Une chaîne de traitement automatisée sous R :

Du chargement de données à la production de cartes et de modèles statistiques pour l'analyse d'une base de données hydromorphologiques nationale

Laboratoire LADYSS UMR 7533 - Université Paris Diderot Paris 7

Clélia Bilodeau, Maître de conférences

Laboratoire de Géographie Physique UMR 8591 CNRS - Paris 1

- **Frédéric Gob**, Maître de conférences
- **Nathalie Thommeret**, Post-doctorante

Vendredi 24 mai 2013 – SeminR

Partenaires

ONEMA

•**Véronique Nicolas**, Direction générale , Direction de l'Action Scientifique et Technique (DAST), co-responsable scientifique du projet,
•**Jean-Marc Baudoin**, Pôle ONEMA-IRSTEA, Aix en Provence, co-responsable scientifique du projet
•**Karl Kreuzenberger**, Direction générale, Direction de la Connaissance et de l'information sur l'Eau (DCIE)
•**Agents des Délégations Interrégionales (DIR) et des services départementaux (SD)**

IRSTEA

•**Marie-Bernadette Albert ,** IRSTEA, unité HBAN, Antony
•**Jérôme Belliard** ,IRSTEA, unité HBAN, Antony
•**Virginie Archaimbault**, IRSTEA, unité DYNAM, Lyon
•**Yves Souchon**, IRSTEA Lyon, Directeur du pôle ONEMA – IRSTEA, Lyon
•**Laurent Valette**, Ingénieur d'étude, pôle ONEMA – IRSTEA, Lyon
•**André Chandesris**, ingénieur IDAE, pôle ONEMA – IRSTEA, Lyon

La Directive Cadre sur l'Eau et ses enjeux

→ Directive Cadre européenne sur l'Eau (DCE) (2000)

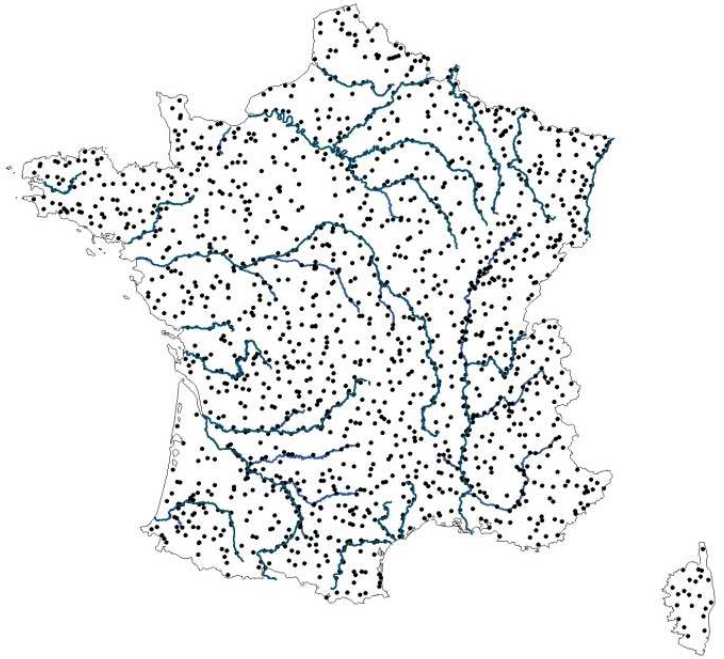
Objectif: « bon état» des cours d'eau d'ici 2015

→ état écologique

→ état chimique

→ **état hydromorphologique**

1500 stations RCS (Réseau de Contrôle et de Surveillance), 2006

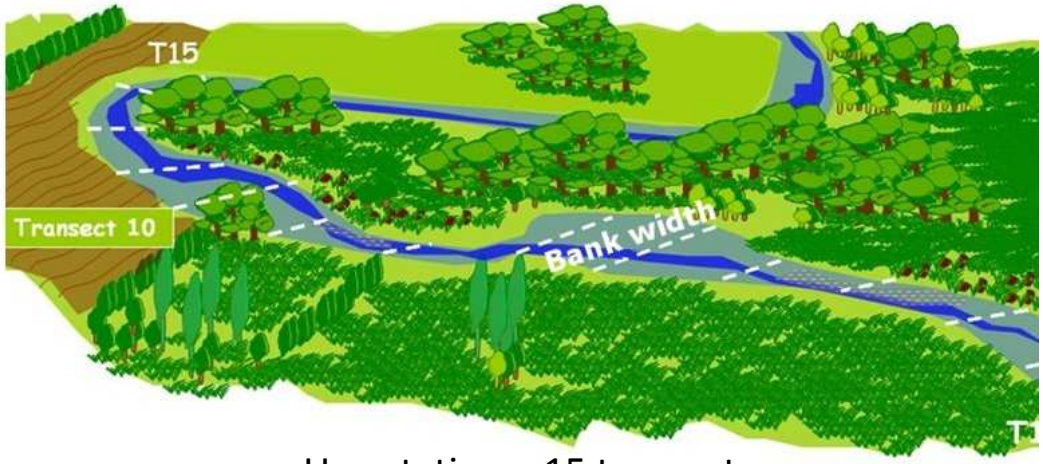


→ **Evaluation** de l'état actuel des cours d'eau

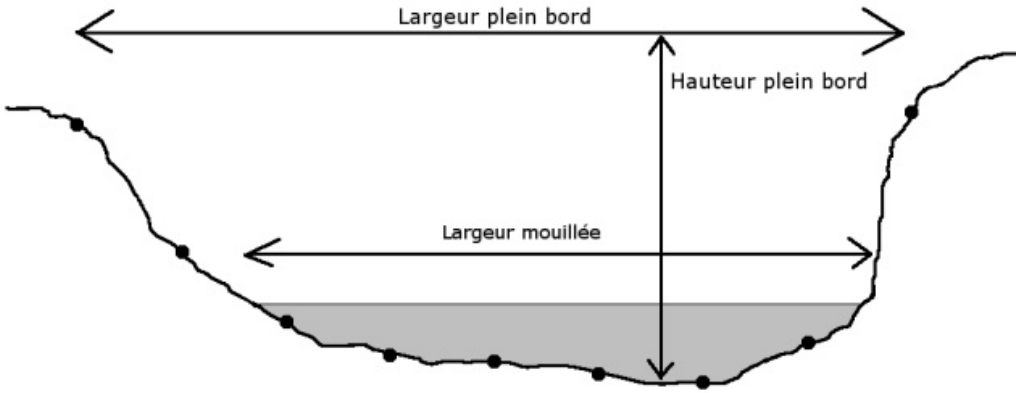
→ **Suivi** de leur évolution (tous les 6 ans)

Acquisition de données

→ Protocole standardisé CARHYCE (CARactérisation HYdromorphologique des Cours d'Eau)



Une station = 15 transects
Un transect = n>8 points



- Données acquises et saisies au niveau régional tout au long de l'année,
- Puis rassemblées dans une BDD nationale.

Objectifs, besoins, solutions

Objectifs

- Reconstituer la morphologie des stations
- Calculer les paramètres hydromorphologiques (variation de la profondeur, vitesse…)
- Visualiser la morphologie des stations
- Croiser des bases de données spatiales
- Cartographier les résultats
- Construire des modèles statistiques
- Répéter ces traitements

Utilisateurs

- Chercheurs
- Agents de l'ONEMA

Besoins

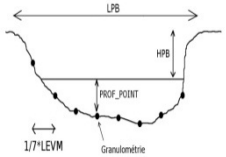
- **Calculs**
- **Graphiques 2D / 3D**
- **Fonctions SIG**
- **Statistiques**
- **Automatisation**
- **Logiciel gratuit**

Chaîne de traitements

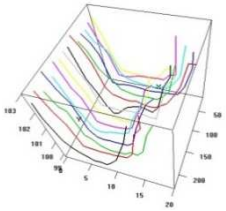
CARHYCE

Coordonnées géographiques des stations

Mesures hydromorphologiques

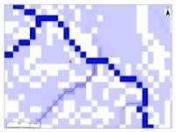


Reconstitution de la morphologie du lit



Croisements géographiques avec d'autres données:

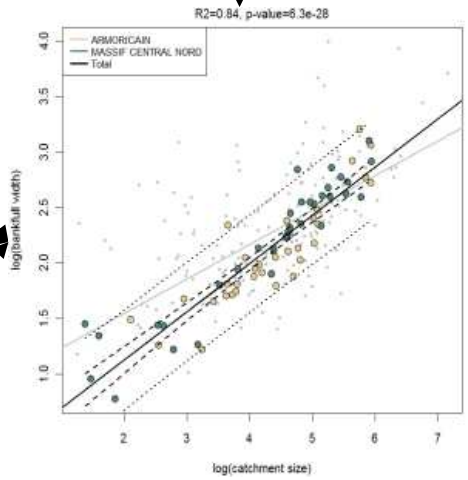
Calculs des paramètres hydromorphologiques



Taille du bassin versant (**BV**)

H.E.R.
Hydro-éco-régions

Base de données altération



Modèles de régression linéaire régionalisés caractérisant les cours d'eau non altérés

Etape 1: Importation des données

- Deux possibilités:
  - Depuis un fichier Excel (format *.csv)

```
#1) Recherche du dossier où se trouvent les données Carhyce
rep_carhyce<-choose.dir(caption = "Sélection du dossier où se trouvent les données Carhyce (fichiers *.csv)")
#2) A partir du répertoire spécifié, chargement des tables
# OPERATION
chemin_operation<-paste(c(rep_carhyce,"/OPERATION.csv"),collapse="")
Operation<-read.csv(chemin_operation,dec=".",sep = ";",header=TRUE)
#on indique que les décimales sont données par des , et que le séparateur est ;
```

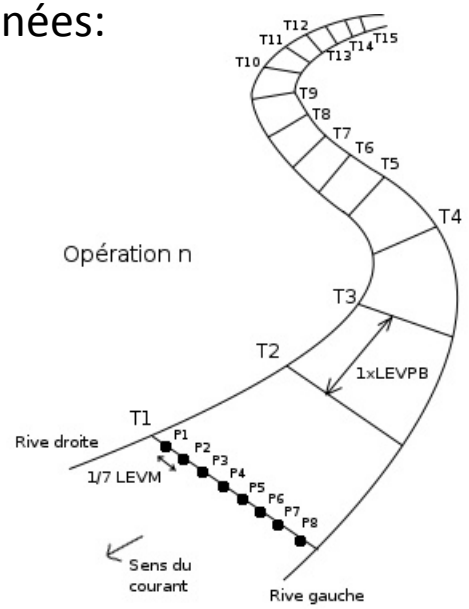
  - Depuis l'ensemble de la BDD MySQL

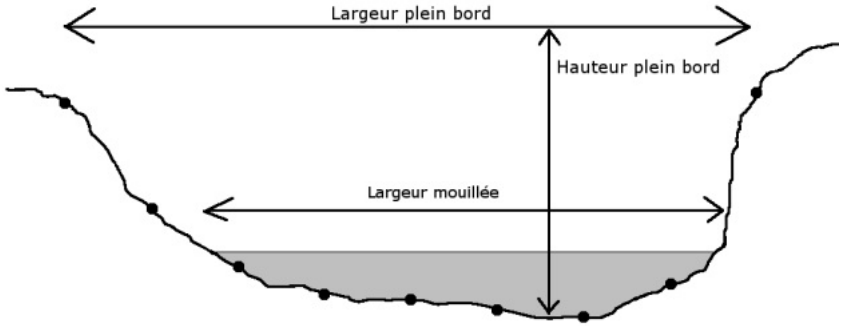
```
#1) Appel de la librairie du package RODBG (qui doit être installé)
# permettant le lien entre R et la base de donnée
# Attention: EasyPHP doit être ouvert
library(RODBC)

#2) Requete appel de la Table
sql_operation<-"SELECT `operations`.* FROM `operations`"

#3) Connection à la base et chargement de la table
connexion_carhyce_appli = odbcConnect(dsn = "carhyce_appli")
Operation <- sqlQuery(connexion_carhyce_appli, sql_operation)

#4) fermeture de la connexion_import dans variables achevés
odbcClose(connexion_carhyce_appli)
```

Etape 2: Structuration des données

Trois échelles de données:

- Les opérations
- Les transects
- Les points





Structure de **liste_operation** (data.frame):

Champ 1	Champ 2	Champ 3	…
Opération 1			
Opération 2			
…			

Structure de **liste_transect** (liste: fonction list()):

Champ 1	Champ 2	Champ 3	…
Transect 1			
Transect 2			
…			

Champ 1	Champ 2	Champ 3	…
Transect 1			
Transect 2			
…			

Structure de **liste_point** (liste de liste):

Champ 1	Champ 2	Champ 3	…
Point 1			
Point 2			
…			

Champ 1	Champ 2	Champ 3	…
Point 1			
Point 2			
…			

Accès à l'opération 1

```
liste_operation[1,]
```

Accès à l'opération 1, transect 2

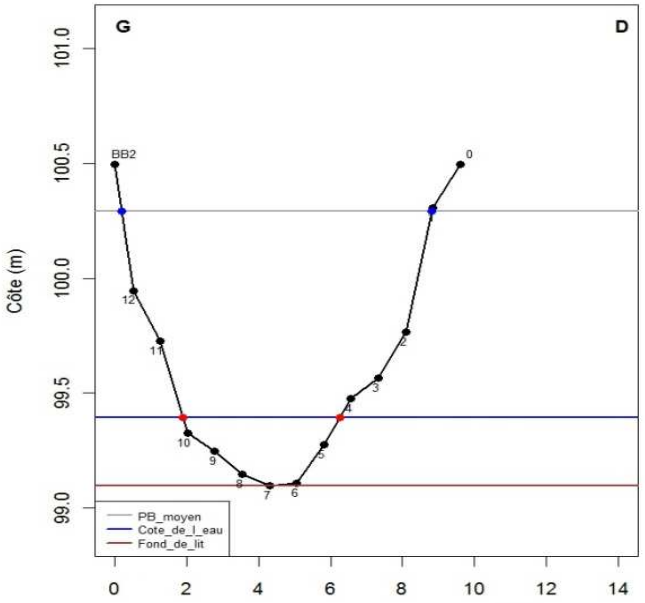
```
liste_transect[[1]][2,]
```

Accès à l'opération 1, transect 2, point 3

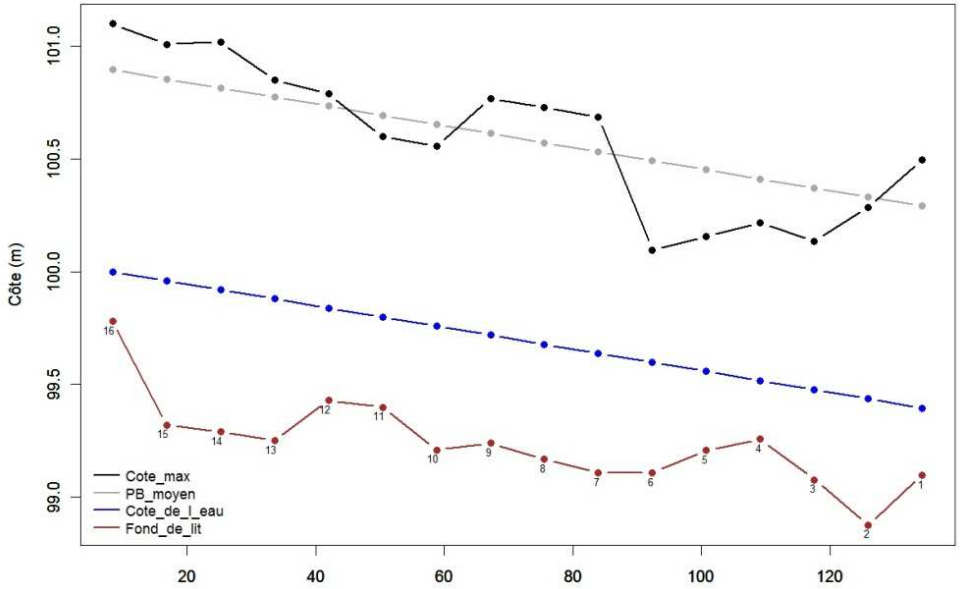
```
liste_point[[1]][[2]][3,]
```

Etape 3: Reconstitution de la morphologie du lit

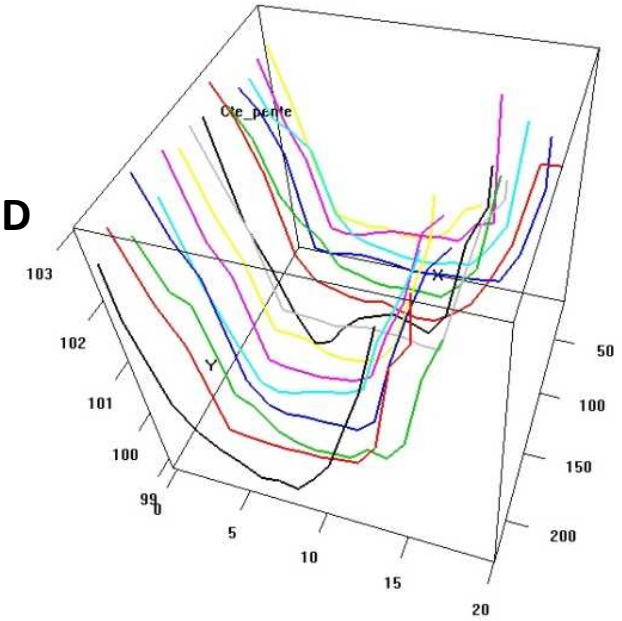
Profil en travers



Profil en long



Morphologie 3D de la station



Création de la fonction

```
Profil_3D<-function(i)
#i est le numéro de l'operation
{
require(rgl)
plot3d(...)
}
```

Appel de la fonction dans le programme

```
source("Profil_3D.R")

# Affichage de l'opération n°100
Profil_3D(100)
```

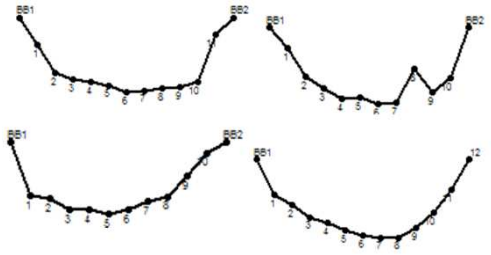
Etape 4: Calculs des paramètres hydromorphologiques

Paramètres mesurés

Paramètres calculés

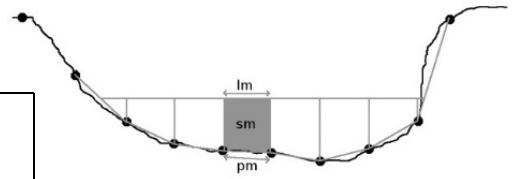
- Mesures moyennées

Largeur plein bord moyenne
Hauteur d'eau maximale et moyenne



- Calculs géométriques

Surface, largeur, et périmètre mouillés



- Calculs combinant d'autres paramètres

Rayon hydraulique Rh
Coefficient de rugosité K
Débit à pleins bords Qb
Nombre de Froude F
Puissance W et puissance spécifique wspe
Vitesse moyenne V
Force tractrice Tau0

Ex.: Rh=Sm/Pm

Etape 5: Importation de données spatiales et cartes

Définition des systèmes de coordonnées et projection

```
library(rgdal)
EPSG <- make_EPSG()
#Exemple 1: par recherche de chaine de caractère
# Projection Lambert 93: "L93"
Code_L93<-EPSG$code[grep("RGF93 / Lambert-93", EPSG$note)]
L93<-CRS(paste(c("+init=epsg:",Code_L93),collapse=""))
#Exemple 2: par recherche de code
# Projection Lambert 2 étendu "LII"
Code_LII<-27572
LII<-CRS(paste(c("+init=epsg:",Code_LII),collapse=""))
```

Import de données Raster

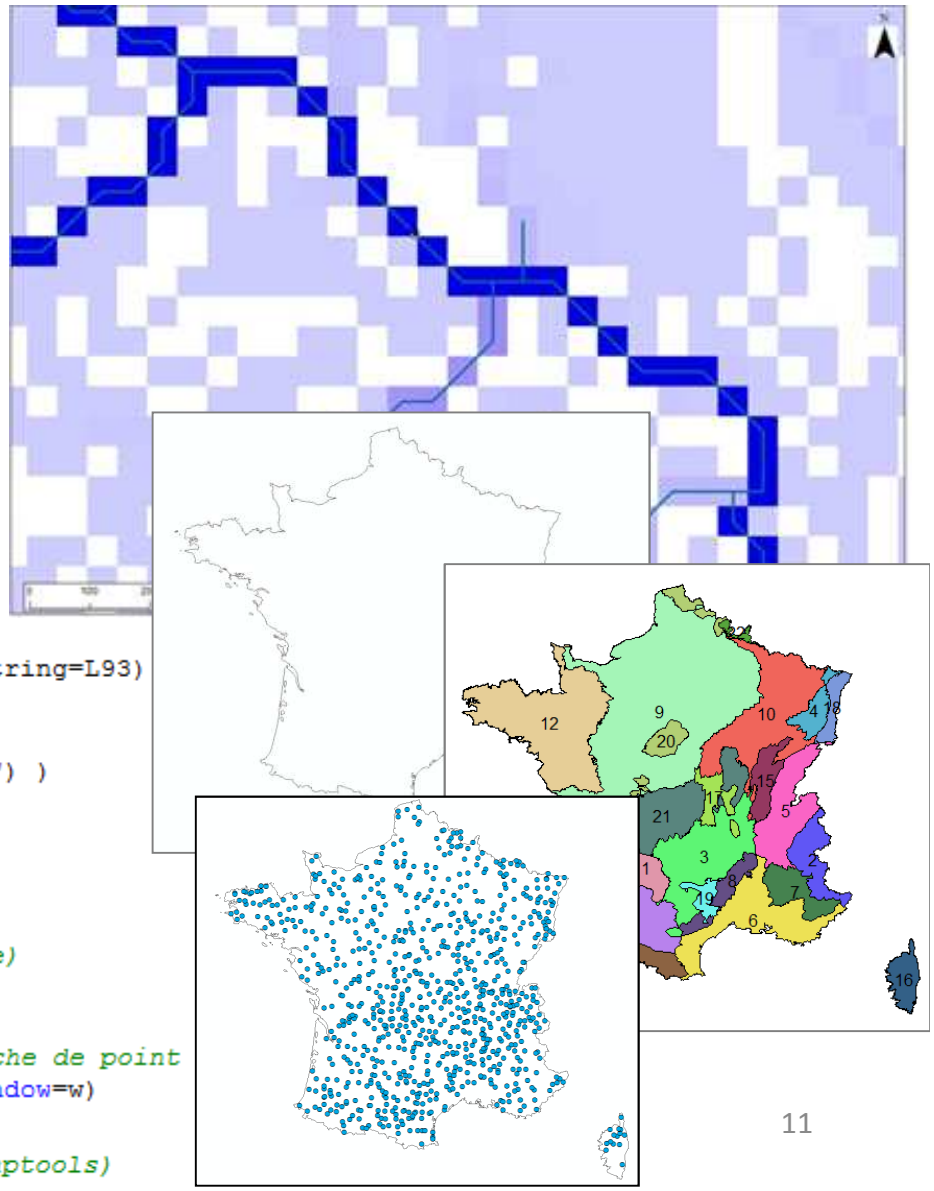
```
library(raster)
Surface_BV<- raster("C:/.../BV.tif")
projection(Surface_BV)<-LII
plot(Surface_BV)
```

Import de données Vecteur

```
library(maptools)
France<-readShapeLines("C:/.../Contour_France.shp",proj4string=L93)
# ou readShapePoints ou readShapePoly
# Convertion de projection (avec le package rgdal)
France_LII <- spTransform (France, CRS ("+init=epsg:27572") )
plot(France,add=TRUE)
```

Création de points à partir de coordonnées GPS

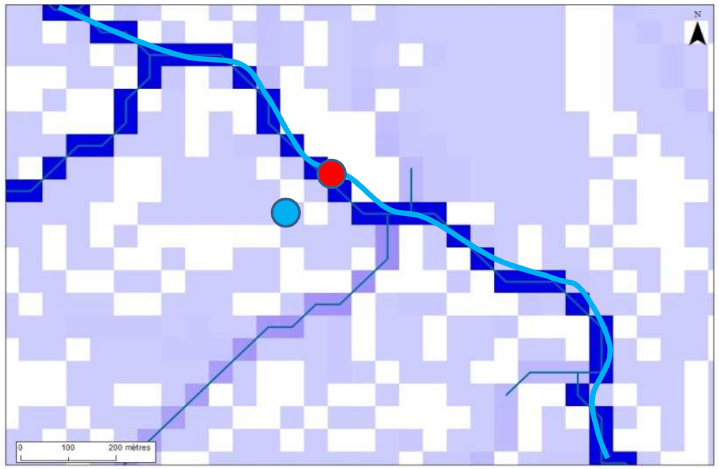
```
library(spatstat)
# Définition du Cadre cartographique (emprise de la France)
xlimite=c(0,1200000);ylimite=c(1600000,2800000)
w<-owin(xlimite,ylimite)
# import des coordonnées des points et création d'une couche de point
point<-ppp(liste_operation$X_GEO,liste_operation$Y_GEO,window=w)
plot(point,add= TRUE,pch=16)
# Autre possibilité fonction writeSpatialShape (package maptools)
```



Etape 6: Croisement avec d'autres BDD

Points Carhyce/Réseau : Fonction Snap

```
library(spatstat)
point<-ppp(liste_operation$X_GEO,liste_operation$Y_GEO,window=w)
reseau_hydro<-readShapeLines("C:/.../reseau_hydro.shp",proj4string=LII)
ligne<-as.psp(reseau_hydro,window=w)
snap<-project2segment(point,ligne)
X_Snap<-coords(snap$Xproj)[1]
Y_Snap<-coords(snap$Xproj)[2])
point_snap<-ppp(X_Snap,Y_Snap,window=w)
```

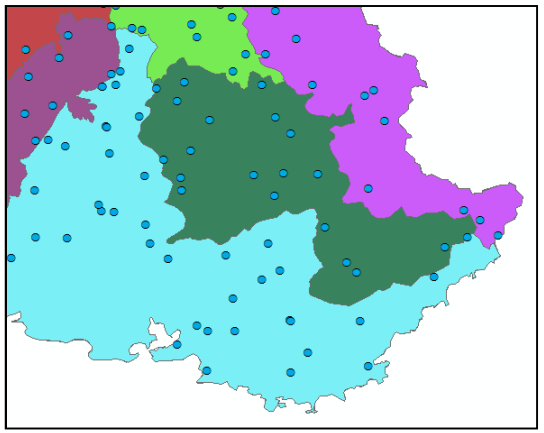


Points Crahyce/Raster : Fonction Extraction de données de raster à partir de points

```
Point_Surf_BV<-extract(Surface_BV,point_snap)
```

Points Carhyce/Polygones: Fonction Extraction de données de polygones à partir de points

```
HER<-readShapePoly("C:/.../HER.shp",proj4string=LII)
library(sp)
Nom_HER<-overlay(point_operation,HER)
```



Etape 7: Régression linéaire

Définition des paramètres

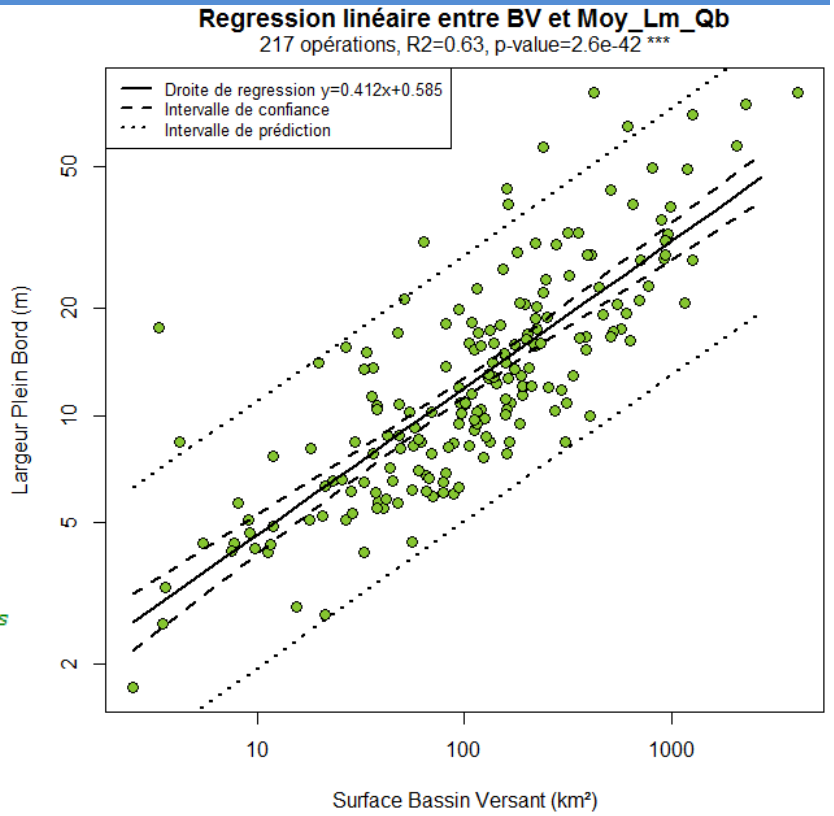
```
# Choix des paramètres
parametre_x<-"BV"
parametre_y <- "Moy_Lm_Qb"

# Transformation des valeurs en log
x<-log(liste_operation2[,parametre_x])
y<-log(liste_operation2[,parametre_y])
```

Calcul de régression linéaire

```
# Calcul de la régression
reg<-lm(y~x,na.action=na.exclude)
```

Calcul des paramètres associés à la régression

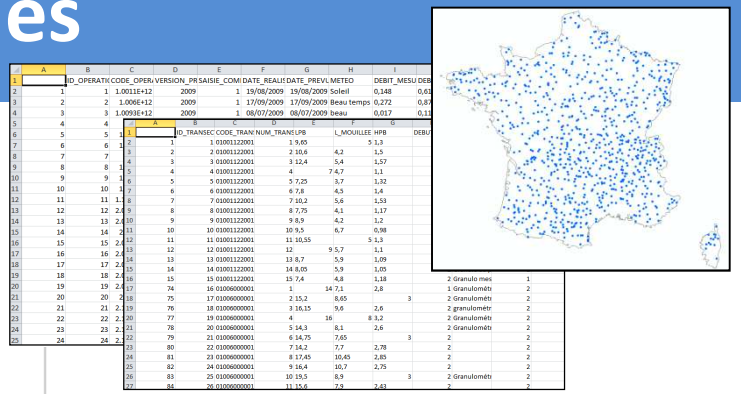
```
# Recupération de la p-value (-> significativité)
pvalue<-summary(reg)$coefficients[2,4]

# Calcul des intervalles de confiance et de prédiction et tracé des droites
xminmax<-c(min(x,na.rm=TRUE),max(x,na.rm=TRUE))
new <- data.frame(x = seq(xminmax[1], xminmax[2], 0.5))
pred.confid <- predict(reg, new, interval="confidence",level = 0.95)
pred.predict <- predict(reg, new, interval="prediction",level = 0.95)
```



Création du graphique

```
# Titre du graphique
titre<-paste("Regression linéaire entre ",parametre_x," et ",parametre_y,sep="")

# Plot
plot(x,y,cex=1.25,pch=21,xlab="Surface Bassin Versant (km²)",ylab="Largeur Plein Bord (m)",
xaxt = "l",yaxt = "l",main=titre,col="black",bg=vert,axes=F)

# Reconstruction des axes avec les valeurs initiales (non log)
axis(1,at=c(log(1),log(10),log(100),log(1000)),label=c(1,10,100,1000))
axis(2,at=c(log(2),log(5),log(10),log(20),log(50),log(100)),label=c(2,5,10,20,50,100))
box()

matplot(add = TRUE,new$x,cbind(pred.confid, pred.predict[,-1]),lwd=2,lty=c(1,2,2,3,3),col="black", type="l", ylab="predicted y")
```

Etape 8: Sorties



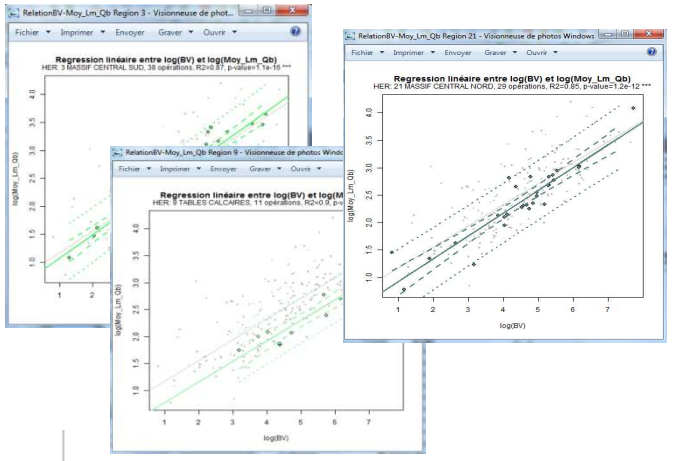
Fonction d'enregistrement de tableau de données

```
#Enregistrement en csv2 (utilise une virgule pour les décimaux)
write.csv2(liste_operation,file=chemin_operation_R)
```

Fonction d'enregistrement de couches vecteur

```
#Enregistrement du shapefile
library(maptools)
point_operation <- SpatialPointsDataFrame(liste_operation[,c("X","Y")],data.frame(liste_operation[,1:length(liste_operation)]))
writeSpatialShape(point_operation,"Stations_Carhyce")
```

Production et enregistrement de graphiques




```
#Définition des paramètres
parametre_x<-"BV"
parametre_y<-"Moy_Lm_Qb"
for (i in 1:22) {
Region<-i
# Création du fichier image
png(paste("Graphique_Region",i,".png", sep=""),width=500,height=500)
#Plot
plot(x,y)
dev.off()

#Definition du répertoire
rep_graphique<-paste(c("C:/.../Graph_",parametre_x,"_",parametre_y,"_Resultat/"),collapse="")
#Definition du ou des fichiers
fichier<- paste(c(rep_graphique,"HER_",as.character(Region),"_graph_",parametre_x,"_",parametre_y,".jpg"),collapse="")

#En jpeg
jpeg(filename = fichier,,  width = 1024, height = 900, units = "px", pointsize = 20, quality = 100, bg = "white")
plot()
dev.off()
#En pdf
pdf("Resultat.pdf")
plot()
dev.off()

}
```

A suivre…

- **Points forts:**
  - R permet de **réaliser l'ensemble de la chaîne de traitements** spatiaux et statistiques
  - **Temps de calcul** : env. 3h
    10^3 opérations
    10^4 transects
    10^6 points

- **Points faibles:**
  - Difficultés d'importer et de traiter des **données spatiales de tailles importantes** telles que les tronçons du réseau hydrographique français (10^6 tronçons, 80/100 Mo)

- **Perspectives:**
  - Intégration d'une **interface** utilisateur
  - Vers la construction d'un outil permettant l'estimation de l'altération hydromorphologique d'un cours d'eau:
    Outil d'aide à la réflexion pour la restauration hydromorphologique des rivières
    Pour cela utilisation d'analyse discriminante linéaire pour un modèle combinant plusieurs variables
    → Package MASS, Fonction lda()