Package 'prevR'

March 12, 2013

Type Package

Title Estimation des tendances régionales d'une prévalence à partir d'une EDS

Version 2.2

Date 2013-03-08

Author Joseph Larmarange - CEPED (Université Paris Descartes Ined IRD) IRD, avec l'appui financier de l'ANRS et l'IRD, et le soutien technique de la SARL LYSIS (info@lysis-consultants.fr)

Maintainer Joseph Larmarange <joseph.larmarange@ird.fr>

Description Ce package fournit des fonctions pour l'estimation spatiale d'une surface de prévalences ou d'une surface de risques relatifs à partir de données issues d'une Enquête Démographique et de Santé (EDS) ou d'une enquête analogue.

License CeCILL

URL http://www.ceped.org/prevR, http://joseph.larmarange.net/prevR

LazyLoad yes

LazyData yes

Depends R (>= 2.10), foreign, methods, sp, gstat, maptools, fields, GenKern, rgdal (>= 0.7-4), geoR, tcltk

Encoding UTF-8

R topics documented:

`prevR-package` *Estimation des tendances régionales d'une prévalence à partir d'une EDS.*

Description

prevR fournit des fonctions pour l'estimation spatiale d'une surface de prévalences ou d'une surface de risques relatifs à partir de données issues d'une Enquête Démographique et de Santé (EDS) ou d'une enquête analogue.

Details

Ce package propose une approche méthodologique pour estimer les tendances régionales d'une prévalence à partir d'enquêtes auprès des ménages échantillonnées en grappes à deux degrés (comme les Enquêtes Démographiques et de Santé). Dans de telles enquêtes, les cas positifs et les cas observés sont positionnés spatialement au niveau de la grappe enquêtée.

Ce package fournit des fonctions pour estimer une surface des prévalences à partir d'estimateurs à noyau à fenêtres adaptatives de même effectif (variante de la méthode des plus proches voisins) ou de même rayon. La surface des prévalences peut également être calculée en procédant à une interpolation spatiale (krigeage ou pondération selon l'inverse de la distance) suite à un lissage préalable des données utilisant des moyennes mobiles basées sur des cercles de même effectif ou de même rayon.

L'approche par les estimateurs à noyau permet également d'estimer une surface de risques relatifs.

Pour une démonstration du package, utilisez la commande `demo(prevR)`.

Le contenu de **prevR** peut être subdivisé ainsi :

Données
`fdhs` est un ensemble de données fictives permettant de tester le package.
`TMWorldBorders` fournit les frontières de chaque pays du monde et peut-être utilisé pour définir les limites de la zone d'étude.

Création d'objet
Les fonctions de **prevR** manipulent des objets de classe `prevR`.
`import.dhs` permet d'importer facilement les données d'une EDS (Enquête Démographique et de Santé), telles que téléchargées sur le site `http://www.measuredhs.com`, en accompagnant l'utilisateur pas à pas.

`as.prevR` est une fonction générique pour créer un objet `prevR`.
`create.boundary` peut être utilisée pour sélectionner les frontières d'un pays et les fournir à `as.prevR` afin de définir la zone d'étude.

Visualisation des données
Les méthodes `show`, `print` et `summary` permettent de visualiser le contenu d'un objet `prevR`.
La méthode `plot` peut être utilisée sur les objets `prevR` pour visualiser la zone d'étude, la position des grappes enquêtées, le nombre d'observations ou le nombre de cas positifs par grappe.

Manipulation des données
La méthode `changeproj` modifie le système de projection des coordonnées.
La méthode `as.data.frame` transforme un objet `prevR` en un tableau de données.
La méthode `export` exporte les données et/ou la zone d'étude au format texte, dbf ou shapefile.

Analyses
`rings` calcule les cercles de même effectif et/ou de même rayon.
`kde` calcule une surface de prévalence ou une surface de risques relatifs à partir d'estimateurs à noyau gaussien à fenêtres adaptatives.
`krige` effectue une interpolation spatiale par krigeage ordinaire.
`idw` calcule une interpolation spatiale en utilisant une pondération selon l'inverse de la distance.

Visualisation et exportation des résultats
`kde`, `krige` et `idw` renvoient des objets de classe `SpatialPixelsDataFrame`{**sp**}.
Ces derniers peuvent être représentés graphiquement à l'aide de la fonction `spplot`{**sp**}.
prevR fournit plusieurs palettes de couleurs (voir `prevR.colors`) pouvant être utilisée avec `spplot`.
Les surfaces calculées peuvent être exportées avec la fonction `writeAsciiGrid`{**maptools**}.

Acknowledgement

prevR a été développé avec l'appui financier de l'Agence Nationale de Recherches sur le Sida et les hépatites virales (ANRS - http://www.anrs.fr) et de l'Institut de Recherche pour le Développement (IRD - http://www.ird.fr), et le soutien technique de la SARL LYSIS (info@lysis-consultants.fr).

Citation

Pour citer **prevR**, l'utilisateur pourra se référer à l'article suivant, en cours de publication : Larmarange J., Vallo R., Yaro S., Msellati P., Meda N., Ferry B. (2011) "Méthodes pour cartographier les tendances régionales de la prévalence du VIH à partir des Enquêtes Démographiques et de Santé (EDS)", *Cybergeo: European Journal of Geography*, http://cybergeo.revues.org/.

Author(s)

Joseph Larmarange <joseph.larmarange@ird.fr>
CEPED (Université Paris Descartes Ined IRD) - IRD

References

Larmarange J., Vallo R., Yaro S., Msellati P., Meda N., Ferry B. (2011) "Méthodes pour cartographier les tendances régionales de la prévalence du VIH à partir des Enquêtes Démographiques et de Santé (EDS)", *Cybergeo: European Journal of Geography*, http://cybergeo.revues.org/.

Larmarange J. (2007) *Prévalences du VIH en Afrique : validité d'une mesure*, thèse de doctorat en démographie, sous la direction de Benoît Ferry, université Paris Descartes, http://tel.archives-ouvertes.fr/tel-00320283.

Larmarange J., Vallo R., Yaro S., Msellati P., Meda N., Ferry B. (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.

Examples

```
# Creation d'un objet de classe prevR
col <- c(id = "cluster",
         x = "x",
         y="y",
         n="n",
         pos = "pos",
         c.type = "residence",
         wn="weighted.n",
         wpos="weighted.pos"
        )
dhs <- as.prevR(fdhs.clusters,col, fdhs.boundary)

str(dhs)
print(dhs)

plot(dhs, main="Clusters position",new.window=FALSE)
plot(dhs, type="c.type", main="Clusters by residence",new.window=FALSE)
plot(dhs, type="count", main="Observations by cluster",new.window=FALSE)
plot(dhs, type="c.type", main="Positive cases by cluster",new.window=FALSE)


# Changement de systeme de coordonnees
plot(dhs,axes=TRUE,new.window=FALSE)
dhs <- changeproj(dhs,
       "+proj=utm +zone=30 +ellps=WGS84 +datum=WGS84 +units=m +no_defs")
print(dhs)
plot(dhs,axes=TRUE,new.window=FALSE)

# Calcul des cercles de meme effectif pour differentes valeurs de N
dhs <- rings(fdhs,N=c(100,200,300,400,500))
print(dhs)
summary(dhs)

# Surface des prevalence pour N=300
prev.N300 <- kde(dhs,N=300,nb.cells=200)
spplot(prev.N300, 'k.wprev.N300.RInf',
       cuts=100, col.regions=prevR.colors.red(101),
       main="Regional trends of prevalence (N=300)"
      )

# Surface des rayons des cercles de meme effectif
radius.N300 <- krige('r.radius',dhs,N=300,nb.cells=200)
spplot(radius.N300,
       cuts=100, col.regions=prevR.colors.blue(101),
       main="Radius of circle (N=300)"
```

)

`as.data.frame.prevR` *Convertir un objet prevR en tableau de données.*

Description

Cette fonction permet de fusionner les éléments `clusters` et `rings` d'un objet `prevR` en un unique tableau de données.

Usage

```
## S3 method for class 'prevR'
as.data.frame(x, ..., N = NULL, R = NULL, clusters.only = FALSE)
```

Arguments

`x`	objet de classe `prevR` à exporter.
`...`	non utilisé, pour compatibilité avec la méthode générique `as.data.frame`.
`N`	entier ou liste d'entiers indiquant les éléments de `rings` à extraire.
`R`	entier ou liste d'entiers indiquant les éléments de `rings` à extraire.
`clusters.only`	renvoyer uniquement l'élément `clusters` de `x` ?

Value

Si `clusters.only = TRUE`, la fonction renverra simplement l'élément `clusters` de `x`.

Sinon, les éléments `clusters` et `rings` de `x` seront fusionnés en un seul tableau de données. Les variables de `rings` seront renommées en leur ajoutant un suffixe de la forme *.N300.RInf*.

`N` et `R` permettent de sélectionner les éléments de `rings` à exporter. S'ils ne sont pas spécifiés (valeur `NULL`), tous les éléments de `rings` seront inclus.

See Also

`as.data.frame`{**base**}, `prevR-class`.

Examples

```
r.fdhs <- rings(fdhs,N=c(100,200,300))
str(r.fdhs)
str(as.data.frame(r.fdhs,clusters.only=TRUE))
str(as.data.frame(r.fdhs))
str(as.data.frame(r.fdhs,N=300))
```

`as.prevR` *Création d'objet prevR.*

Description

Cette fonction permet de créer un objet de la classe `prevR` à partir d'un tableau de données.

Usage

```
as.prevR(data, col, boundary = NULL, proj = "+proj=longlat")
```

Arguments

`data`	un tableau de données, chaque ligne correspondant à une grappe observée.
`col`	un vecteur permettant d'identifier les variables de `data` à utiliser pour construire l'élément `clusters` de l'objet `prevR`. Les noms des colonnes de `clusters` sont normalisés et ont pour valeur : • "id" identifiant de la grappe. • "x" longitude de la grappe. • "y" latitude de la grappe. • "n" nombre d'observations. • "pos" nombre des cas positifs. • "wn" (facultatif) somme des poids individuels des observations. • "wpos" (facultatif) somme des poids individuels des cas positifs. • "c.type" (facultatif) variable de classification des grappes (utilisée seulement par `plot`). Voir les exemples.
`boundary`	objet de classe `SpatialPolygons` définissant les limites de la zone d'étude.
`proj`	projection dans laquelle les coordonnées des grappes sont exprimées dans `data` (longitude / latitude par défaut).

Details

Seuls "x", "y" "n" et "pos" sont obligatoires dans `col`. Si "id" n'est pas spécifié, un identifiant numérique sera créé.

`proj` définie la projection utilisée par `data`. Il peut s'agir d'une chaîne de caractères correspondant à une projection *PROJ.4* (voir http://trac.osgeo.org/proj/ pour plus de détails) ou un objet de classe `CRS`{**sp**}.
Si la projection de `boundary` est précisée dans son élément `proj4string`, `boundary` sera transformée dans la projection définie par `proj`. Si l'élément `proj4string` est manquant, `boundary` sera considéré comme étant défini dans le système de coordonnées de `proj`.

Si `boundary` n'est pas précisé (`NULL`), un rectangle défini par les coordonnées maximales et minimales de `data` sera calculé.
`boundary` peut être le résultat d'un appel à la fonction `create.boundary`.

`as.prevR` ne permet pas de changer de projection. Pour cela, utilisez `changeproj` sur le résultat de `as.prevR`.

Value

Un objet de classe prevR (voir la documentation de la classe pour plus de détails).

See Also

prevR-class, create.boundary, changeproj, import.dhs.

Examples

```
col <- c(id = "cluster",
         x = "x",
         y="y",
         n="n",
         pos = "pos",
         c.type = "residence",
         wn="weighted.n",
         wpos="weighted.pos"
        )
dhs <- as.prevR(fdhs.clusters,col, fdhs.boundary)

str(dhs)
print(dhs)
```

---

as.SpatialGrid,prevR-method

Création d'une grille à partir d'un objet prevR.

Description

Cette fonction permet de calculer une grille de points régulièrement répartis à partir de l'élément `boundary` d'un objet prevR. Elle est notamment appelée en interne par les méthodes kde, krige et idw.

Usage

```
as.SpatialGrid(object, nb.cells = 100, cell.size = NULL)
```

Arguments

`object`	objet de classe prevR.
`nb.cells`	nombre de cellules sur la plus grande dimension (inutilisé si `cell.size` est défini).
`cell.size`	côté d'une cellule (dans l'unité de la projection).

Details

Cette fonction crée une grille de points régulièrement espacés, chaque cellule étant carrée et ayant un côté égal à `cell.size`. Si `cell.size` n'est pas défini, le côté des cellules sera défini en divisant le plus grand côté de l'élément `boundary` de `object` par `nb.cells`.

Value

Objet de classe SpatialGrid{**sp**}.

See Also

GridTopology{**sp**}, SpatialGrid-class{**sp**}.

Examples

```
str(as.SpatialGrid(fdhs))
str(as.SpatialGrid(fdhs, nb.cells=200))
```

---

changeproj,prevR-method
Changer la projection d'un objet prevR.

Description

Cette fonction permet de modifier la projection d'un objet prevR.

Usage

```
## S4 method for signature 'prevR'
changeproj(object, proj)
```

Arguments

object	objet de classe prevR.
proj	projection de destination.

Details

proj peut être une chaîne de caractères correspondant à une projection *PROJ.4* (voir http://trac.osgeo.org/proj/ pour plus de détails) ou un objet de classe CRS{**sp**}.

changeproj modifie les colonnes "x" et "y" de l'élément clusters de object et projette boundary dans le nouveau système de coordonnées.
Le cas échéant, l'élément rings est recalculé.

Value

Renvoie object exprimé dans la projection proj.

See Also

spTransform{**rgdal**}, prevR-class.

Examples

```
print(fdhs)
plot(fdhs,axes=TRUE,main="Projection : longitude/latitude")

fdhs2 <- changeproj(fdhs,
           "+proj=utm +zone=30 +ellps=WGS84 +datum=WGS84 +units=m +no_defs")
print(fdhs2)
dev.new()
plot(fdhs2,axes=TRUE, main="Projection: UTM Zone 30")
```

`create.boundary` *Fournit les frontières d'un pays.*

Description

Cette fonction embarque le jeu de données `TMWorldBorders`. Elle permet de sélectionner un pays et le renvoie sous la forme d'un objet de classe `SpatialPolygons`.

Usage

```
create.boundary(countries = NULL, multiple = F, proj = "+proj=longlat")
```

Arguments

`countries`	un vecteur de chaînes de caratères contenant le nom des pays que vous désirez extraire du jeu de données. Si `NULL`, une boite de dialogue sera affichée pour sélectionner le pays désiré.
`multiple`	la boîte de dialogue doit-elle permettre de sélectionner plusieurs pays (inutilisé si `countries` est fourni) ?
`proj`	projection dans laquelle le résultat doit être exprimé (en longitude/latitude par défaut).

Details

`proj` peut être une chaîne de caractères correspondant à une projection *PROJ.4* (voir http://trac.osgeo.org/proj/ pour plus de détails) ou un objet de classe `CRS`{**sp**}.

Value

Objet de classe `SpatialPolygons`{**sp**}.

See Also

`TMWorldBorders`.

Examples

```
## Not run:
boundary <- create.boundary()

## End(Not run)

boundary <- create.boundary("Burkina Faso")
boundary <- create.boundary("Burkina Faso",
  proj="+proj=utm +zone=30 +ellps=WGS84 +datum=WGS84 +units=m +no_defs")
boundary <- create.boundary(countries = c("Burkina Faso", "Ghana", "Benin"))
```

`export,prevR-method` *Exporte un objet prevR.*

Description

Cette méthode permet d'exporter un objet de classe `prevR` dans différents formats.

Usage

```
export(object, element,
       format, file,
       N = NULL, R = NULL, clusters.only = FALSE,
       ext = NULL, sep = NULL, dec = NULL, ...
      )
```

Arguments

`object`	objet de classe `prevR`.
`element`	élément à exporter : "clusters" ou "boundary".
`format`	format d'exportation : "dbf", "txt", csv", "csv2" ou "shp" (inutilisé si `element="boundary"`).
`file`	nom du fichier à créer (sans extension).
`N`	entier ou liste d'entiers indiquant les éléments de `rings` à extraire (inutilisé si `element="boundary"`).
`R`	entier ou liste d'entiers indiquant les éléments de `rings` à extraire (inutilisé si `element="boundary"`).
`clusters.only`	renvoyer uniquement l'élément `clusters` de `object` (inutilisé si `element="boundary"`)?
`ext`	forcer l'extension du fichier créé (inutilisé si `element="boundary"` ou si `format="shp"`).
`sep`	forcer le séparateur de champs (inutilisé si `element="boundary"` ou si `format="shp"` ou si `format="dbf"`).
`dec`	forcer le séparateur de décimal. (inutilisé si `element="boundary"` ou si `format="shp"` ou si `format="dbf"`).
`...`	arguments additionnels transmis à `writePolyShape`, `writePointsShape`, `write.dbf` ou `write.table`.

Details

Si `element="boundary"`, l'élément `boundary` est exporté au format *shapefile*.

Sinon, c'est l'élément `clusters`, fusionné avec `rings`, qui sera fusionné.
Voir `as.data.frame.prevR` pour l'utilisation des paramètres `N`, `R` et `clusters.only`.
`format` permet de spécifier le format d'export du tableau de données obtenu avec `as.data.frame.prevR` :

"shp"	Shape File
"dbf"	DBASE format (extension: .dbf)
"txt"	texte tabulé (extension: .txt)
"csv"	'comma separated values' (extension : .csv)
"csv2"	Variante de CSV utilisant un point-virgule (extension: .csv)

`ext` permet de forcer l'extension du fichier produit, excepté pour les exports en *shapefile* qui produisent systématiquement trois fichiers (.shp, .shx et .dbf).

Le format "txt" utilise par défaut une tabulation comme séparateur et un point pour indiquer les décimales. Le format "csv" utilise une virgule comme séparateur et un point pour les décimales. Le format "csv2" est une variante utilisant le point-virgule comme séparateur et la virgule pour les décimales, convention utilisée par certaines versions d'Excel (notamment la version française). `sep` et `dec` permettent de forcer le séparateur et le point décimal à utiliser (en conjonction avec le format "txt").

Methods

```
signature(object = "prevR")
```

See Also

`writePolyShape` {**maptools**}, `writePointsShape` {**maptools**}, `write.dbf` {**foreign**}, `write.table` {**utils**}.

Examples

```
## Not run:
export(fdhs, element="boundary", file="area")
export(fdhs, element="clusters", format="shp", file="points")

dhs <- rings(fdhs,N=c(100,300,500))
export(dhs, element="clusters", format="csv", N=300, file="points")

## End(Not run)
```

`fdhs` *Données issues d'une simulation d'EDS.*

Description

Jeu de données issu d'une simulation d'Enquête Démographique et de Santé (EDS) sur un pays fictif présentant une prévalence nationale de 10 % avec 8000 personnes enquêtées réparties en 401 grappes. Il contient trois objets :

- `fdhs.clusters` : tableau de données aggrégées par grappe.
- `fdhs.boundary` : objet de classe `SpatialPolygons` définissant les frontières du pays fictif.
- `fdhs` : objet de classe `prevR` basé sur les 2 jeux de données précédents et obtenu avec `as.prevR`.

Usage

```
fdhs
```

References

Larmarange J., Vallo R., Yaro S., Msellati P., Meda N., Ferry B. (2011) "Méthodes pour cartographier les tendances régionales de la prévalence du VIH à partir des Enquêtes Démographiques et de Santé (EDS)", *Cybergeo: European Journal of Geography*, http://cybergeo.revues.org/.

Larmarange J. (2007) *Prévalences du VIH en Afrique : validité d'une mesure*, thèse de doctorat en démographie, sous la direction de Benoît Ferry, université Paris Descartes, http://tel.archives-ouvertes.fr/tel-00320283.

Larmarange J., Vallo R., Yaro S., Msellati P., Meda N., Ferry B. (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.

Examples

```
str(fdhs)
str(fdhs.clusters)
str(fdhs.boundary)
demo(prevR)
```

`import.dhs` *Importer les données d'une EDS dans prevR.*

Description

Cette fonction guide l'utilisateur pas à pas pour importer les données d'une Enquête Démographique et de Santé (EDS) et créer un objet de classe `prevR`.

Usage

```
import.dhs(file.sav, file.dbf)
```

Arguments

`file.sav`	fichier contenant les données individuelles d'une Enquête Démographique et de Santé (EDS), téléchargé au format SPSS (.sav) sur http://www.measuredhs.com.
`file.dbf`	fichier contenant les données GPS d'une Enquête Démographique et de Santé (EDS), téléchargé au format DATABASE (.dbf) sur http://www.measuredhs.com.

Note

Si vous ne précisez pas le chemin des fichiers, R inspectera le dossier de travail (voir `setwd`). Pour spécifier le chemin des fichiers, voir `file.path`.

Cette fonction est destinée spécifiquement à l'importation des Enquêtes Démograhiques et de Santé. Pour créer un objet de classe `prevR` de manière plus générique, voir `as.prevR`.

See Also

as.prevR, prevR-class.

Examples

```
## Not run:
import.dhs("data.sav", "gps.dbf")

## End(Not run)
```

`is.prevR` *Tester si un objet est de classe prevR.*

Description

Cette fonction permet de tester si un objet est de classe `prevR`. Elle permet aussi de tester la présence des éléments `rings` et `boundary`.

Usage

```
is.prevR(object, slot = NULL)
```

Arguments

`object`	un objet quelconque.
`slot`	un vecteur de chaînes de caractères pouvant contenir "clusters", "rings","boundary" ou "proj".

Details

Attention les éléments `rings` et `boundary` sont toujours présents dans un objet de classe `prevR`, mais `rings` peut être une liste `NULL` et `boundary` un `SpatialPolygons` avec un attribut `valid` posé à `FALSE` (lorsque les limites de la zone d'étude n'ont pas été définies explicitement).

- Si `rings` est une liste `NULL`, `is.prevR(object,"rings")` renverra `FALSE`.
- Si `boundary` a un attribut `valid` égal à `FALSE`, `is.prevR(object,"boundary")` renverra `FALSE`.

Value

Cette fonction renvoie un vecteur logique.

See Also

prevR-class.

Examples

```
col <- c(id = "cluster",
         x = "x",
         y="y",
         n="n",
         pos = "pos",
         c.type = "residence",
         wn="weighted.n",
         wpos="weighted.pos"
        )
dhs <- as.prevR(fdhs.clusters,col, fdhs.boundary)

is.prevR(dhs)
is.prevR(dhs,"rings")
is.prevR(dhs,"boundary")

dhs <- rings(dhs,N=300)
is.prevR(dhs,"rings")
```

---

kde,prevR-method *Estimateurs à noyau pour les objets prevR.*

Description

Cette fonction permet de calculer une surface de prévalences (ratio de deux surfaces d'intensité) et/ou une surface de risques relatifs (ratio de deux surfaces de densité) en utilisant des estimateurs à noyau gaussien à fenêtres adaptatives de même effectif et/ou de même rayon.

Usage

```
## S4 method for signature 'prevR'
kde(object,
    N = NULL, R = NULL, weighted = TRUE,
    risk.ratio = FALSE, keep.details = FALSE,
    nb.cells = 100, cell.size = NULL,
    progression=TRUE
   )
```

Arguments

`object`	objet de classe `prevR`.
`N`	entier ou liste d'entiers indiquant les cercles à utiliser.
`R`	entier ou liste d'entiers indiquant les cercles à utiliser.
`weighted`	utiliser les données pondéréses (TRUE, FALSE ou "2") ?
`risk.ratio`	calculer une surface de risques relatifs au lieu d'une surface de prévalences (TRUE, FALSE ou "2") ?
`keep.details`	renvoyer les surfaces des cas positifs et des cas observés (TRUE, FALSE) ?
`nb.cells`	nombre de cellules sur la plus grande dimension (voir `as.SpatialGrid`).
`cell.size`	côté d'une cellule (dans l'unité de la projection, voir `as.SpatialGrid`).
`progression`	afficher une barre de progression des calculs ?

Details

Cette fonction calcule une surface des prévalences en faisant le ratio de la surface d'intensité (exprimée en cas par unité de surface) des cas positifs sur la surface d'intensité des cas observés. Elle peut également calculer une surface de risques relatifs qui correspod au ratio de la surface de densité (dont l'intégrale a été ramenée à l'unité) des cas positifs sur la surface de densité des cas observés.

La méthode utilisée ici est une variante de la technique des plus proches voisins. Il s'agit d'une estimation par noyau gaussien à fenêtres adaptatives, la taille des fenêtres étant déterminée par un effectif minimum d'observations dans le voisinage (voir `rings` pour plus de détails). Il est également possible d'utiliser des fenêtres de taille fixe. Plus précisément, la fenêtre utilisée correspond à la moitié du rayon des cercles de même effectif et/ou de même rayon (paramètres `N` et `R`) calculés avec la fonction `rings`.
Voir les références pour une explication plus détaillée de la méthodologie implémentée.

`N` et `R` permettent de sélectionner les cercles à utiliser pour l'estimation des prévalences. S'ils ne sont pas définis, les surfaces de prévalences seront estimées pour chacun des couples (N,R) disponibles. Plusieurs interpolations peuvent être réalisées simultanément en transmettant une liste de plusieurs valeurs de N et R.

Value

Objet de classe `SpatialPixelsDataFrame`. Les surfaces estimées sont nommées à partir du nom de la variable interpolée, de N et de R (exemple : *k.prev.N300.RInf*).

Les variables crées sont (selon les arguments passés à la fonction) :

- "k.pos" surface d'intensité non pondérée des cas positifs.
- "k.obs" surface d'intensité non pondérée des cas observés.
- "k.prev" surface non pondérée des prévalences (k.pos/k.obs).
- "k.case" surface de densité non pondérée des cas positifs.
- "k.control" surface de densité non pondérée des cas observés.
- "k.rr" surface non pondérée des risques relatifs (k.case/k.control).
- "k.wpos" surface d'intensité pondérée des cas positifs.
- "k.wobs" surface d'intensité pondérée des cas observés.
- "k.wprev" surface pondérée des prévalences (k.wpos/k.wobs).
- "k.wcase" surface de densité pondérée des cas positifs.
- "k.wcontrol" surface de densité pondérée des cas observés.
- "k.wrr" surface pondérée des risques relatifs (k.wcase/k.wcontrol).

La valeur `NA` est appliquée aux points de la grille situés en-dehors de la zone d'étude (voir `NA.outside.SpatialPolygons`).

Methods

```
signature(object = "prevR")
```

Note

Les résultats peuvent être représentés graphiquement à l'aide de la fonction spplot{**sp**}. **prevR** fournit plusieurs palettes de couleurs (voir prevR.demo.pal) pouvant être utilisées avec spplot.

Les surfaces calculées peuvent être exportées avec la fonction writeAsciiGrid{**maptools**}.

Voir le package **sparr** pour une autre approche méthodologique de calcul d'une surface de risques relatifs, adaptée à d'autres types de données que les Enquêtes Démographiques et de Santé (EDS).

References

Larmarange J., Vallo R., Yaro S., Msellati P., Meda N., Ferry B. (2011) "Méthodes pour cartographier les tendances régionales de la prévalence du VIH à partir des Enquêtes Démographiques et de Santé (EDS)", *Cybergeo: European Journal of Geography*, http://cybergeo.revues.org/.

See Also

KernSur{**GenKern**}, rings,prevR-method.

Examples

```
dhs <- rings(fdhs,N=c(100,200,300,400,500))

prev.N300 <- kde(dhs,N=300,nb.cells=200)
spplot(prev.N300, 'k.wprev.N300.RInf',
       cuts=100, col.regions=prevR.colors.red(101),
       main="Regional trends of prevalence (N=300)"
      )

prev.krige <- kde(dhs,N=c(100,300,500),R=Inf,
                  nb.cells=200, risk.ratio=2, keep.details=FALSE
                 )
str(prev.krige)
dev.new()
spplot(prev.krige,
       c('k.wprev.N100.RInf','k.wprev.N300.RInf','k.wprev.N500.RInf'),
       cuts=100, col.regions=prevR.colors.red(101)
      )
```

`krige,prevR-method` *Interpolation spatiale (krigeage et distance inverse) pour les objets prevR.*

Description

Ces fonctions permettent d'interpoler spatialement les variables d'un élément `rings` d'un objet de classe prevR. La méthode `krige` effectue un krigeage ordinaire tandis que la méthode `idw` procède à une interpolation pondérée par l'inverse de la distance.

Usage

```
## S4 method for signature 'ANY,prevR'
krige(formula, locations,
      N = NULL, R = NULL, model = NULL,
      nb.cells = 100, cell.size = NULL,
      fit = "auto", keep.variance = FALSE, show.variogram = FALSE,
      ...
     )
## S4 method for signature 'ANY,prevR'
idw(formula, locations,
    N = NULL, R = NULL,
    nb.cells = 100, cell.size = NULL,
    idp = 2,
    ...
   )
```

Arguments

`formula`	variable à interpoler (voir détails).
`locations`	objet de classe `prevR`.
`N`	entier ou liste d'entiers indiquant les cercles à utiliser.
`R`	entier ou liste d'entiers indiquant les cercles à utiliser.
`model`	modèle de variogramme, défini par un appel à la fonction `vgm`{**gstat**}.
`nb.cells`	nombre de cellules sur la plus grande dimension (voir `as.SpatialGrid`).
`cell.size`	côté d'une cellule (dans l'unité de la projection, voir `as.SpatialGrid`).
`fit`	"auto" pour un ajustement automatique du semi-variogramme à partir des données, "manual" pour un ajustement graphique (inutilisé si `model` est fournir).
`keep.variance`	renvoyer la variance des estimations ?
`show.variogram`	afficher graphiquement le semi-variogramme utilisé ?
`idp`	puissance de la pondération selon l'inverse de la distance (voir `idw`{**gstat**}).
`...`	arguments additionnels transmis à `krige`{**gstat**} ou `idw`{**gstat**}.

Details

`formula` permet de spécifier la ou les variables à interpoler. Il s'agit des variables disponibles dans l'élément `rings` de l'objet de classe `prevR` transmis. Les valeurs possibles sont "r.pos", "r.n", "r.prev", "r.radius", "r.clusters", "r.wpos", "r.wn" ou "r.wprev". Le nom des variables peut être indiqué sous forme d'une chaîne de caractères ou sous la forme de formules (exemple : *list(r.pos~1,r.prev~1)*). Seul les formules de la forme *une.variable~1* sont acceptées. Pour des interpolations plus complexes, vous devez appeler manuellemet les fonctions `krige` et `idw` de **gstat**.

`N` et `R` permettent de sélectionner les cercles à utiliser pour récupérer la variable à interpoler. S'ils ne sont pas définis, la variable `formula` sera interpolée pour chacun des coupes (N,R) disponibles. Plusieurs interpolations peuvent être réalisées simultanément en transmettant une liste de plusieurs variables et/ou plusieurs valeurs de N et R.

Dans le cas d'un krigeage ordinaire, la méthode `krige` de **prevR** ajustera automatiquement un modèle (exponentiel) de semi-variogramme au semi-variogramme expérimental

(`fit="auto"`). Si vous choisissez `fit="manual"`, la fonction affichera une fenêtre graphique (adaptée de `eyefit`{**geoR**}) vous permettant d'ajuster visuellement le modèle de semi-variogramme au semi-variogramme expérimental. Vous pouvez également définir explicitement le modèle de semi-variogramme à utiliser avec le paramètre `model`.

Les interpolations sont effectuées sur une grille de points uniformément répartis, obtenue avec `as.SpatialGrid`.

Value

Objet de classe `SpatialPixelsDataFrame`. Les surfaces estimées sont nommées à partir du nom de la variable interpolée, de N et de R (exemple : *r.radius.N300.RInf*). Si la variance des estimations a été demandée à `krige` (`keep.variance=TRUE`), elle sera indiquée avec le suffixe *.var*.

La valeur `NA` est appliquée aux points de la grille situés en-dehors de la zone d'étude (voir `NA.outside.SpatialPolygons`).

Methods

`signature(formula = "ANY", locations = "prevR")`

Note

Les résultats peuvent être représentés graphiquement à l'aide de la fonction `spplot`{**sp**}. **prevR** fournit plusieurs palettes de couleurs (voir `prevR.colors`) pouvant être utilisées avec `spplot`.

Les surfaces calculées peuvent être exportées avec la fonction `writeAsciiGrid`{**maptools**}.

References

Larmarange J., Vallo R., Yaro S., Msellati P., Meda N., Ferry B. (2011) "Méthodes pour cartographier les tendances régionales de la prévalence du VIH à partir des Enquêtes Démographiques et de Santé (EDS)", *Cybergeo: European Journal of Geography*, http://cybergeo.revues.org/.

See Also

`krige`{**gstat**}, `idw`{**gstat**}, `rings,prevR-method`.

Examples

```
dhs <- rings(fdhs,N=c(100,200,300,400,500))
radius.N300 <- krige('r.radius',dhs,N=300,nb.cells=200)
spplot(radius.N300,
       cuts=100, col.regions=prevR.colors.blue(101),
       main="Radius of circle (N=300)"
      )

## Not run:
prev.krige <- krige('r.wprev',dhs,N=c(100,300,500),R=Inf,
                    fit="manual",keep.variance=TRUE
                   )
str(prev.krige)
spplot(prev.krige,
       c('r.wprev.N100.RInf','r.wprev.N300.RInf','r.wprev.N500.RInf'),
       cuts=100, col.regions=prevR.colors.red(101)
```

```
      )

## End(Not run)
```

`NA.outside.SpatialPolygons` *Affecter une valeur manquante aux points situés en-dehors d'un polygone.*

Description

Cette fonction affecte la valeur `NA` aux points d'un objet de classe `SpatialPixelsDataFrame` situés en dehors d'un objet `SpatialPolygons`.

Usage

```
NA.outside.SpatialPolygons(sp.data, sp.poly)
```

Arguments

`sp.data`	objet de classe `SpatialPixelsDataFrame`.
`sp.poly`	objet `SpatialPolygons`.

Value

Renvoie `sp.data` modifié.

See Also

`point.in.SpatialPolygons`.

`plot,prevR-method` *Graphiques pour les objets prevR.*

Description

Méthode `plot` pour les objets de classe `prevR`. Affiche la position des grappes, le nombre d'observations ou le nombre de cas positifs.

Usage

```
## S4 method for signature 'prevR,missing'
plot(x,
     type = "position",
     add.legend = TRUE,
     legend.location = "bottomright",
     factor.size = 0.2,
     new.window = TRUE,
     axes = F,
     ...
    )
```

Arguments

`x`	objet de classe `prevR`.
`type`	graphique à afficher : • ”position” position des grappes. • ”c.type” grappes selon `c.type`. • ”count” nombre d’observations par grappe. • ”flower” nombre de cas positifs par grappe.
`add.legend`	ajouter une légende ?
`legend.location`	position de la légende.
`factor.size`	facteur d’agrandissement/réduction des cercles (pour `type="count"`).
`new.window`	afficher dans une nouvelle fenêtre ?
`axes`	afficher les axes ?
`...`	arguments additionnels transmis à `title`.

Details

L’argument `legend.location` peut recevoir les valeurs suivantes : *”bottomright”*, *”bottom”*, *”bottomleft”*, *”left”*, *”topleft”*, *”top”*, *”topright”*, *”right”* et *”center”*.

Utilisez, entre autres, l’argument `main` pour définir un titre et `sub` pour un sous-titre.

Methods

`signature(x = "prevR", y = "missing")`

See Also

`title`{**graphics**}, `legend`{**graphics**}.

Examples

```
plot(fdhs,type = "position",main="position",new.window=FALSE,axes=TRUE)
plot(fdhs,type = "c.type",main="c.type",new.window=FALSE)
plot(fdhs,type = "count",main="count",factor.size = 0.1,new.window=FALSE)
plot(fdhs,type = "flower",main="flower",new.window=FALSE)
```

---

point.in.SpatialPolygons

Détermine si un point appartient à un polygone.

Description

Cette fonction teste si un point ou plusieurs points sont situés à l’intérieur d’un objet `SpatialPolygons`.

Usage

`point.in.SpatialPolygons(point.x, point.y, SpP)`

Arguments

`point.x`	coordonnées en x des points à tester.
`point.y`	coordonnées en y des points à tester.
`SpP`	objet `SpatialPolygons`

Value

Renvoie un vecteur logique.

See Also

`point.in.polygon`{**sp**}, `NA.outside.SpatialPolygons`.

`prevR-class` *Objets de classe prevR.*

Description

Classe utilisée par le package **prevR**

Objects from the Class

Les objets de cette classe peuvent être créés à l'aide de la fonction `as.prevR`.

Slots

`clusters` objet `data.frame` contenant les données observées avec une ligne par grappe. Les colonnes sont nommées :

- "id" identifiant de la grappe.
- "x" longitude.
- "y" latitude.
- "n" nombre d'observations valides dans la grappe.
- "pos" nombre de cas positifs observés dans la grappe.
- "prev" prévalence (en %) observée dans la grappe.
- "wn" (facultatif) somme des poids des observations de la grappe.
- "wpos" (facultatif) somme des poids des cas positifs de la grappe.
- "wprev" (facultatif) prévalence pondérée (en %) observée dans la grappe.
- "c.type" (facultatif) variable de classification des grappes.

`boundary` objet de classe `SpatialPolygons` définissant les limites de la zone d'étude.

`proj` objet de classe `CRS` définissant le système de coordonnées utilisé.

`rings` liste contenant les résultats de la fonction `rings`. Chaque entrée de la liste est composée de trois éléments : `N`, effectif minimum des cercles ; `R`, rayon maximum des cercles et `estimates`, un tableau de données contenant les variables suivantes :

- "id" identifiant de la grappe.
- "r.pos" nombre de cas positifs dans le cercle.
- "r.n" nombre d'observations dans le cercle.
- "r.prev" prévalence (en %) observée dans le cercle.

- "r.radius" rayon du cercle (en kilomètres si les coordonnées sont exprimées en degrés décimaux, dans l'unité de la projection sinon).
- "r.clusters" nombre de grappes situées dans le cercle.
- "r.wpos" (facultatif) somme des poids des cas positifs situés dans le cercle.
- "r.wn" (facultatif) somme des poids des observations situées dans le cercle.
- "r.wprev" (facultatif) prévalence pondérée (en %) observée dans le cercle.

Remarque : la liste `rings` est nommée, le nom de chacun de ses éléments étant de la forme N(valeur de N).R(valeur de R), par exemple *N300.RInf*.

Methods

as.data.frame `signature(x = "prevR")` transforme un objet prevR en tableau de données.

as.SpatialGrid `signature(object = "prevR")` calcule une grille de points.

export `signature(object = "prevR")` exporte un objet prevR en shapefile, fichier dbf ou fichier texte.

idw `signature(formula = "ANY", locations = "prevR")` effectue une interpolation spatiale selon une pondération inverse de la distance.

kde `signature(object = "prevR")` estime une surface de prévalence par des estimateurs à noyaux

krige `signature(formula = "ANY", locations = "prevR")` effecture une interpolatio spatiale par krigeage ordinaire.

plot `signature(x = "prevR", y = "ANY")` représente graphiquement les données d'un objet prevR.

print `signature(x = "prevR")` affiche le résumé d'un objet prevR.

rings `signature(object = "prevR")` calcule des cercles de même effectif et/ou de même rayon.

show `signature(object = "prevR")` affiche le résumé d'un objet prevR.

summary `signature(object = "prevR")` affiche un résumé des variables d'un objet prevR.

changeproj `signature(object = "prevR")` modifie le système de coordonnées d'un objet prevR.

See Also

`as.prevR`, `is.prevR`, `changeproj,prevR-method`, `rings,prevR-method`, `print,prevR-method`, `plot,prevR-method`, `summary,prevR-method`, `kde,prevR-method`, `krige,prevR-method`, `idw,prevR-method`, `export,prevR-method`.

Examples

```
showClass("prevR")

col <- c(id = "cluster",
         x = "x",
         y="y",
         n="n",
         pos = "pos",
         c.type = "residence",
         wn="weighted.n",
         wpos="weighted.pos"
```

```
        )
dhs <- as.prevR(fdhs.clusters,col, fdhs.boundary)
str(dhs)
print(dhs)

dhs <- rings(fdhs,N=c(100,300,500))
str(dhs)
print(dhs)
```

`prevR.colors` *Palettes de couleurs continues.*

Description

Fonctions générant des palettes de couleurs utilisables par les fonctions graphiques de R, en particulier `spplot`. Elles créent des palettes de couleurs continues, les contrastes étant renforcés par l'éclaircissement ou l'assombrissement des valeurs extrêmes. `prevR.demo.pal` permet d'afficher les différentes palettes de couleurs disponibles. `prevR.colors.qgis.pal` permet d'exporter une palette dans un fichier texte lisible par le logiciel libre de cartographie Quantum GIS.

Usage

```
prevR.demo.pal(n, border, main, ch.col)
prevR.colors.red(n)
prevR.colors.red.inverse(n)
prevR.colors.blue(n)
prevR.colors.blue.inverse(n)
prevR.colors.green(n)
prevR.colors.green.inverse(n)
prevR.colors.gray(n)
prevR.colors.gray.inverse(n)
prevR.colors.qgis.pal(file, at, pal="red", inverse=FALSE)
```

Arguments

`n`	nombre de couleurs devant constiter la palette.
`border`	couleur de la bordure.
`main`	titre du graphique.
`ch.col`	liste des palettes à afficher.
`file`	nom du fichier à créer avec son extension.
`at`	liste des valeurs de la palette.
`pal`	palette de couleur à utiliser (”red”, ”green”, ”blue” ou ”gray”).
`inverse`	utiliser la palette inverse ?

Details

`prevR.colors.red` réalise un dégradé allant du blanc/jaune au rouge/rouge foncé.
`prevR.colors.blue` réalise un dégradé allant du bleu pâle au bleu foncé.
`prevR.colors.green` réalise un dégradé allant du vert pâle au vert foncé.
`prevR.colors.gray` réalise un dégradé allant du blanc/gris clair au gris foncé/noir.

Les fonctions avec le suffixe *.inverse* réalisent les mêmes dégradés mais en partant des couleurs foncées vers les couleurs claires.

Value

`prevR.demo.pal` affiche les différentes palettes.

`prevR.colors.qgis.pal` exporte la palette de couleurs dans un fichier texte utilisable par Quantum GIS.

Les autres fonctions renvoient une liste de couleurs codées de manière hexadécimale.

Note

Pour obtenir la liste des couleurs au format RGB (Red/Green/Blue), utilisez la fonction `col2rgb`{**grDevices**}.

Le code de `prevR.demo.pal` a été repris sur celui de la fonction `demo.pal` décrite dans les exemples de la documentation de `rainbow`.

See Also

D'autres palettes de couleurs existent sous R. Voir `rainbow`{**grDevices**} ainsi que le package **RColorBrewer**.

Examples

```
prevR.demo.pal(25)
prevR.colors.red(5)
col2rgb(prevR.colors.red(5))

## Not run:
prevR.colors.qgis.pal('palette.txt',seq(0,25,length.out=100),'red')

## End(Not run)
```

`print,prevR-method` *Affiche le résumé d'un objet prevR.*

Description

Méthode `print` pour les objets de classe `prevR` : affiche un résumé des caractéristiques de l'objet.

Usage

```
## S4 method for signature 'prevR'
print(x)
```

Arguments

`x`	objet de classe `prevR`.

Methods

`signature(x = "prevR")`

Note

Identique à `show,prevR-method`.

See Also

`summary,prevR-method`.

Examples

```
print(fdhs)
dhs <- rings(fdhs,N=c(100,300,500))
print(dhs)
```

`rings,prevR-method` *Calculer des cercles de même effectif et/ou de même rayon.*

Description

Cette fonction détermine pour chaque grappe un cercle de même effectif et/ou de même rayon et calcule plusieurs indicateurs à partir des observations situées dans le cercle.

Usage

```
rings(object, N = seq(100, 500, 50), R = Inf, progression = TRUE)
```

Arguments

`object`	objet de classe `prevR`.
`N`	effectif minimum des cercles.
`R`	rayon maximum des cercles (en kilomètres si `object` est exprimé en longitude/latitude, dans l'unité de la projection sinon).
`progression`	afficher une barre de progression des calculs ?

Details

Pour chaque grappe de l'élément `clusters` de `object`, `rings` détermine un cercle, centré sur la grappe. Il peut s'agir :

- d'un cercle de même effectif si `N` est fini et `R=Inf` ;
- d'un cercle de rayon fixe si `N=Inf` et `R` est fini ;
- d'un cercle mixte si si `N` et `R` sont finis.

Pour les *cercles de même effectif*, `rings` sélectionne le plus petit cercle tel que le nombre d'observations valides dans ce cercle soit au moins égal à `N`.
Pour les *cercles de même rayon*, `rings` sélectionne toutes les grappes situées à une distance inférieure à `R` de la grappe centrale.
Pour les *cercles mixtes*, `rings` calcule tout d'abord le cercle d'effectif minimum et vérifie si son rayon est inférieur à `R`. Si oui, il conserve ce cercle, sinon il calcule le cercle de rayon maximum.

Il est possible de calculer différentes séries de cercles en transmettant plusieurs valeurs aux arguments `N` et `R`. `rings` calculera alors les cercles correspondant à chaque couple (N,R).

Value

Renvoie `object` en complétant son élément `rings` pour chaque couple (N,R) calculé.

Chaque entrée de la liste `rings` est composé de trois éléments : `N`, effectif minimum des cercles ; `R`, rayon maximum des cercles et `estimates`, un tableau de données contenant les variables suivantes :

- "id" identifiant de la grappe.
- "r.pos" nombre de cas positifs dans le cercle.
- "r.n" nombre d'observations dans le cercle.
- "r.prev" prévalence (en %) observée dans le cercle.
- "r.radius" rayon du cercle (en kilomètres si les coordonnées sont exprimées en degrés décimaux, dans l'unité de la projection sinon).
- "r.clusters" nombre de grappes situées dans le cercle.
- "r.wpos" (facultatif) somme des poids des cas positifs situés dans le cercle.
- "r.wn" (facultatif) somme des poids des observations situées dans le cercle.
- "r.wprev" (facultatif) prévalence pondérée (en %) observée dans le cercle.

Les variables *r.wpos*, *r.wn* et *r.wprev* ne sont calculées que si l'élément `clusters` de `object` contient des données pondérées.

References

Larmarange J., Vallo R., Yaro S., Msellati P., Meda N., Ferry B. (2011) "Méthodes pour cartographier les tendances régionales de la prévalence du VIH à partir des Enquêtes Démographiques et de Santé (EDS)", *Cybergeo: European Journal of Geography*, http://cybergeo.revues.org/.

Larmarange J. (2007) *Prévalences du VIH en Afrique : validité d'une mesure*, thèse de doctorat en démographie, sous la direction de Benoît Ferry, université Paris Descartes, http://tel.archives-ouvertes.fr/tel-00320283.

Larmarange J., Vallo R., Yaro S., Msellati P., Meda N., Ferry B. (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.

See Also

prevR-class.

Examples

```
print(fdhs)
dhs <- rings(fdhs,N=c(100,200,300,400,500))
print(dhs)
```

`show,prevR-method` *Affiche le résumé d'un objet prevR.*

Description

Méthode `show` pour les objets de classe `prevR` : affiche un résumé des caractéristiques de l'objet.

Usage

```
## S4 method for signature 'prevR'
show(object)
```

Arguments

`object`	objet de classe `prevR`.

Methods

```
signature(object = "prevR")
```

Note

Identique à `print,prevR-method`.

See Also

`summary,prevR-method`.

Examples

```
fdhs
dhs <- rings(fdhs,N=c(100,300,500))
dhs
```

`summary,prevR-method` *Affiche un résumé des variables contenues dans un objet prevR.*

Description

Méthode `summary` pour les objets de classe `prevR` : affiche un résumé des variables de l'objet.

Usage

```
## S4 method for signature 'prevR'
summary(object,probs = c(0,.10,.25,.50,.75,.80,.90,.95,.99,1))

prevRsummary(object, probs = c(0,.10,.25,.50,.75,.80,.90,.95,.99,1))
```

Arguments

`object`	objet de classe `prevR`.
`probs`	vecteur de probabilités pour le calcul des quantiles des rayons des cercles.

Methods

```
signature(object = "prevR")
```

See Also

`print,prevR-method`.

Examples

```
summary(fdhs)
dhs <- rings(fdhs,N=c(100,300,500))
summary(dhs)
```

`TMWorldBorders` *Jeu de données "TM World Borders Dataset 0.3".*

Description

Ce jeu de données contient les frontières de l'ensemble des pays du monde, exprimées en longitude/latitude. Les attributs disponibles sont :

- "FIPS" code pays FIPS 10-4.
- "ISO2" code pays ISO 3166-1 Alpha-2.
- "ISO3" code pays ISO 3166-1 Alpha-3.
- "UN" code pays ISO 3166-1 Numeric-3.
- "NAME" nom du pays/zone (en anglais).
- "AREA" superficie des terres, FAO Statistics (2002).
- "POP2005" population, World Population Prospects (2005).
- "REGION" région continentale, UN Statistics.

- "SUBREGION" sous-région, UN Statistics.
- "LON" longitude.
- "LAT" latitude.

Usage

```
TMWorldBorders
```

Format

Objet de classe `SpatialPolygonsDataFrame`.

Note

Les frontières et désignations employées n'impliquent l'expression d'aucune opinion ni acceptation de la part des auteurs.

Source

Le jeu de données *World Borders Dataset 0.3* a été élaboré par Bjorn Sandvik, Schuyler Erle et Sean Gilles à partir de sources dans le domaine publique. Ce jeu de données, disponible sur `http://thematicmapping.org/downloads/world_borders.php`, est placé sous licence *Creative Commons Attribution-Share Alike* (`http://creativecommons.org/licenses/by-sa/3.0/`).

Examples

```
plot(TMWorldBorders)
```

`xyz2dataframe` *Transforme une surface définie en xyz en un tableau de données.*

Description

Plusieurs fonctions (par exemple `KernSur`{**GenKern**}) gèrent les surfaces sous la forme d'une liste, dite xyz, de trois éléments : un vecteur avec les coordonnées en x, un vecteur avec les coordonnées en y et une matrice avec les valeurs de la surface en x et y. La présente fonction transforme une liste xyz en un tableau de données.

Usage

```
xyz2dataframe(xyz, xcol = 1, ycol = 2, zcol = 3)
```

Arguments

`xyz`	une liste contenant 3 éléments : un élément vecteur contenant les coordonnees sur x, un élément vecteur contenant les coordonnees sur y et un élément matrice contenant des valeurs en chaque point de coordonnées x[i],y[j].
`xcol`	indice de x.
`ycol`	indice de y.
`zcol`	indice de z.

Value

Un `data.frame`.

Note

`xyz` peut être une liste de la forme `x,y,z1,z2,z3`. L'argument `zcol` sera alors égal à `c("z1","z2","z3")` ou `c(3,4,5)`.

Examples

```
x <- c(2,4,6,8,10)
y <- x
op <- KernSur(x,y, xgridsize=50, ygridsize=50,
              correlation=0,
              xbandwidth=1, ybandwidth=1,
              range.x=c(0,13), range.y=c(0,13)
             )
str(op)

op.df <- xyz2dataframe(op)
str(op.df)
```

Index