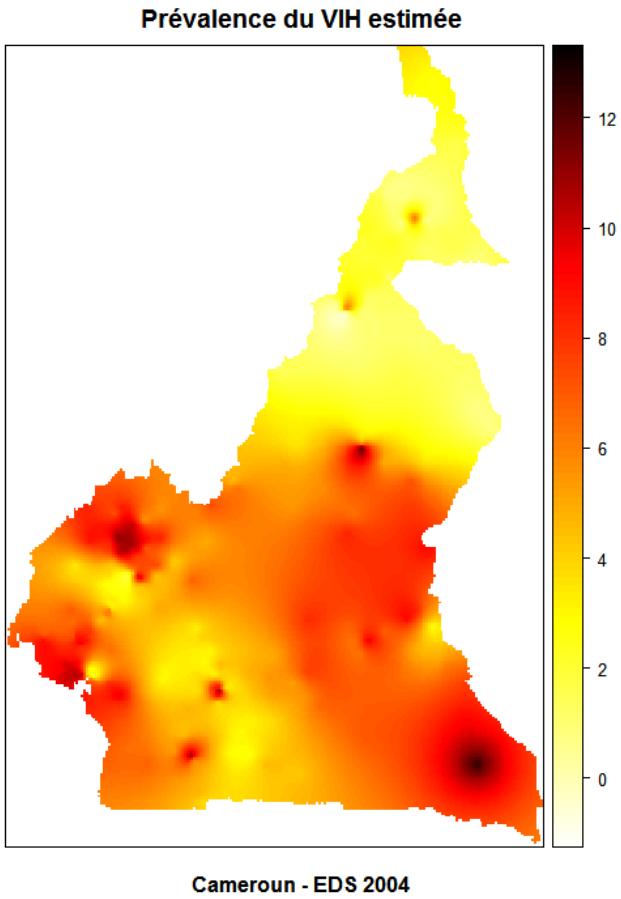
Il est possible de représenter plusieurs variables simultanément, avec la même échelle. Pour cela, il faut passer une liste de noms de variables à spplot() via le second paramètre appelé également zcol.

Inspirez-vous des exemples suivants :

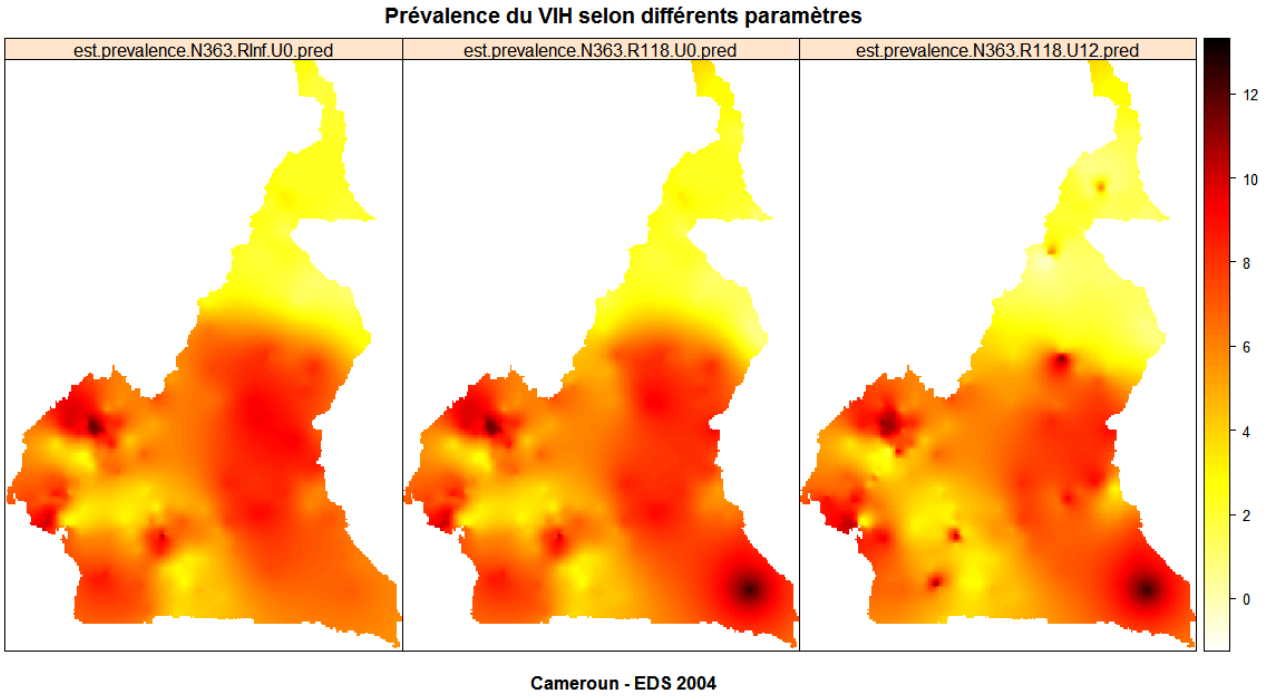
```
> spplot(
      cm.krige,
      'est.prevalence.N363.R118.U12.pred',
      cuts=15,
      col.regions=prevR.colors.red(16),
      main='Prévalence du VIH estimée',
      sub='Cameroun - EDS 2004'
)
```



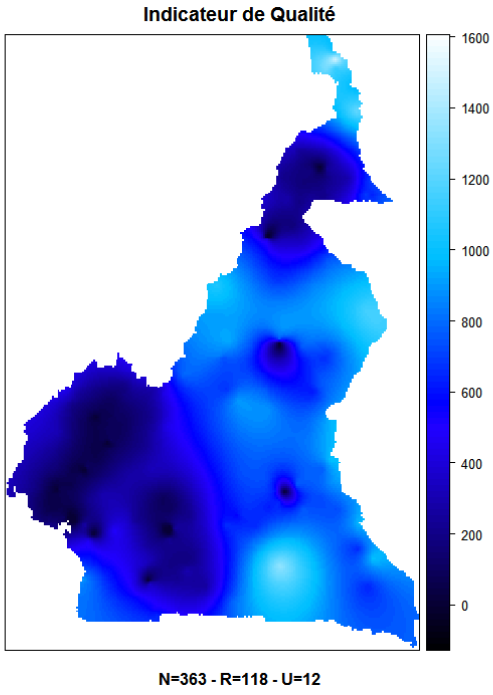

```
> spplot(
      cm.krige,
      'est.prevalence.N363.R118.U12.pred',
      cuts=100,
      col.regions=prevR.colors.red(101),
      main='Prévalence du VIH estimée',
      sub='Cameroun - EDS 2004'
)
```



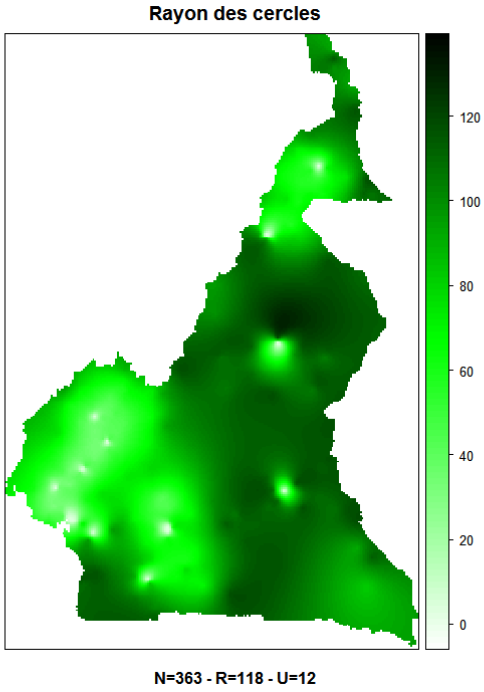
```
> spplot(
      cm.krige,
      c(
            'est.prevalence.N363.RInf.U0.pred',
            'est.prevalence.N363.R118.U0.pred',
            'est.prevalence.N363.R118.U12.pred'
      ),
      cuts=100,
      col.regions=prevR.colors.red(101),
      main='Prévalence du VIH selon différents paramètres',
      sub='Cameroun - EDS 2004'
)
```




```
> spplot(
      cm.krige,
      'quality.indicator.N363.R118.U12.pred',
      cuts=100,
      col.regions=prevR.colors.blue.inverse(101),
      main='Indicateur de Qualité',
      sub='N=363 - R=118 - U=12'
)
```



```
> spplot(
      cm.krige,
      'circle.radius.N363.R118.U12.pred',
      cuts=100,
      col.regions=prevR.colors.green(101),
      main='Rayon des cercles',
      sub='N=363 - R=118 - U=12'
)
```




La fonction spplot() crée par défaut une légende dont le minimum et le maximum diffèrent sensiblement du minimum et du maximum réel de la variable à représenter. Il est possible de connaître facilement le minimum et le maximum de chaque variable à l'aide la fonction summary().

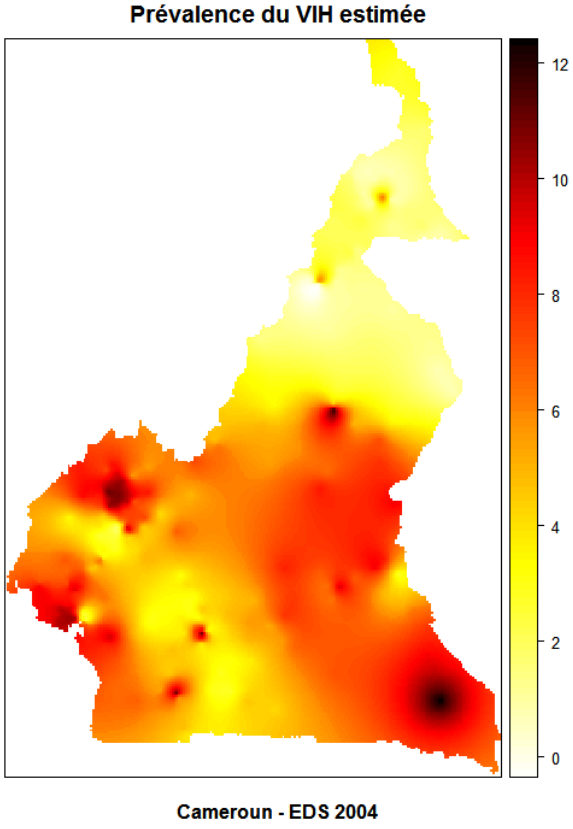
```
> summary(cm.krige)
Object of class SpatialPixelsDataFrame
Coordinates:
       min      max
x 8.477263 16.17726
y 1.635048 13.08005
Is projected: NA
proj4string : [NA]
Number of points: 71940

Data attributes:
 obs.prevalence.pred obs.prevalence.var  est.prevalence.N363.RInf.U0.pred
 Min.   :   -3.246   Min.   :3.958e-02   Min.   :    1.066
 1st Qu.:    1.617   1st Qu.:8.995e+00   1st Qu.:    4.237
 Median :    4.240   Median :1.491e+01   Median :    6.161
 Mean   :    4.897   Mean   :1.712e+01   Mean   :    5.753
 3rd Qu.:    7.152   3rd Qu.:2.361e+01   3rd Qu.:    7.301
 Max.   :   42.710   Max.   :5.892e+01   Max.   :   11.715
 NA's   :40816.000   NA's   :4.082e+04   NA's   :40816.000

 est.prevalence.N363.RInf.U0.var est.prevalence.N363.R118.U0.pred
 Min.   :4.288e-03               Min.   :7.145e-01
 1st Qu.:9.745e-01               1st Qu.:4.111e+00
 Median :1.615e+00               Median :6.069e+00
 Mean   :1.855e+00               Mean   :5.807e+00
 3rd Qu.:2.558e+00               3rd Qu.:7.666e+00
 Max.   :6.383e+00               Max.   :1.242e+01
 NA's   :4.082e+04               NA's   :4.082e+04

 est.prevalence.N363.R118.U0.var est.prevalence.N363.R118.U12.pred
 Min.   :5.033e-03               Min.   :   -0.3528
 1st Qu.:1.138e+00               1st Qu.:    3.4219
 Median :1.870e+00               Median :    5.6103
 Mean   :2.094e+00               Mean   :    5.2425
 3rd Qu.:2.907e+00               3rd Qu.:    7.0649
 Max.   :6.161e+00               Max.   :   12.4176
 NA's   :4.082e+04               NA's   :40816.0000
```

```
 est.prevalence.N363.R118.U12.var quality.indicator.N363.R118.U12.pred
 Min.   :5.888e-03                Min.   :  -20.76
 1st Qu.:1.331e+00                1st Qu.:  304.77
 Median :2.188e+00                Median :  666.37
 Mean   :2.449e+00                Mean   :  593.69
 3rd Qu.:3.400e+00                3rd Qu.:  818.85
 Max.   :7.207e+00                Max.   : 1498.96
 NA's   :4.082e+04                NA's   :40816.00

 quality.indicator.N363.R118.U12.var circle.radius.N363.R118.U12.pred
 Min.   :    46.17                   Min.   :    3.006
 1st Qu.: 10580.41                   1st Qu.:   75.258
 Median : 17796.29                   Median :  101.290
 Mean   : 21752.55                   Mean   :   92.755
 3rd Qu.: 29077.42                   3rd Qu.:  112.409
 Max.   :123239.31                   Max.   :  130.503
 NA's   : 40816.00                   NA's   :40816.000

 circle.radius.N363.R118.U12.var
 Min.   :7.916e-01
 1st Qu.:1.814e+02
 Median :3.051e+02
 Mean   :3.729e+02
 3rd Qu.:4.985e+02
 Max.   :2.113e+03
 NA's   :4.082e+04
```

Concernant la prévalence estimée pour *N=363, R=118* et *U=12*, le minimum et le maximum de l'interpolation spatiale correspondent respectivement à -0,3528 et 12,4176 %. La valeur négative du minimum peut surprendre. Elle correspond à des limites de prédiction en des points atypiques et ne doit donc pas être considérée comme une valeur précise localement, valeur qui serait aberrante. Cela doit s'interpréter comme un point de très faible prévalence proche de zéro[8].

On peut indiquer à spplot() les minimum et maximum à prendre en compte pour la représentation graphique à l'aide du paramètre at qui permet d'indiquer les limites des différents niveaux.

```
> spplot(
      cm.krige,
      'est.prevalence.N363.R118.U12.pred',
      cuts=100,
      col.regions=prevR.colors.red(101),
      main='Prévalence du VIH estimée',
      sub='Cameroun - EDS 2004',
      at = seq(-0.3529, 12.4177, length.out=100)
)
```




[8] Dans certains cas, on pourra considérer que les points présentant une prévalence estimée négative doivent être ramenés à la valeur 0. Pour cela, il suffit d'entrer la commande suivante :
`cm.krige$est.prevalence.N363.R118.U12.pred[cm.krige$est.prevalence.N363.R118.U12.pred<0] <- 0`

# 10. Exporter les résultats

## 10.1 Export vers un logiciel de statistiques

La majorité des logiciels de statistiques ainsi que les principaux tableurs tels que Excel ou OpenOffice Calc sont capables de lire des fichiers au format texte ou au format *dbf*.

Il est possible d'exporter facilement des tableaux de données (*data.frame*) ainsi que les résultats de krige.prev() à l'aide des fonctions write.table(), write.dbf() et write.txt().

- write.dbf() permet, comme son nom l'indique d'exporter au format *dbf*.
- write.table() est la fonction générique de R pour exporter au format texte, que soit tabulé ou de type *csv*. Elle dispose de nombreuses options.
- write.txt() permet d'appeler write.table() avec les options adéquates pour un export au format texte tabulé. Autrement dit, write.txt() génère un fichier texte dont les colonnes sont séparées par une tabulation, sans ajout des noms de ligne générés automatiquement par R, sans guillemets encadrant pour les valeurs textes. Seul le séparateur de décimales peut-être modifié. Par défaut, il s'agit du point. Pour certains logiciels, notamment Excel dans sa version française, il est nécessaire d'utiliser la virgule comme séparateur de décimal.

Quelques exemples :

```
> write.dbf(cm.prev,'cm_prev.dbf')
> write.dbf(cm.bounds, 'cm_bounds.dbf')

> write.txt(cm.prev, 'cm_prev.txt')
> write.txt(cm.prev, 'cm_prev_fr.txt', dec=',')
> write.csv(cm.prev, 'cm_prev.csv')
```

Les fichiers *dbf* ont une limitation. En effet, les noms de variables pour ce type de fichiers ne peuvent excéder dix caractères. Ainsi, au moment de l'export avec write.dbf(), les noms de colonnes trop longs du tableau de données seront tronqués. Or, si deux colonnes ont deux noms différents mais ont les mêmes dix premiers caractères, elles porteront le même nom dans le fichier exporté. Cela est fréquent dans un ensemble de données comme cm.prev où *est.prevalence.N338.R118.U12* et *est.prevalence.N338.RInf.Uo* seront tous deux renommés en *est_preval*. Afin de pouvoir différencier ces deux variables dans le fichier exporté, il est préférable de renommer manuellement les variables du tableau de données. Cela se fait nativement sous R à partir de la fonction names() (voir la documentation de cette fonction).

Pour rendre cette opération plus facile, il suffit d'utiliser la fonction check.names() fournie par prevR. Cette fonction vérifie la longueur des noms de colonne. Si un nom est trop long, une fenêtre s'ouvre permettant de modifier les noms de chaque colonne. La longueur des nouveaux noms est elle aussi vérifiée. Pendant cette opération, il est possible de supprimer certaines colonnes en les renommant *NULL*.

```
> str(cm.prev)
'data.frame':   466 obs. of  34 variables:
 $ cluster                          : int  1 10 100 101 102 103 104 105 106 107 ...
 $ x                                : num   9.72 13.57 11.23 14.71 11.55 ...
 $ y                                : num   4.04 10.25  4.74 10.43  3.88 ...
 $ residence                        : Factor w/ 2 levels "Rural","Urban": 2 1 2 1 2 2 2 2 1 2 ...
 $ region                           : num  3 5 2 5 12 7 8 12 7 3 ...
 $ region.name                      : Factor w/ 12 levels "ADAMAOUA","CENTRE",..: 3 5 2 5 12 7 8 ...
 $ longitude                        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ latitude                         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ n                                : num  9 22 18 8 17 36 57 16 16 10 ...
 $ nweight                          : num  10.8 34.8 28.7 12.7 22.1 ...
 $ obs.prevalence                   : num   0.00 13.63  0.00  0.00  5.81 ...
 $ dist.city                        : num    1.87  91.90 102.20  45.20   3.59 ...
 $ city.name                        : chr  "DOUALA" "MAROUA" "YAOUNDE" "MAROUA" ...
 $ urban.area                       : Factor w/ 2 levels "in urban area",..: 1 2 2 2 1 2 2 1 2 1 ...
 $ est.prevalence.N363.R118.U0      : num  5.46 1.94 3.16 2.49 7.61 ...
 $ circle.count.N363.R118.U0        : num  380 369 375 379 367 376 387 370 380 386 ...
 $ circle.radius.N363.R118.U0       : num   3.11 70.93 69.52 61.67  3.73 ...
 $ circle.nb.clusters.N363.R118.U0  : int  17 16 15 15 17 18 18 16 18 18 ...
 $ quality.indicator.N363.R118.U0   : num    0.498 261.887 249.583 195.371   0.727 ...
 $ est.prevalence.N363.RInf.U0      : num  5.46 1.94 3.16 2.49 7.61 ...
 $ circle.count.N363.RInf.U0        : num  380 369 375 379 367 376 387 370 380 386 ...
 $ circle.radius.N363.RInf.U0       : num   3.11 70.93 69.52 61.67  3.73 ...
 $ circle.nb.clusters.N363.RInf.U0  : int  17 16 15 15 17 18 18 16 18 18 ...
 $ quality.indicator.N363.RInf.U0   : num    0.498 261.887 249.583 195.371   0.727 ...
 $ est.prevalence.N363.R118.U12     : num  5.46 1.94 3.16 1.54 7.61 ...
 $ circle.count.N363.R118.U12       : num  380 369 375 384 367 356 387 370 371 386 ...
 $ circle.radius.N363.R118.U12      : num   3.11 70.93 69.52 72.50  3.73 ...
 $ circle.nb.clusters.N363.R118.U12: int  17 16 15 15 17 15 18 16 15 18 ...
 $ quality.indicator.N363.R118.U12 : num    0.498 261.887 249.583 268.231   0.727 ...
 $ est.prevalence.N363.RInf.U12     : num  5.46 1.94 3.16 1.54 7.61 ...
 $ circle.count.N363.RInf.U12       : num  380 369 375 384 367 393 387 370 371 386 ...
 $ circle.radius.N363.RInf.U12      : num   3.11 70.93 69.52 72.50  3.73 ...
 $ circle.nb.clusters.N363.RInf.U12: int  17 16 15 15 17 16 18 16 15 18 ...
 $ quality.indicator.N363.RInf.U12 : num    0.498 261.887 249.583 268.231   0.727 ...
```

```
> cm.prev.check <- check.names(cm.prev, lang='fr')
Certaines variables ont un nom dépassant 10 caractères. Veuillez entrer de nouveaux noms.
Pour supprimer une variable, entrez NULL.
```

**avant saisie**

Editeur de données

Fichier Edition Aide

| | variable | size |
|---|---|---|
| 1 | cluster | 7 |
| 2 | x | 1 |
| 3 | y | 1 |
| 4 | residence | 9 |
| 5 | region | 6 |
| 6 | region.name | 11 |
| 7 | longitude | 9 |
| 8 | latitude | 8 |
| 9 | n | 1 |
| 10 | nweight | 7 |
| 11 | obs.prevalence | 14 |
| 12 | dist.city | 9 |
| 13 | city.name | 9 |
| 14 | urban.area | 10 |
| 15 | est.prevalence.N363.R118.U0 | 27 |
| 16 | circle.count.N363.R118.U0 | 25 |
| 17 | circle.radius.N363.R118.U0 | 26 |
| 18 | circle.nb.clusters.N363.R118.U0 | 31 |
| 19 | quality.indicator.N363.R118.U0 | 30 |
| 20 | est.prevalence.N363.RInf.U0 | 27 |
| 21 | circle.count.N363.RInf.U0 | 25 |
| 22 | circle.radius.N363.RInf.U0 | 26 |
| 23 | circle.nb.clusters.N363.RInf.U0 | 31 |
| 24 | quality.indicator.N363.RInf.U0 | 30 |
| 25 | est.prevalence.N363.R118.U12 | 28 |
| 26 | circle.count.N363.R118.U12 | 26 |
| 27 | circle.radius.N363.R118.U12 | 27 |
| 28 | circle.nb.clusters.N363.R118.U12 | 32 |
| 29 | quality.indicator.N363.R118.U12 | 31 |
| 30 | est.prevalence.N363.RInf.U12 | 28 |
| 31 | circle.count.N363.RInf.U12 | 26 |
| 32 | circle.radius.N363.RInf.U12 | 27 |
| 33 | circle.nb.clusters.N363.RInf.U12 | 32 |
| 34 | quality.indicator.N363.RInf.U12 | 31 |

**après saisie**

Editeur de données

Fichier Edition Aide

| | variable | size |
|---|---|---|
| 1 | cluster | 7 |
| 2 | x | 1 |
| 3 | y | 1 |
| 4 | residence | 9 |
| 5 | region | 6 |
| 6 | reg.name | 11 |
| 7 | NULL | 9 |
| 8 | NULL | 8 |
| 9 | n | 1 |
| 10 | nweight | 7 |
| 11 | obs.prev | 14 |
| 12 | dist.city | 9 |
| 13 | city.name | 9 |
| 14 | urban.area | 10 |
| 15 | eprev.NR | 27 |
| 16 | NULL | 25 |
| 17 | NULL | 26 |
| 18 | NULL | 31 |
| 19 | NULL | 30 |
| 20 | eprev.N | 27 |
| 21 | NULL | 25 |
| 22 | NULL | 26 |
| 23 | NULL | 31 |
| 24 | NULL | 30 |
| 25 | eprev.NRU | 28 |
| 26 | cc.NRU | 26 |
| 27 | cr.NRU | 27 |
| 28 | cnc.NRU | 32 |
| 29 | qual.NRU | 31 |
| 30 | NULL | 28 |
| 31 | NULL | 26 |
| 32 | NULL | 27 |
| 33 | NULL | 32 |
| 34 | NULL | 31 |

```
> str(cm.prev.check)
'data.frame':   466 obs. of  19 variables:
 $ cluster   : int  1 10 100 101 102 103 104 105 106 107 ...
 $ x         : num   9.72 13.57 11.23 14.71 11.55 ...
 $ y         : num   4.04 10.25  4.74 10.43  3.88 ...
 $ residence : Factor w/ 2 levels "Rural","Urban": 2 1 2 1 2 2 2 2 1 2 ...
 $ region    : num  3 5 2 5 12 7 8 12 7 3 ...
 $ reg.name  : Factor w/ 12 levels "ADAMAOUA","CENTRE",..: 3 5 2 5 12 7 8 12 7 3 ...
 $ n         : num  9 22 18 8 17 36 57 16 16 10 ...
 $ nweight   : num  10.8 34.8 28.7 12.7 22.1 ...
 $ obs.prev  : num   0.00 13.63  0.00  0.00  5.81 ...
 $ dist.city : num    1.87  91.90 102.20  45.20   3.59 ...
 $ city.name : chr  "DOUALA" "MAROUA" "YAOUNDE" "MAROUA" ...
 $ urban.area: Factor w/ 2 levels "in urban area",..: 1 2 2 2 1 2 2 1 2 1 ...
 $ eprev.NR  : num  5.46 1.94 3.16 2.49 7.61 ...
 $ eprev.N   : num  5.46 1.94 3.16 2.49 7.61 ...
 $ eprev.NRU : num  5.46 1.94 3.16 1.54 7.61 ...
 $ cc.NRU    : num  380 369 375 384 367 356 387 370 371 386 ...
 $ cr.NRU    : num   3.11 70.93 69.52 72.50  3.73 ...
 $ cnc.NRU   : int  17 16 15 15 17 15 18 16 15 18 ...
 $ qual.NRU  : num    0.498 261.887 249.583 268.231   0.727 ...
```

```
> write.dbf(cm.prev.check, 'cm_prev.dbf')
```

## 10.2 Export vers un logiciel de cartographie (SIG)

La majorité des logiciels permettant de gérer des informations géolocalisées sont en capacité d'importer des fichiers au format *shapefile* et au format *asciigrid*. Ces deux formats ont été initialement développés par la société ESRI pour ses propres produits mais sont devenus des standards.

Extrait de l'encyclopédie Wikipedia (http://fr.wikipedia.org/wiki/Shapefile) :

> Le shapefile, ou "fichier de formes" est un format de fichier issu du monde des Systèmes d'Informations Géographiques (ou SIG). Initialement développé par ESRI pour ses logiciels commerciaux, ce format est désormais devenu un standard de facto, et largement utilisé par un grand nombre de logiciels libres (MapServer, Grass, Udig, MapGuide OpenSource ... ) comme propriétaires.
> 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

Le format *shapefile* permet donc de décrire des points, des lignes ou des polygones. Le format *asciigrid* décrit, quant à lui, une grille de cellules et leurs valeurs.

Le package maptools fournit plusieurs fonctions pour lire et écrire dans ces formats. Le résultat de la fonction krige.prev() peut être directement exporté au format *asciigrid*.

```
> writeAsciiGrid(cm.krige, 'cm_krige_obs_prev.asc')
```

Le format *asciigrid* ne peut contenir qu'une seule grille, tandis que le tableau de données cm.krige en comporte plusieurs. Par défaut, writeAsciiGrid() exporte la première grille. Cependant, on peut lui indiquer, avec le paramètre attr, le nom ou le numéro de la grille à exporter.

L'extension usuelle des fichiers au format *asciigrid* est *.asc*.

```
> writeAsciiGrid(cm.krige, 'cm_krige_est_prev_N363.asc',
attr='est.prevalence.N363.RInf.U0.pred')
> writeAsciiGrid(cm.krige, 'cm_krige_quality_NRU.asc',
              attr=' quality.indicator.N363.R118.U12.pred')
```

Les exports au format *shapefile* sont un peu plus complexes à réaliser, les tableaux de données devant d'abord être convertis dans des formats géographiques.

Deux fonctions permettent d'automatiser ces opérations : write.boundary.shp() permet d'exporter les frontières d'un pays contenu dans un tableau de données à deux colonnes sous la forme d'un *shapefile* comportant un seul polygone. write.prev.shp() permet d'exporter au format *shapefile* un tableau de données où chaque ligne correspond à un point. Si les coordonnées des points ne sont pas contenues dans les colonnes $x$ et $y$, il est possible de spécifier les colonnes adéquates avec le

paramètre coords (voir l'aide de cette fonction). Pour ces deux fonctions, il ne faut pas préciser l'extension des fichiers à créer dans leur nom, elle sera ajoutée automatiquement.

Par ailleurs, les données étant stockées dans des fichiers *dbf*, on retrouve la limitation évoquée plus haut spécifiant que la longueur des noms de colonnes ne doit pas dépasser dix caractères. Par défaut, la fonction check.names() est appliquée au tableau de données fourni à write.prev.shp(). Si le tableau de données comporte des colonnes dont le nom dépasse les dix caractères, alors l'utilisateur sera invité à renommer les noms des colonnes. Il est possible d'appeler write.prev.shp() sans que la longueur des noms de colonnes ne soit vérifiée en lui passant en paramètre check=FALSE.

```
> write.boundary.shp(cm.bounds, 'cm_bounds', 'Cameroun')
> write.prev.shp(cm.prev, 'cm_prev', lang='fr')
> write.prev.shp(cm.cities, 'cm_cities', lang='fr')
```

Les fichiers suivants sont alors créés dans le répertoire de travail :

- *cm_krige.asc,*
- *cm_bounds.shp, cm_bounds.bdf, cm_bounds.shx,*
- *cm_prev.shp, cm_prev.dbf, cm_prev.shx,*
- *cm_cities.shp, cm_cities.dbf* et *cm_cities.shx.*

## 10.3 Importer les résultats dans un SIG

La manière d'importer des fichiers *shapefile* et *asciigrid* diffère selon chaque logiciel. Nous vous renvoyons donc à la documentation spécifique de chacun d'eux.

Les principaux logiciels commerciaux sont en capacité d'importer des fichiers au format *shapefiles* ou *asciigrid*. Vous pouvez aussi avoir recours à des solutions SIG libres et/ou gratuites. Voici une liste des principaux SIG gratuits :

- SavGIS (http://www.savgis.org/). Développé depuis 1984 par l'IRD, SavGIS est distribué gratuitement en français, en anglais et en espagnol.
- GRASS (http://grass.itc.it/). Le plus connu des logiciels libres de cartographie, il est de plus en plus utilisé à travers le monde et permet de lire la majorité des formats de données existants. Par ailleurs, il existe des plugins permettant de faire interagir R avec GRASS.
- Quantum GIS (http://qgis.org/). Disponible en français, ce logiciel dispose d'une interface graphique relativement simple. Il peut lire les principaux formats de données et permet de réaliser facilement des cartes. Relativement simple, il permet de s'initier aux SIG.
- GMT (http://gmt.soest.hawaii.edu/). Il s'agit d'une bibliothèque logicielle permettant de réaliser des cartes vectorielles de haute qualité. Cependant, la prise en main peut être difficile dans la mesure où les cartes doivent être programmées en lignes de commandes.

Pour une présentation plus détaillée, nous vous recommandons d'aller visiter le site *framasoft* : http://www.framasoft.net/rubrique425.html.

# Annexe 1 : exporter un graphique au format SVG

Pour cela, il vous faudra d'abord charger en mémoire le package *RSvgDevice*.

```
> library(RSvgDevice)
```

Ensuite, réaliser votre (vos) graphique(s) de manière habituelle. Si vous avez plusieurs fenêtres graphiques ouvertes, vous verrez que chacune possède un numéro. Dans l'en-tête de la fenêtre est indiqué si cette sortie graphique est actuellement active. Pour afficher la liste des sorties graphiques ouvertes, utilisez dev.list(). Pour savoir qu'elle est la sortie actuellement active, utilisez dev.cur().

```
> dev.list()
windows windows windows
      2       3       4
> dev.cur()
windows
      4
```

Supposons que nous souhaitons exporter en *svg* le graphique de la fenêtre 3. Nous devons tout d'abord rendre la fenêtre 3 active à l'aide de dev.set().

```
> dev.set(3)
windows
      3
> dev.cur()
windows
      3
```

Nous allons ensuite copier le contenu de la fenêtre courante (la 3 en l'occurrence) dans une sortie de type SVG, à l'aide de dev.copy(). Le paramètre devSVG permet de spécifier le type de sortie graphique désirée. Il nous faudra spécifier le nom du fichier *svg* qui doit être créé, *essai.svg* dans notre exemple. Ensuite, nous devrons fermer la sortie SVG créée à l'aide de dev.off(). Le fichier *svg* ne sera généré qu'à ce moment là. On écrira donc :

```
> dev.copy(devSVG, file='essai.svg')
devSVG
     5
> dev.off()
windows
      2
```

Le fichier *essai.svg* sera créé dans le répertoire de travail. Il pourra être lu notamment par Firefox et Inkscape.

**Exemple :** exporter la carte des clusters par milieu de résidence d'Alicante.

```
> data(alicante)
> map.clust(alicante.clust,alicante.bounds,lang='fr')
> library(RSvgDevice)
> dev.copy(devSVG,file='alicante-cluster.svg')
> dev.off()
```

# Annexe 2 : appliquer une transparence avec Inkscape

Inkscape est un logiciel libre de dessin vectoriel, équivalent au logiciel commercial Adobe Illustrator. Il peut être téléchargé gratuitement depuis http://www.inkscape.org/. Il utilise de manière native le format *svg*.

Nous reprendrons ici le graphique réalisé en 5.2, qui aura été préalablement exporté au format *svg* selon la méthode mentionnée dans l'annexe 1.

Ouvrez le fichier *svg* avec Inkscape.

Il nous faut tout d'abord sélectionner l'ensemble des cercles. Pour cela, choisissez *Édition* > *Rechercher*. Entrez *circle* dans le champ *ID*. Cela permettra de sélectionner tous les objets ayant le mot *circle* dans leur identifiant (ce qui est le cas par défaut pour un export depuis R). Cliquez sur *Recherchez* puis fermer la fenêtre de recherche.

Allez dans *Objet* > *Remplissage et contour*… Sous l'onglet *Remplissage*, modifiez le niveau de transparence (paramètre A). Vous pouvez également, si vous le souhaitez, modifier la couleur des cercles, ainsi que l'épaisseur et la couleur des contours des cercles. Dans le résultat présenté ci-dessous, nous avons opté pour une épaisseur des traits des cercles de 0,5 et une transparence de 35.

Le résultat peut ensuite être exporté dans différents formats pour intégration dans un document.