



OSU-OREME – Apéro technique #4 – 05/02/2015

INTÉGRATION DE SCRIPTS R DANS LES OUTILS DE TRAITEMENT QGIS

Cyril Bernard, CEFE-CNRS

Pourquoi R dans QGIS ?

QGIS, un soft plutôt
'pythonesque' ...

- **Python 2.7** inclus !
- **Numpy**, une bibliothèque pour le calcul matriciel
 - ▶ exemple : calculatrice raster
- **PyQGIS**, une bibliothèque pour interagir avec l'interface de QGIS
 - ▶ exemple : créer des plugins

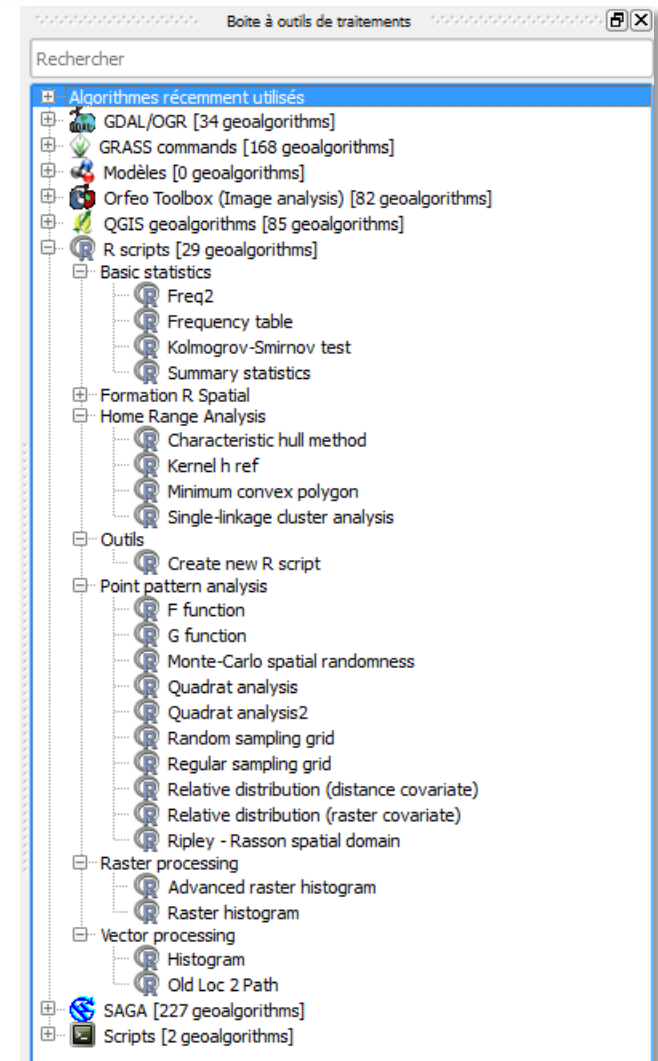
Mais l'intégration de R offre des
possibilités supplémentaires ...

- Pour les habitués de R !
- Intégration d'outils très spécialisés et basés sur des packages R comme **adehabitatHR** (domaines vitaux ou *home range*)



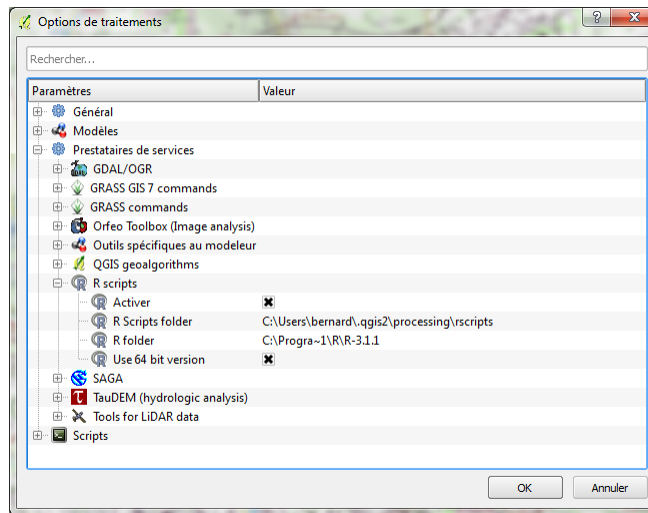
Intégration de scripts R dans QGIS

- Boîte à outils de traitements de QGIS 2.6
 - ▶ à l'origine un plugin (Sextante), plus tard intégré dans QGIS
- But : appeler des outils Grass, SAGA, R, OrfeoToolbox depuis QGIS
- Un certain nombre d'outils R sont déjà présents dans 'QGIS par défaut' ... comment les faire fonctionner ?



Installation et configuration

- **Installer R** si ce n'est pas déjà fait !
- Configurer le **chemin d'accès de R** et des **scripts .rsx** dans QGIS
 - ▶ Menu *Traitement / Options ...*

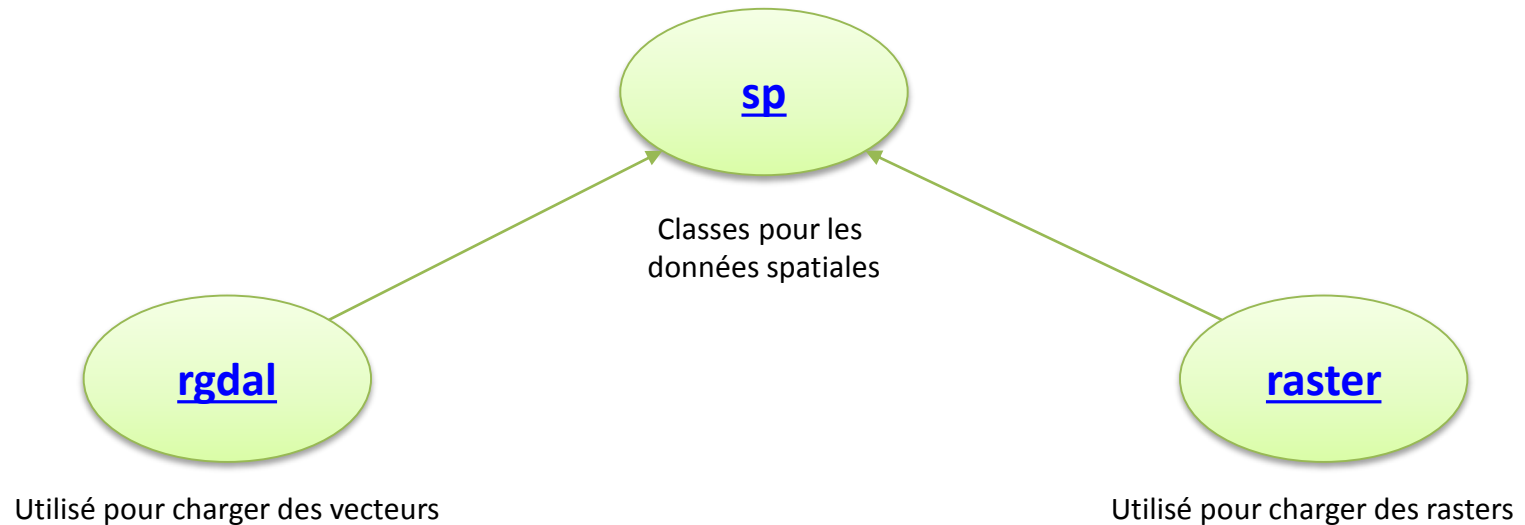


- **Installer les packages R nécessaires** au fonctionnement des scripts



Packages obligatoires

Ces 3 packages sont **obligatoires** pour la lecture / écriture des données SIG



Ils seront systématiquement chargés à chaque exécution d'un outil R dans QGIS



Quelques packages recommandés

adehabitatHR

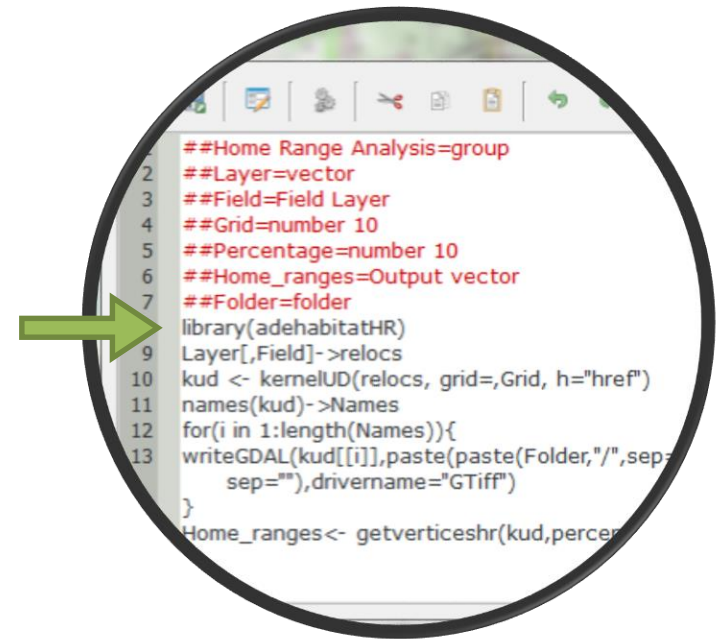
Outils Home Range analysis

spatstat

Outils Point pattern analysis

**rgeos,
rasterVis,
maptools ...**

Susceptibles d'être utilisé
par d'autres outils ...

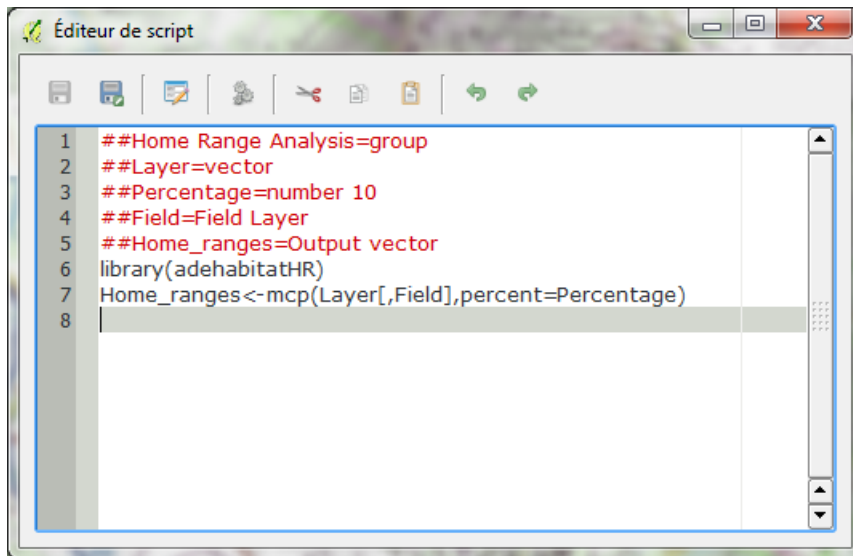


Clic-droit + Editer script pour voir
quel package il utilise



Quel différence entre un script .R classique et un script .rsx dans QGIS ?

- Le script .rsx intègre une liste des paramètres précédés par des ##
 - ▶ Couches vecteur ou raster en entrée
 - ▶ Choix d'un champ dans la table d'un vecteur
 - ▶ Paramètres numériques en entrée
 - ▶ Couches vecteur ou raster en sortie



```
1 ##Home Range Analysis=group
2 ##Layer=vector
3 ##Percentage=number 10
4 ##Field=Field Layer
5 ##Home_ranges=Output vector
6 library(adehabitatHR)
7 Home_ranges<-mcp(Layer[,Field],percent=Percentage)
8
```

Command	Function
##[datagis]=group	Sets the group to "datagis"
##layer = vector or raster	Specifies the input layer to use
##distance=number 100	Sets a number field with default 100
##title=string France	Get text input. Default is "France"
##field=field layer	Select a field from the vector layer "layer"
hist(layer[[field]])	R-command: Histogram for fields from layer
##showplots	Has to be set in order to see plot outputs
>t.test(layer[[field]])	Console output with a ">" before command
##output=output vector	File output as vector or raster

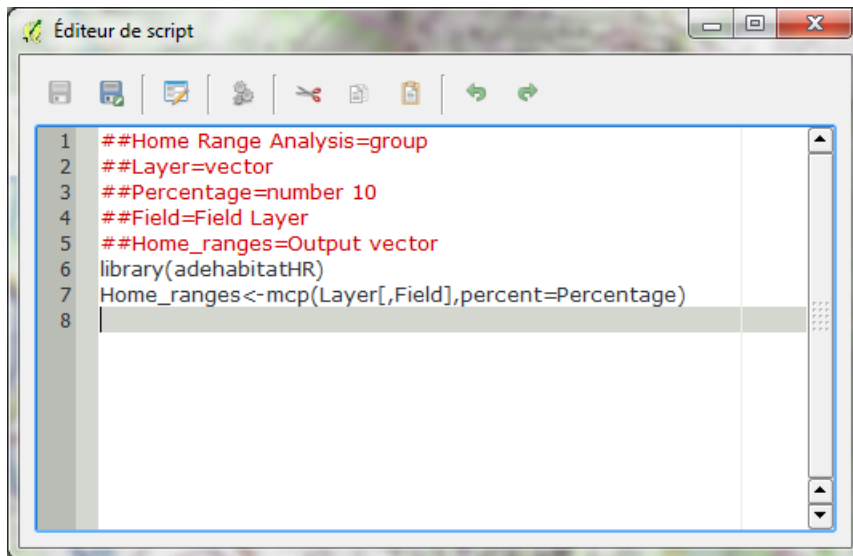
Source :

<https://conservationecology.wordpress.com/2013/08/14/port-your-r-scripts-to-qgis-using-sextante/>

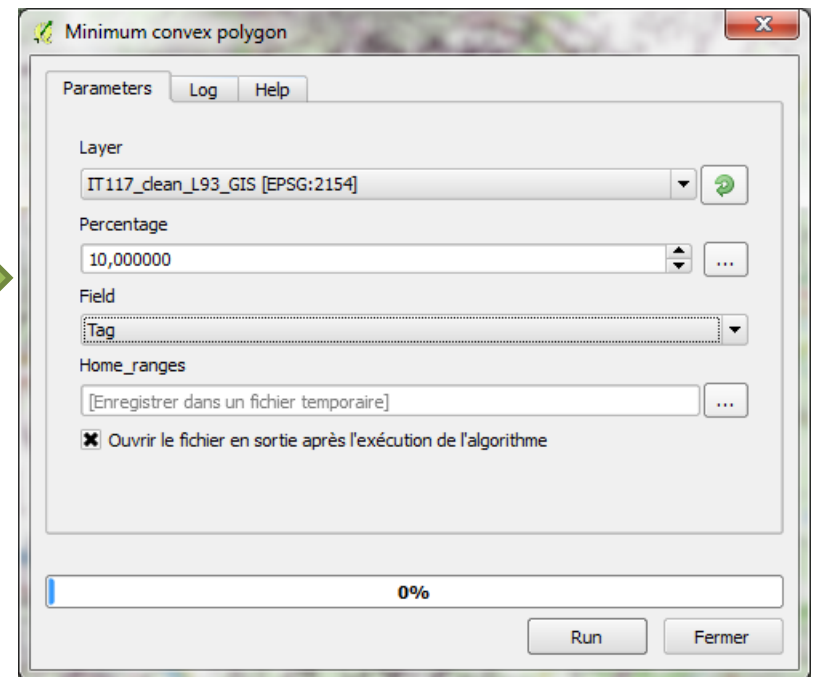


Interface des outils R dans QGIS

- L'interface est automatiquement générée d'après les paramètres du fichier .rsx !
 - ▶ Dans cet exemple de script, le calcul de polygone convexe à partir d'une couche de points ne représente qu'une seule ligne de code (appel de la fonction `mcp {adehabitatHR}`)

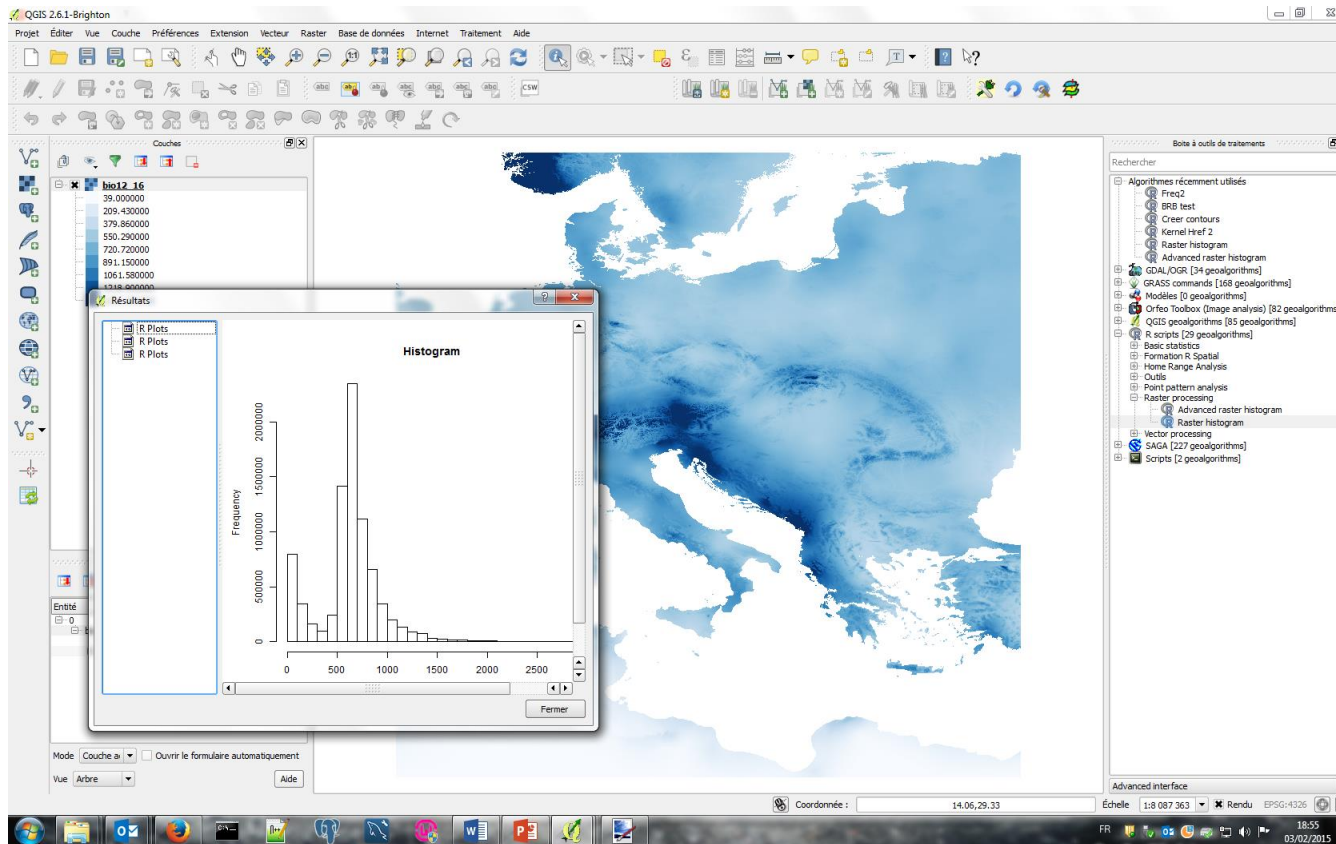


```
1 ##Home Range Analysis=group
2 ##Layer=vector
3 ##Percentage=number 10
4 ##Field=Field Layer
5 ##Home_ranges=Output vector
6 library(adehabitatHR)
7 Home_ranges<-mcp(Layer[,Field],percent=Percentage)
8
```



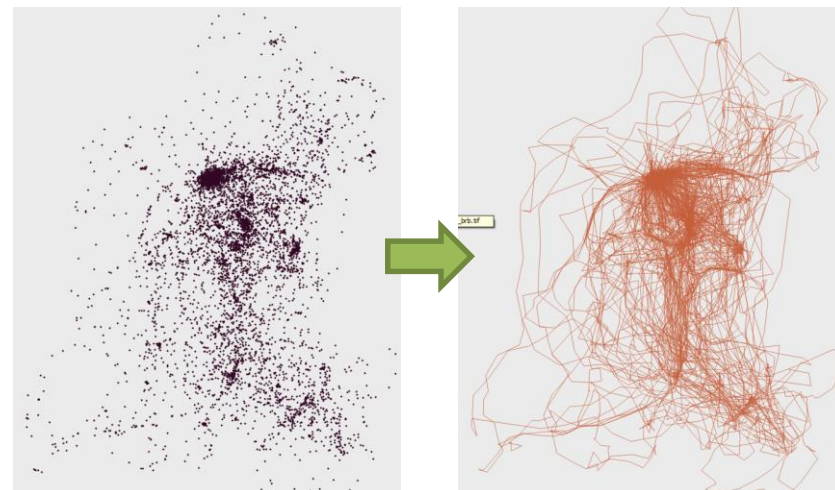
Outils R pour afficher des graphiques

- Possibilité d'appeler la fonction **plot** dans un script .rsx
 - ▶ Exemple : outil *Raster Histogram*
 - ▶ Génère un fichier PNG (inclus dans un HTML)



Outils R pour générer des vecteurs / raster

- Une bonne connaissance des packages [sp](#), [raster](#) et [rgdal](#) est requise pour écrire vos propres scripts
- Exemple : générer le trajet d'un animal (relier des points ordonnés)
 - ▶ La lecture et l'écriture de données font appel aux classes de données spatiales du package **sp**
 - ▶ Code concis mais complexe



```
1 ##Formation R Spatial=group
2 ##Layer_In=vector
3 ##Field=Field Layer_In
4 ##Layer_Out=output vector
5 # lire systeme coordonnees shape entree
6 sysCoord <- proj4string(Layer_In)
7 # trier shape
8 vSort <- Layer_In[order(Field)]
9 Layer_In <- Layer_In[order(vSort),]
10 # lire coordonnees points et creer une ligne avec
11 mat_xy <- coordinates(Layer_In)
12 oLine <- Line(mat_xy)
13 oLines <- Lines(list(oLine), ID=1)
14 oSpatLines <- SpatialLines(list(oLines), proj4string = CRS(sysCoord))
15 # creer table attributaire a 1 colonne
16 id_ligne <- as.integer(1)
17 dfAttr <- data.frame(id_ligne)
18 # enregistrer shape sortie
19 Layer_Out <- SpatialLinesDataFrame(oSpatLines, dfAttr)
20
```



Autres possibilités intéressantes

- Documenter les scripts
- Associer une légende par défaut (.qml) aux couches en sortie

Pour plus d'informations ...

- Tutoriel 'Intégration de scripts R dans les outils de traitement QGIS' :
https://www.cefe.cnrs.fr/images/stories/PLATEFORMES/SIE/docs/Outils_R_avec_QGIS.zip

