

Analyse quantitative du paysage sous R

Jean-Pierre Rossi (INRA - CBGP - Montpellier)

Claire Lavigne (INRA - PSH - Avignon)

TD du 19 janvier 2016

Contents

Objectifs de la séance	2
1-Décollage	2
1-1 Chargement du package	2
1.2 Jeux de données	2
data()	3
Accès direct	3
2 Opérations basiques sur les objets raster	5
2.1 Changement de résolution : aggregate	5
2.2 Découpage	6
2.3 Opérations sur les valeurs d'un raster	7
2.4 Remplacement de certaines valeurs	8
3 Calcul des métriques de paysage	9
3.1 Métriques de classe	9
3.2 Métriques de patch	9
4 Découpage d'un buffer circulaire centré sur un point	10
4.1 Positionner un point au hasard dans r	10
4.2 Découpage <i>per se</i>	10
4.3 Calcul des métriques de classe sur le buffer	10
4.4 Extraction et analyse d'une série de buffers distribués sur une maille régulière	10
4.5 Extraction et analyse d'une série de paysages carrés distribués sur une maille régulière	10
4.5.1 Création de la grille	11
4.5.2 Création de fenêtres juxtaposées et extraction des paysages	11
4.5.3 Création de fenêtres juxtaposées et extraction des paysages	11
4.5.4 Calcul des métriques	11
4.5.5 Réarrangement des sorties	11
4.5.6 Extraction des valeurs de la métrique percentage of landscape pour le landuse "prairie"	11
4.5.7 Représentation graphique des résultats sous forme de raster	11

4.5.8 Conclusions !	11
4.6 Analyse des paysages extraits sur la grille	11
4.6.1 Gestion des données : obtention d'un data.frame	11
4.6.2 Analyse multivariable	11
4.6.3 Analyse multivariable - cartographie	12
4.6.4 Analyse spatiale	12
4.6.5 Clustering	12

Objectifs de la séance

Produire des données quantitatives (variables aléatoires) décrivant la structure et la composition des paysages

Explorer quelques pistes de traitements simples

Points abordés

- Présentation de quelques ressources R particulièrement importantes
- Comment lire des données cartographiques ?
- Opérations basiques sur les objets **raster**
- Calcul des métriques paysagères
- Emprise et position de l'aire d'étude
- Effets contrastés suivant les métriques : pourquoi ?
- Redondances entre métriques

1-Décollage

1-1 Chargement du package

Le package ECPaysage a été développé spécifiquement pour la formation :

```
#install.packages("ECPaysage", repos="http://R-Forge.R-project.org", type="source")
library(ECPaysage)
```

```
## Vous avez chargé la version 2.5.4 rev 30 du package ECPaysage. Bravo !
```

Normalement, les participants ont également installé une série de packages qui sont nécessaires pour la suite de la formation. Il s'agit notamment de : **sp**, **rgdal**, **SDMTools**, **maptools**, **ade4**, **gstat** et **geoR**.

```
library(raster)
```

```
## Loading required package: sp
```

1.2 Jeux de données

ECPaysage contient différents jeux de données utiles pour les TD de l'école. On peut y accéder de deux façons : fonction **data** ou accès direct

`data()`

```
data(xy104pt)
```

xy104pt est un objet de la classe `SpatialPoints` (package `sp`)

```
xy104pt
```

```
## class      : SpatialPoints
## features   : 104
## extent     : 735787.9, 775780.1, 6277958, 6301287  (xmin, xmax, ymin, ymax)
## coord. ref. : NA
```

Accès direct

Les données sont contenues dans le dossier `extdata` qui se trouve sur votre disque dur dans le dossier d'installation de R. On y accède facilement en suivant le chemin indiqué par la fonction `system.file`

Tapez le code suivant pour savoir où se trouve ce dossier :

```
system.file("extdata", package="ECPaysage")
```

```
## [1] "/Library/Frameworks/R.framework/Versions/3.2/Resources/library/ECPaysage/extdata"
```

Pour lire un fichier contenu dans le dossier `extdata` il faut connaître le chemin complet vers ce fichier. Par exemple, pour le fichier `r.tif` la fonction `system.file` indique :

```
system.file("extdata/r.tif", package="ECPaysage")
```

```
## [1] "/Library/Frameworks/R.framework/Versions/3.2/Resources/library/ECPaysage/extdata/r.tif"
```

On peut lire ce fichier pour comme on le ferait pour n'importe quel autre fichier geotiff :

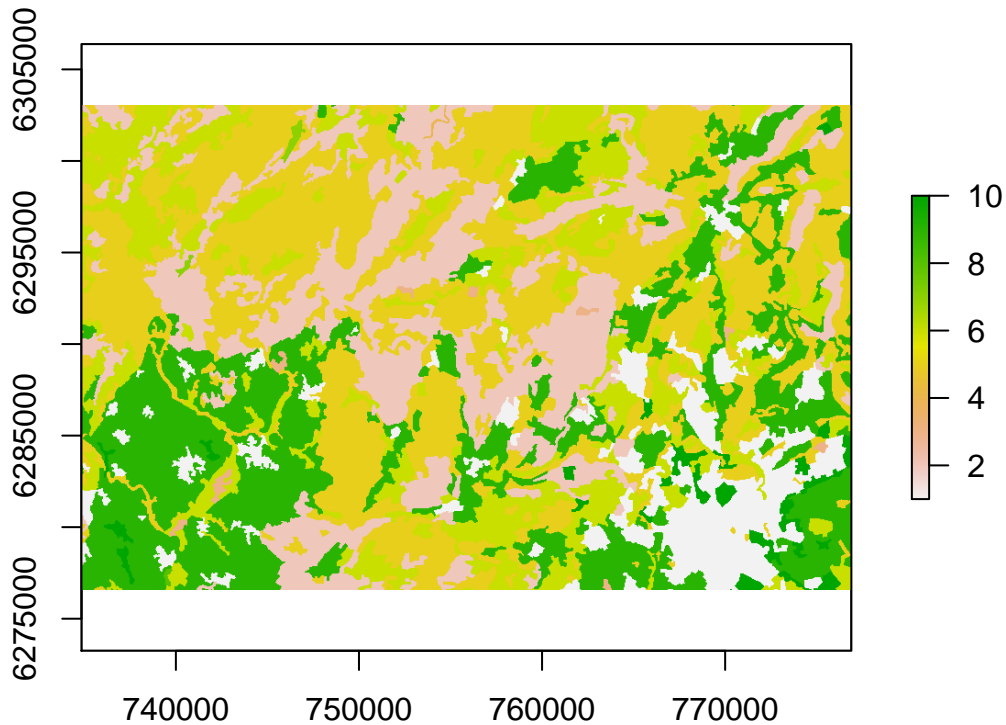
```
r <- raster(system.file("extdata/r.tif", package="ECPaysage"))
```

r est un objet de la classe `RasterLayer` (package `raster`)

```
r
```

```
## class      : RasterLayer
## dimensions : 528, 841, 444048  (nrow, ncol, ncell)
## resolution : 50, 50  (x, y)
## extent     : 734843, 776893, 6276616, 6303016  (xmin, xmax, ymin, ymax)
## coord. ref. : NA
## data source : /Library/Frameworks/R.framework/Versions/3.2/Resources/library/ECPaysage/extdata/r.tif
## names      : r
## values     : 1, 10  (min, max)
```

```
plot(r)
```



ECPaysage contient une série de fichiers (liste ci-dessous) dont le shapefile `contours` :

```
library(maptools)
```

```
## Checking rgeos availability: TRUE
```

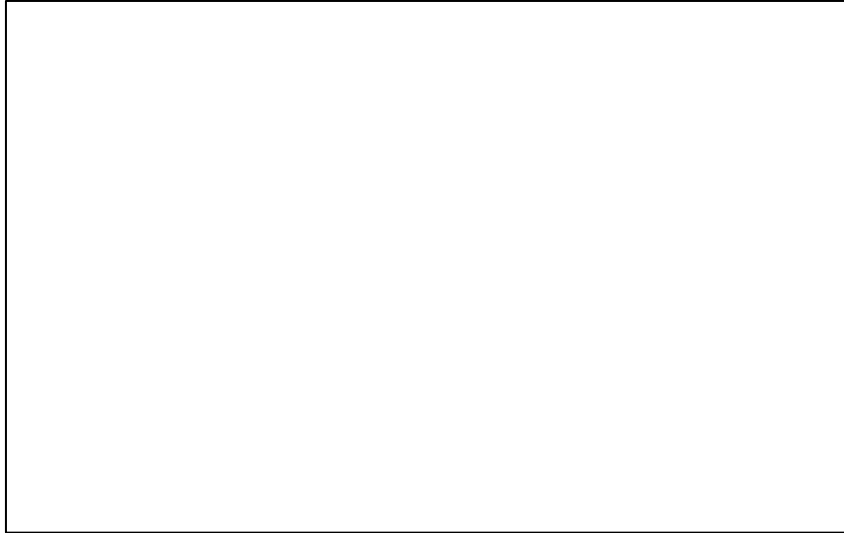
```
#gpclibPermit()
```

```
contours <- readShapeSpatial(system.file("extdata/contours.shp", package="ECPaysage"))  
contours
```

```
## class      : SpatialPolygonsDataFrame  
## features   : 1  
## extent     : 734843, 776887, 6276599, 6303016 (xmin, xmax, ymin, ymax)  
## coord. ref.: NA  
## variables  : 3  
## names      : POINTA, POINTB, POINTC  
## min values : 1, 2, 3  
## max values : 1, 2, 3
```

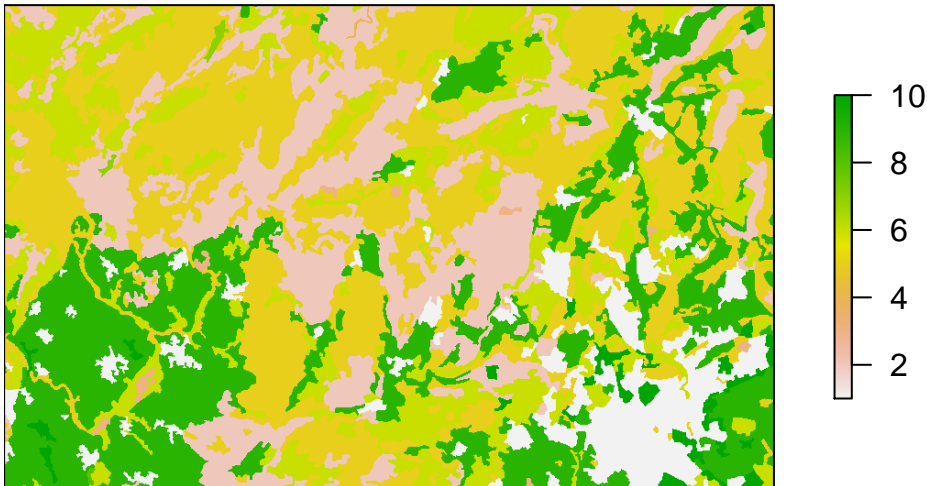
`contours` est un objet de la class `SpatialPolygonsDataFrame` (package `sp`)

```
plot(contours)
```



RasterLayer et SpatialPolygonsDataFrame peuvent être superposés avec la fonction `plot` de `raster`:

```
plot(r, axes=F, addfun=plot(contours, add=TRUE), box=F)
```



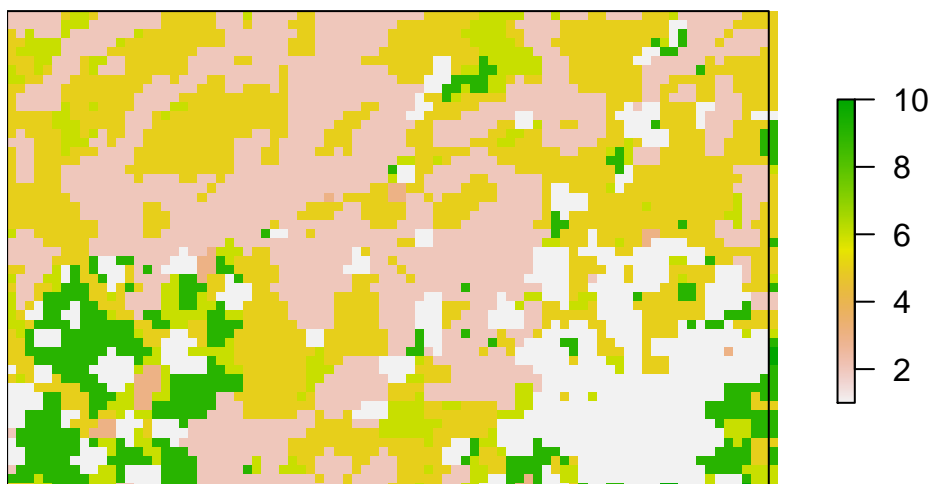
2 Opérations basiques sur les objets raster

2.1 Changement de résolution : aggregate

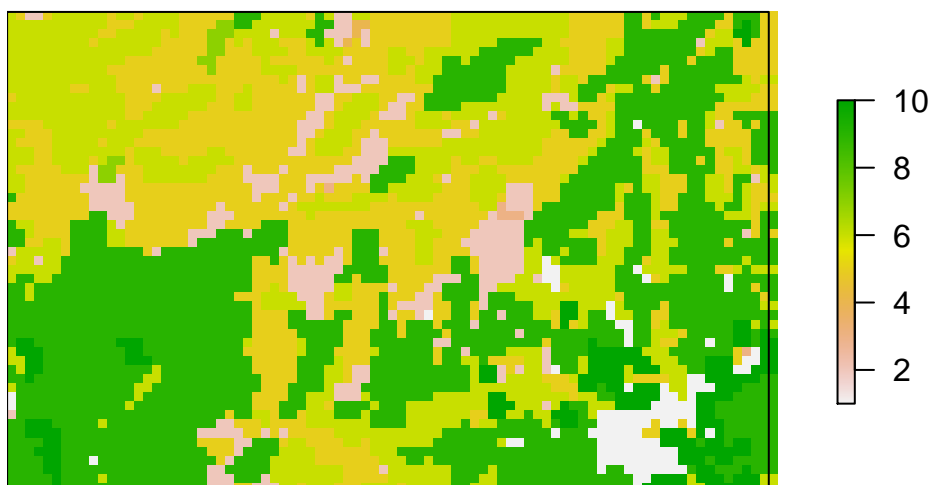
```
data(r)
```

```
## Warning in data(r): data set 'r' not found
```

```
ra <- aggregate(r, fact=10, fun=min)
plot(ra, axes=F, addfun=plot(contours, add=TRUE), box=F)
```



```
rb <- aggregate(r, fact=10, fun=max)
plot(rb, axes=F, addfun=plot(contours, add=TRUE), box=F)
```



2.2 Découpage

```
crop(x, y, filename="", snap='near', datatype=NULL, ...)
```

```
r <- raster(nrow=45, ncol=90)
r[] <- 1:ncell(r)
length(r[])
```

```
## [1] 4050
```

```
e <- extent(-160, 10, 30, 60)
rc <- crop(r, e)
length(rc[])
```

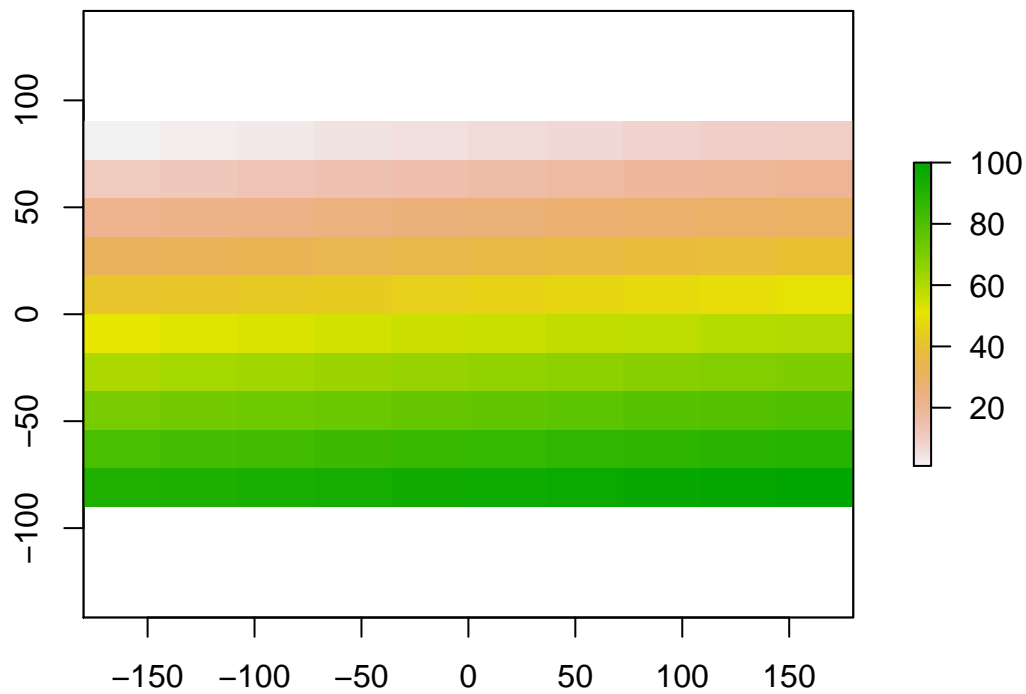
```
## [1] 294
```

2.3 Opérations sur les valeurs d'un raster

```
r <- raster(nrow=10, ncol=10)
r[] <- 1:ncell(r)
mean(r[])
```

```
## [1] 50.5
```

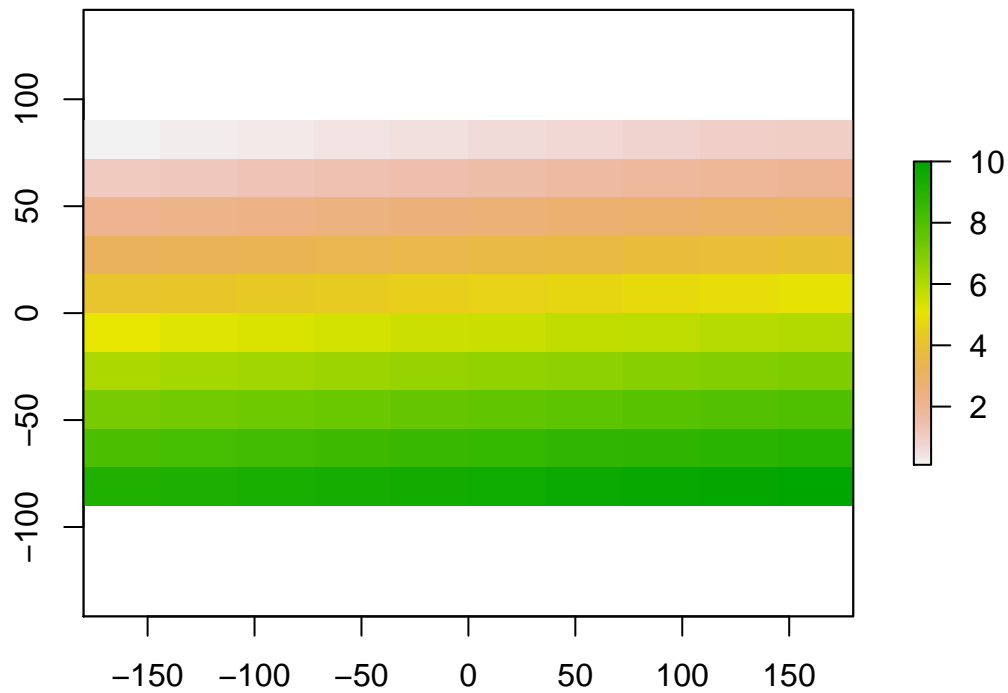
```
plot(r)
```



```
r[] <- r[] / 10
mean(r[])
```

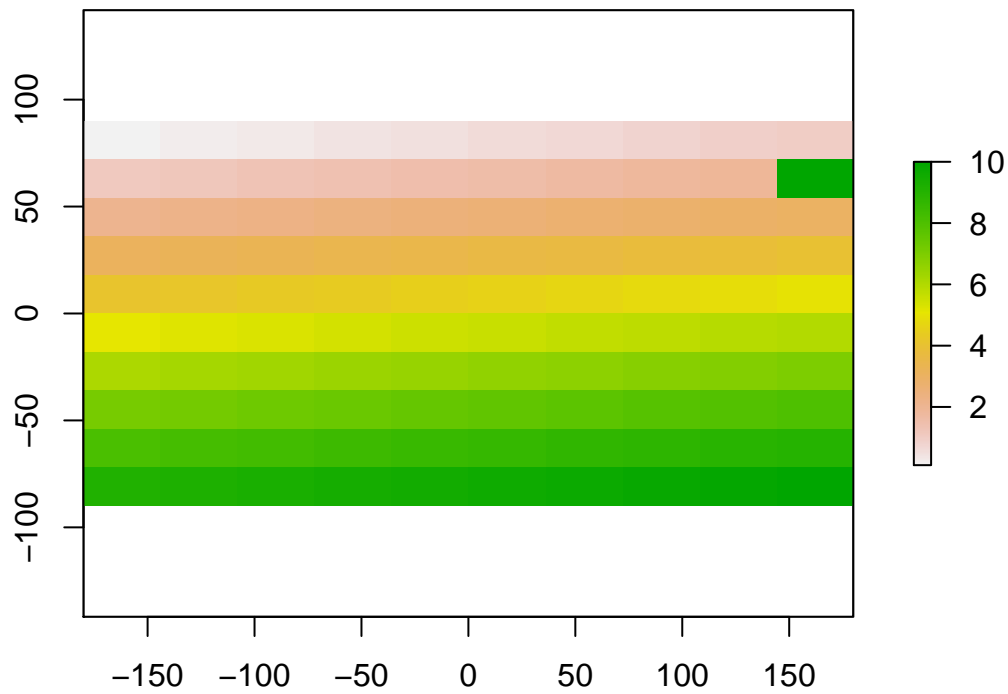
```
## [1] 5.05
```

```
plot(r)
```



2.4 Remplacement de certaines valeurs

```
w <- which(r[] == 2)
r[w] <- 10
plot(r)
```



3 Calcul des métriques de paysage

L'analyse est réalisée ici sur des objets **raster**. D'autres solutions existent.

?ClassStat

Sous R le package **SDMTools** permet le calcul des métriques de patch et de classe. D'autres packages proposent ces métriques. Pour le moment, les métriques de paysage (*landscape metrics* dans Fragstats) ne sont pas encore implémentées sous R.

```
library(SDMTools)
```

```
##  
## Attaching package: 'SDMTools'  
##  
## The following object is masked from 'package:raster':  
##  
## distance
```

- PatchStat : métriques de patch
- ClassStat : métriques de classe

3.1 Métriques de classe

Calculez la métrique “percentage of landscape” pour chaque classe du raster **r**. `vecnoms <- c(“Bati”, “Broussailles”, “Carriere decharge”, “Eau libre”, “Foret”, “Prairie”, “Rocher eboulis”, “Sable gravier”, “Vigne verger”, “Zone d’activites”)`

3.2 Métriques de patch

Calculez les métriques de patch pour la classe 1 “bâti” raster **r**.

```
?PatchStat  
?ConnCompLabel
```

Calculez les métriques de patch pour la classe 1 “bâti” raster **r**.

-préparer la matrice binaire pour le landuse 1 (`matbi`)

-utiliser cette matrice avec `ConnCompLabel` pour obtenir la matrice identifiant les patches (`matpatch`)

-calculer les métriques avec la matrice identifiant les patches

Cartographiez les valeurs de la métrique `shape.index` calculées pour les patches du landuse “bâti” dans le raster **r**.

-Combien y a-t-il de patches différents pour le landuse “bâti” ?

-Quelle matrice contient le n° de chaque patch ?

4 Découpage d'un buffer circulaire centré sur un point

4.1 Positionner un point au hasard dans **r**

4.2 Découpage *per se*

Découpez un buffer circulaire de 7000 m centré sur le point

Listez dans la console les différentes valeurs contenues dans **buf**

Les valeurs qui nous intéressent sont celles contenues dans **r** pour les pixels prenant une valeur de 1 dans **buf**.

Comment les isoler ?

Quelles sont les dimensions des 2 rasters ?

Comment découper la partie qui nous intéresse ?

- Il existe d'autres méthodes (avec d'autres fonctions) qui aboutissent au même résultat
- On peut refaire le travail sur les $n = 104$ points avec une boucle par exemple

4.3 Calcul des métriques de classe sur le buffer

4.4 Extraction et analyse d'une série de buffers distribués sur une maille régulière

```
sample <- spsample(contours,n=100,type="regular", offset=c(00.0,00.0))
s <- as.data.frame(sample)
head(s)
```

```
##           x1          x2
## 1 734843.0 6276599
## 2 738175.7 6276599
## 3 741508.4 6276599
## 4 744841.0 6276599
## 5 748173.7 6276599
## 6 751506.4 6276599
```

```
?xy104pt
data(xy104pt)
s<-data.frame(xy104pt)
```

Comment extraire les 104 buffers ?

4.5 Extraction et analyse d'une série de paysages carrés distribués sur une maille régulière

Les 104 points pourraient représenter des points d'échantillonnage. Une forme d'analyse du paysage **r** consiste à créer des pixel carrés distribués sur une maille carrée couvrant **r**.

Chaque pixel correspond à un paysage "local" qui est analysé comme les buffers précédents.

L'analyse de l'ensemble des pixels avec les métriques de paysage fournit une représentation de la variation spatiale des structures au sein de la zone d'étude. C'est utile (par exemple) pour positionner des placettes d'étude dans des zones homogènes du point de vue de telle(s) ou telle(s) caractéristique(s) du paysage.

Exercice : A partir du paysage `r` mettez en oeuvre cette approche sur les points définis par `spsample(contours,n=100,type="regular", offset=c(01.0,01.0))`. Pour simplifier l'analyse, n'utilisez que la métrique `percentage of landscape` des prairies (`landuse=6`).

4.5.1 Création de la grille

4.5.2 Création de fenêtres juxtaposées et extraction des paysages

4.5.3 Création de fenêtres juxtaposées et extraction des paysages

4.5.4 Calcul des métriques

4.5.5 Réarrangement des sorties

4.5.6 Extraction des valeurs de la métrique `percentage of landscape` pour le `landuse` "prairie"

4.5.7 Représentation graphique des résultats sous forme de raster

4.5.8 Conclusions !

L'analyse précédente débouche sur de nombreuses options (statistiques spatiales)

Attention, la seule composante pertinente n'est pas forcément la métrique `percentage of landscape` pour le `landuse` "prairie" !

⇒ Quelle(s) stratégie(s) intégrer les autres métriques ?

4.6 Analyse des paysages extraits sur la grille

4.6.1 Gestion des données : obtention d'un `data.frame`

Il y a des `NA`. Il faut les retirer, du moins avant certaines analyses. Remplacez les `NA` se trouvant dans `df` ; essayez d'utiliser `sapply`.

4.6.2 Analyse multivariable

Chargez le package `ade4` et utilisez la fonction `dudi.pca` pour faire l'ACP de `df2`

Tracez le cercle des corrélations avec `s.corcircle`.

Plusieurs variables sont superposées. Elles apportent des informations redondantes (du moins du point de vue des axes 1 et 2).

Visualisez les coordonnées `pca$co` avec `table.paint`.

Refaites l'ACP en n'utilisant que les variables `"prop.landscape"`.

4.6.3 Analyse multivariable - cartographie

Chaque axe de l'ACP propose une synthèse de l'information extraite des 84 cellules.

Pouvez-vous cartographier cette information (par exemple sous forme de raster) ?

Chaque axe offre une carte correspondant à l'information qu'il porte...

4.6.4 Analyse spatiale

Ces cartes représentent de façon simple la variation spatiale de la synthèse multivariable des métriques paysagères réalisée par l'ACP.

Les **coordonnées** des cellules sur les axes sont des **combinaisons linéaires** des valeurs décrivant chaque cellule (les métriques calculées par **ClassStat**) : ce sont donc des variables aléatoires qui peuvent être analysées comme n'importe quelle autre VA...

On peut faire une analyse spatiale de ces coordonnées. Essayez de calculer le variogramme des coordonnées des cellules sur les axes 1, 2 et 3.

```
library(geoR)
```

```
## -----  
## Analysis of Geostatistical Data  
## For an Introduction to geoR go to http://www.leg.ufpr.br/geoR  
## geoR version 1.7-5.1 (built on 2015-04-15) is now loaded  
## -----
```

```
?variog
```

Dessinez les 3 variogrammes sur le même graphique.

Le variogramme du premier axe exprime une structure spatiale beaucoup plus marquée et correspond à un niveau de variation plus élevé. Pourquoi ?

4.6.5 Clustering

Chaque cellule est caractérisée par ses coordonnées sur les axes de l'ACP. Les premiers axes ont une inertie plus élevée que les suivants et on peut utiliser cette propriété pour faire des classifications focalisées sur les axes portant la plus grande quantité d'information. ce qui revient à délaissier les axes les moins informatifs (attention cependant...).

Faites un dendrogramme des 84 cellules avec l'information contenue sur les 3 premiers axes.

Utilisez `dist::stat`, `hclust::stat`