Fiche TD avec le logiciel  : tdr26

Courbes de niveau

D. Chessel

Comment représenter la variation d'une mesure spatialisée ? On utilise pour les illustrations une carte météorologique.

Table des matières

1 Introduction

La question est simple.

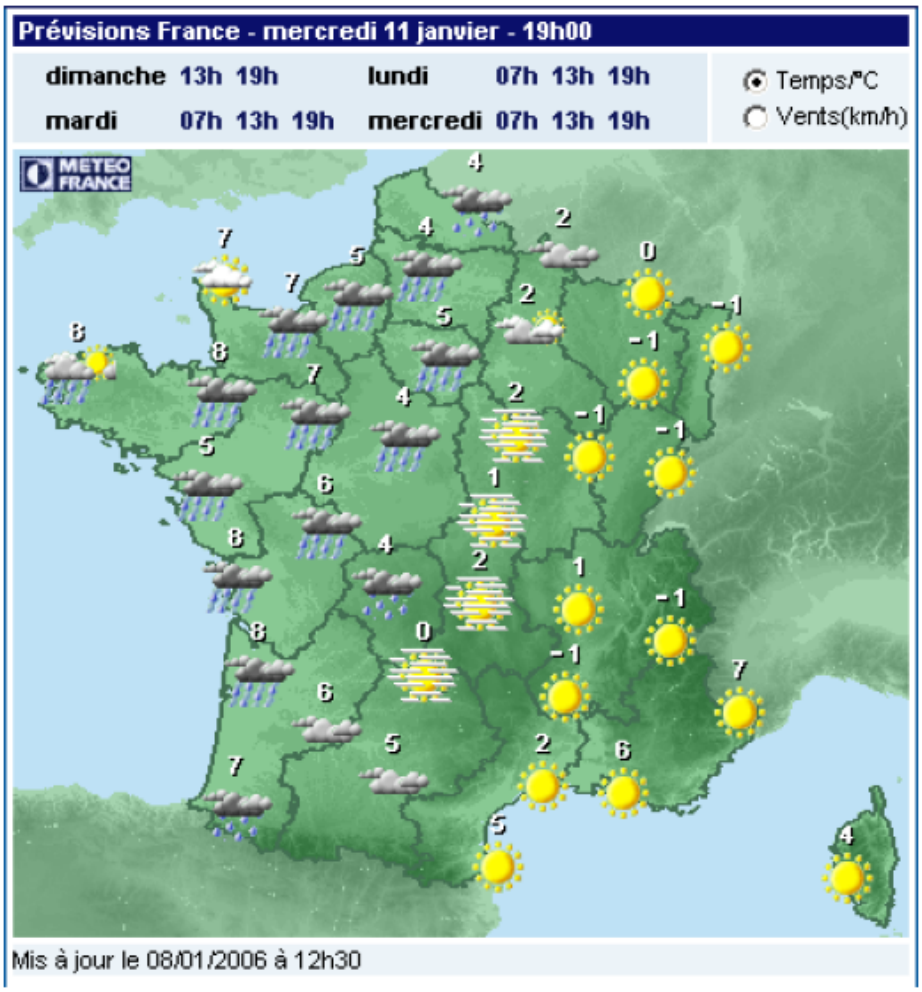


FIG. 1 – Carte météorologique éditée trois fois par jour sur le site de Meteo France.

`www.meteofrance.com/FR/index.jsp`

Quand on voit une carte qui porte des valeurs, on pense immédiatement qu'elle représente une fonction. En tout point d'une carte, on peut mesurer la température. En quelques points on peut l'éditer. Comment choisir et exprimer un modèle sous-jacent.

2 Implanter le matériel

Télécharger et installer un logiciel de capture d'image :

`http://www.gratuiciel.com/freeware/capture-ecran.htm`

pour sauvegarder dans le dossier de travail une copie de la figure 1. On peut utiliser un format courant (tiff, jpeg ou png).

On peut aussi sauvegarder directement après capture de l'image dans le fichier pdf (utiliser acrobat 7).

Télécharger et installer un logiciel de conversion de format de fichier graphique :

`http://www.imagemagick.org/script/index.php`

Ces outils peuvent être disponibles directement dans votre système d'exploitation. Transformer la figure au format `pnm` dans un fichier `tdr26.pnm`. Afficher cette carte dans R. La librairie `pixmap` est indispensable :

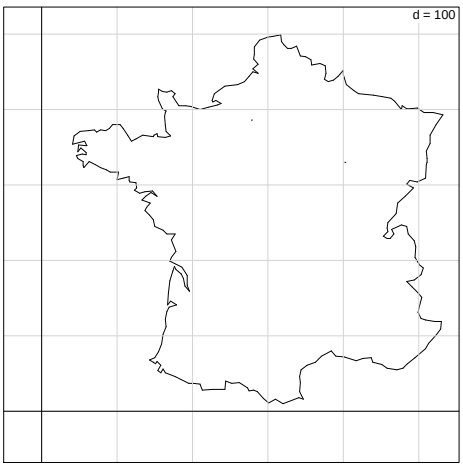
```
library(pixmap)
tbg <- read.pnm("tdr26.pnm")
plot(tbg)
```

La suite de la fiche suppose que vous voyez à l'écran dans R la carte étudiée. Si c'est le cas, pour tracer des courbes de niveaux, il faut réunir les éléments constitutifs.

2.1 Fond de dessin

La figure proposée est lourde et encombrée de nombreux logos. Elle ne peut pas servir de fond. On peut utiliser :

```
library(ade4)
data(elec88)
par(mar = rep(0, 4))
wa <- elec88$contour
s.label(elec88$xy, contour = wa, clab = 0, cpoi = 0)
```



Le caractère insulaire de la Corse implique une discontinuité qui n'est pas cohérente avec le reste du problème de spatialisation. Son absence est justifiée.

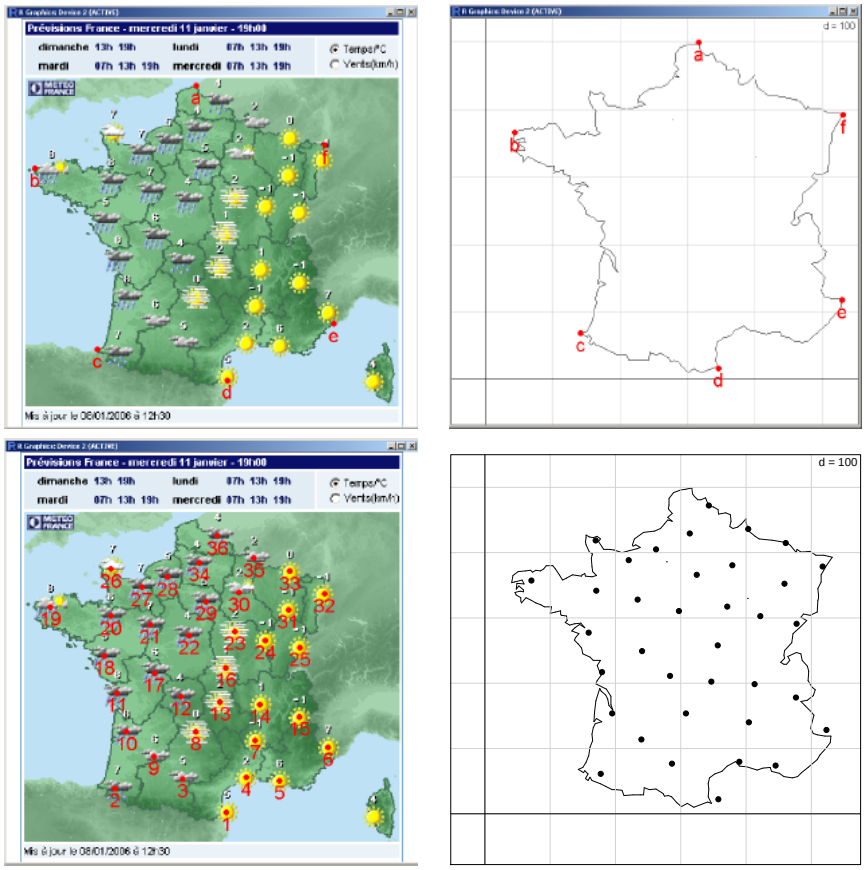


FIG. 2 – Report d'échelle

2.2 Digitalisation des coordonnées

Il faut maintenant importer dans ce système de coordonnées tous les éléments de la carte d'entrée. Implanter un petit utilitaire pour récupérer des coordonnées à l'écran :

```
"enreg" <- function(n = 3, labels = as.character(1:n)) {
    k <- 1
    res <- matrix(0, n, 2)
    while (k <= n) {
        a <- as.numeric(locator(1))
        points(a[1], a[2], col = "red", pch = 20, cex = 2)
        text(a[1], a[2], labels[k], col = "red", cex = 2, pos = 1)
        res[k, ] <- a
        k <- k + 1
    }
    res <- as.data.frame(res)
    names(res) = c("x", "y")
    row.names(res) <- labels
    return(res)
}
```

Essayer cette fonction. On pourra ensuite l'utiliser trois fois. Digitaliser d'abord sur la carte à étudier les six points de l'hexagone (figure 3 en haut, à gauche) :

```
plot(tbg)
hexaold <- enreg(6, letters[1:6])
```

Repérer les mêmes points sur le fond utile (figure 3 en haut, à droite) :

```
par(mar = rep(0, 4))
s.label(elec88$xy, contour = wa, clab = 0, cpoi = 0)
hexanew <- enreg(6, letters[1:6])
```

Digitaliser ensuite les points de mesure (figure 3 en bas, à gauche). Il y en a 36 :

```
plot(tbg)
xyold <- enreg(36)
```

Utiliser enfin un utilitaire d'ade4 pour faire le transfert d'échelle (c'est une règle de trois !) :

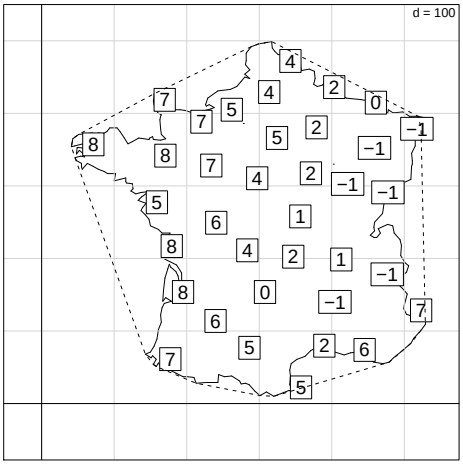
```
xynew <- scatterutil.scaling(hexaold, hexanew, xyold)
names(xynew) <- names(xyold)
par(mar = rep(0, 4))
s.label(xynew, contour = wa, cpoi = 1.5, clab = 0)
```

Il s'agit évidemment d'un exercice pour disposer d'un ensemble de mesures spatialisées qui a un sens immédiat.

2.3 Valeurs observées

Enregistrer, à l'aide la fonction `scan`, dans un vecteur `obs`, les 36 valeurs observées. Bien garder l'ordre utilisé dans la digitalisation. Vérifier en éditant les valeurs observées en clair :

```
par(mar = rep(0, 4))
s.label(xynew, contour = wa, , clab = 1.5, label = as.character(obs))
lines(wa[chull(wa[, 1:2]), 1:2], lty = 2)
```

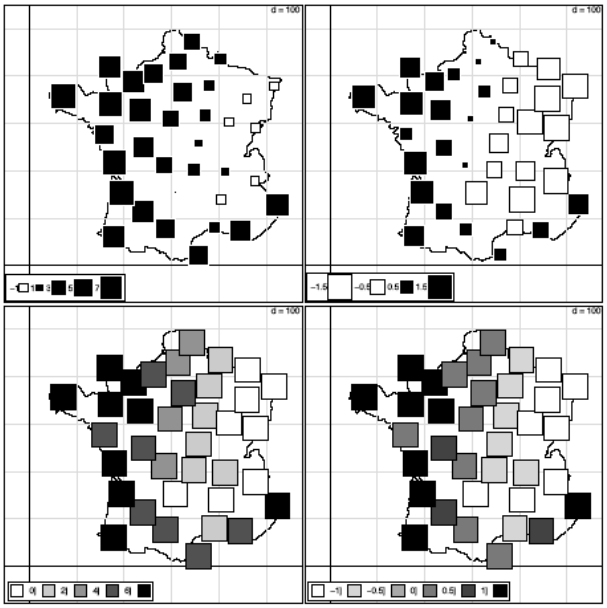


On dispose du matériel nécessaire à l'étude d'une variable spatialisée.

3 Représentation des données

3.1 Cartographie par valeurs

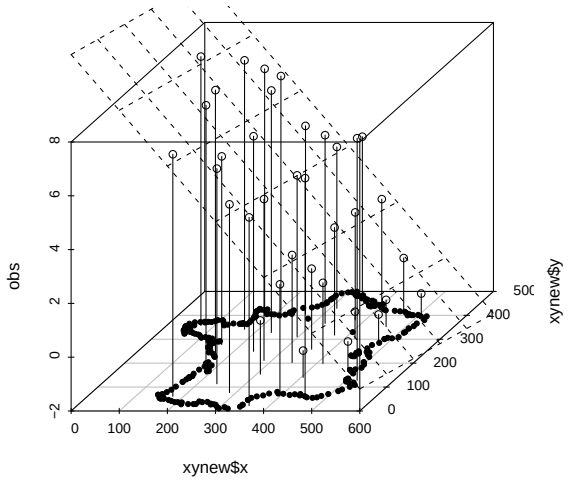
On commencera par cartographier la valeur observée.



Commentez la figure. Il n'est jamais simple de choisir une representation graphique.

3.2 Données à trois dimensions

Utiliser la librairie `scatterplot3d` [3]. La fonction renvoie comme valeurs ... des fonctions ! Lesquelles ?




```
library(scatterplot3d)
w3d <- scatterplot3d(x = xynew$x, y = xynew$y, z = obs, type = "h")
w3d$points3d(x = wa[, 1], y = wa[, 2], z = rep(-2, nrow(wa)), type = "p",
```

```
    pch = 20)
w3d$plane3d(lm(obs ~ xynew[, 1] + xynew[, 2]))
```

4 Représentation des modèles

En fait, la variable est régionalisée. Elle prend une valeur en tout point mais n'est mesurée qu'en quelques échantillons. Ce qu'on cherche implicitement, c'est à représenter, non pas l'échantillon mais la structure spatiale (mode de variation dans l'espace) qu'il estime.

4.1 Grille d'estimation

La librairie `splancs` [4, 1] permet de comprendre le fonctionnement des courbes de niveaux.

```
library(splancs)
```

```
Spatial Point Pattern Analysis Code in S-Plus
 Version 2 - Spatial and Space-Time analysis
```

Commencer par mettre en place un polygone de contour qui limitera l'espace dans lequel est construit le modèle (figure 3 en haut à gauche).

```
par(mar = rep(0, 4))
s.label(xynew, contour = wa)
wpoly <- enreg(18)
```

Implanter une grille de points (figure 3 en haut à droite) :

```
wg <- expand.grid(25 * (0:22), 25 * (0:22))
names(wg) <- c("x", "y")
par(mar = rep(0, 4))
s.label(xynew, contour = wa)
polygon(wpoly)
points(wg)
```

Que fait la fonction `expand.grid` ? Limiter la grille à la partie utile (figure 3 en bas à gauche) :

```
wgutil <- wg[inout(wg, wpoly), ]
par(mar = rep(0, 4))
s.label(xynew, contour = wa, clab = 0, cpoi = 0)
polygon(wpoly)
points(wgutil, pch = 20)
names(wgutil) <- c("x", "y")
```

Que fait la fonction `inout` ? Vérifier le résultat (figure 3 en bas à droite)

```
inoutmat <- matrix(inout(wg, wpoly), 23)
plot(wgutil, cex = 2)
points(xynew, pch = 20, cex = 2)
```

Ceci permet de manipuler une matrice de valeurs sur une grille et la partie utile de celle-ci.

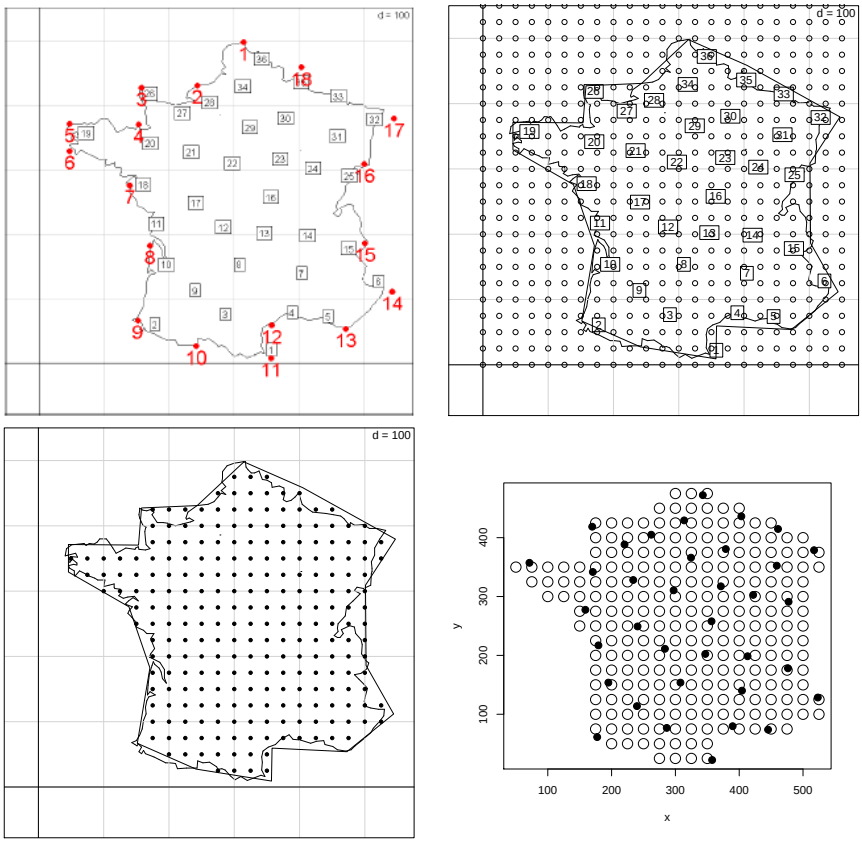


FIG. 3 – Implantation d'une grille d'estimation d'une variable régionalisée.

4.2 Estimation par régression polynomiale

Estimer les valeurs sur la grille par une régression polynomiale sur les valeurs observées (figure 4 en haut à gauche) :

```
lpoly <- lm(obs ~ x + y + I(x^2) + I(y^2) + x * y, data = xynew)
modpolyvec <- predict(lpoly, newdata = wgutil)
s.label(wgutil, clab = 0.7, lab = as.character(round(modpolyvec,
    1)), contour = wa)
```

Représenter ces valeurs en niveaux de couleur (figure 4 en haut à droite) :

```
modpolymat <- matrix(predict(lpoly, newdata = wg), 23)
modpolymat[!inout(wg, wpoly)] <- NA
xmat <- 25 * (0:22)
ymat <- 25 * (0:22)
s.label(xynew, contour = wa)
image(xmat, ymat, modpolymat, add = T)
```

4.3 Estimation par régression locale

Estimer les valeurs sur la grille par une régression locale sur les valeurs observées (figure 4 en bas à gauche) :

```
lo <- loess(obs ~ x + y, data = xynew, span = 0.5)
modloessmat <- predict(lo, newdata = wg)
modloessmat[!inout(wg, wpoly)] <- NA
modloessvec <- as.numeric(modloessmat)
wgloess <- wg[!is.na(modloessvec), ]
modloessvec <- modloessvec[!is.na(modloessvec)]
s.label(wgloess, clab = 0.7, lab = as.character(round(modloessvec,
    1)), contour = wa)
```

Représenter ces valeurs en niveaux de couleur (figure 4 en bas à droite) :

```
s.label(xynew, contour = wa)
modloessmat <- matrix(modloessmat, 23)
s.label(xynew, contour = wa)
image(xmat, ymat, modloessmat, add = T)
```

Le premier modèle est évidemment très mauvais. Le second est un peu plus compliqué. Lire avec attention la fiche de `loess`. On suppose ici que le lecteur a déjà pratiqué la régression locale à une dimension (voir tdr45). La prédiction du modèle se fait automatiquement sur toute la grille. On ne conserve que les points à l'intérieur du domaine et ceux qui ont permis l'estimation. Ceci permet de montrer le fonctionnement de la pratique des courbes de niveaux : les difficultés techniques disparaîtrons ensuite dans les fonctions. Imprimer les deux dernières figures et tracer à la main des courbes de niveaux.

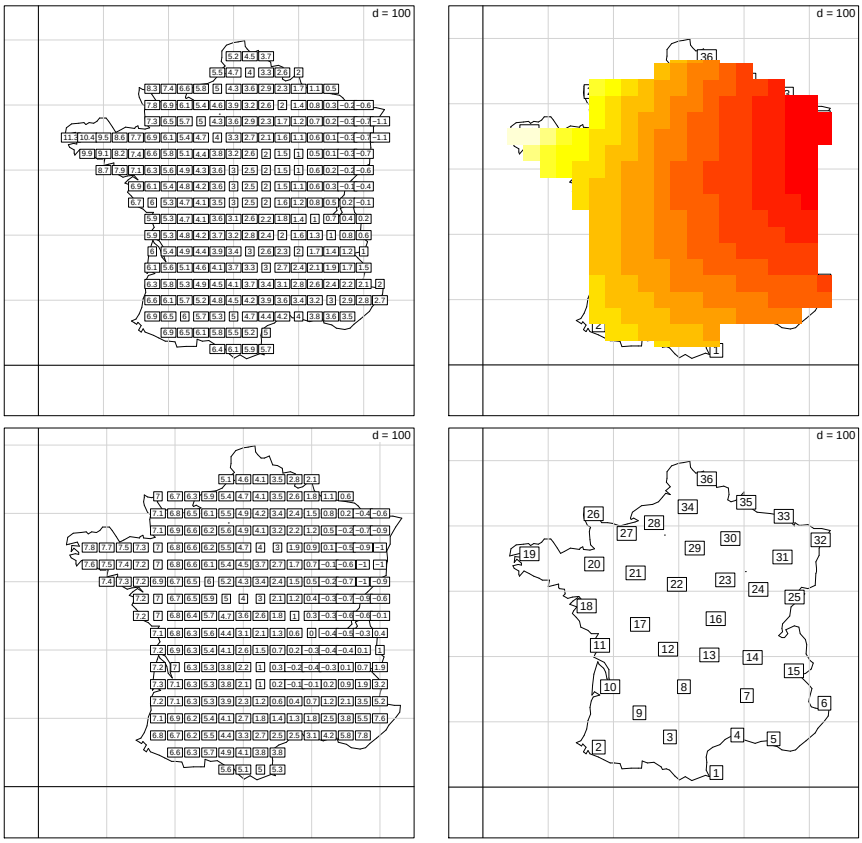


FIG. 4 – Estimation d'une variable régionalisée sur une grille de points par régression polynomiale, en haut, par régression locale, en bas.

4.4 Courbes de niveau

C'est le moyen le plus commun de restituer ces modèles. Observer d'abord la définition (figure 5 en haut à gauche) :

```
par(mar = rep(0, 4))
s.label(xynew, clab = 0, cpoi = 0, xlim = c(350, 500), ylim = c(150,
    300), grid = F)
abline(h = ymat)
abline(v = xmat)
s.label(wgutil, lab = as.character(round(modpolyvec, 1)), add.plot = T)
contour(xmat, ymat, modpolymat, lev = c(0, 0.5, 1, 1.5, 2, 2.5),
    add = T, lwd = 2, labcex = 2)
```

Superposer image et courbes (figure 5 en haut à droite) :

```
s.label(xynew, clab = 0, cpoi = 0, contour = wa)
image(xmat, ymat, modpolymat, add = T)
contour(xmat, ymat, modpolymat, add = T, lwd = 2, labcex = 2)
```

Superposer modèle et données (en bas) :

```
s.label(xynew, clab = 0, cpoi = 0, contour = wa)
contour(xmat, ymat, modpolymat, add = T, lwd = 2, labcex = 2)
s.label(xynew, clab = 1, label = as.character(obs), add.p = T)
```

```
s.label(xynew, clab = 0, cpoi = 0, contour = wa)
contour(xmat, ymat, modloessmat, add = T, lwd = 2, labcex = 2)
s.label(xynew, clab = 1, label = as.character(obs), add.p = T)
```

L'exercice permet de bien séparer le procédé graphique de restitution d'une mesure sur une grille de points (image et/ou courbes de niveaux) et le modèle qui passe des observations à la grille de mesure. On pourra continuer cet exercice avec les données `t3012` de la librairie `ade4` ou la question posée dans la fiche `pps013` sur les données de J.M. Chacornac [2].

Références

[1] R. Bivand and A. Gebhardt. Implementing functions for spatial statistical analysis using r language. *Journal of Geographical Systems*, 2 :307–317, 2000.

[2] J.M. Chacornac. *Lacs d'altitude : Métabolisme oligotrophique et approche typologique des écosystèmes*. PhD thesis, 1986.

[3] U. Ligges and M. Maechler. Scatterplot3d - an r package for visualizing multivariate data. *Journal of Statistical Software*, 8(11) :1–20, 2003.

[4] B. Rowlingson and P.J. Diggle. Splancs : spatial point pattern analysis code in s-plus. *Computers and Geosciences*, 19 :627–655, 1993.

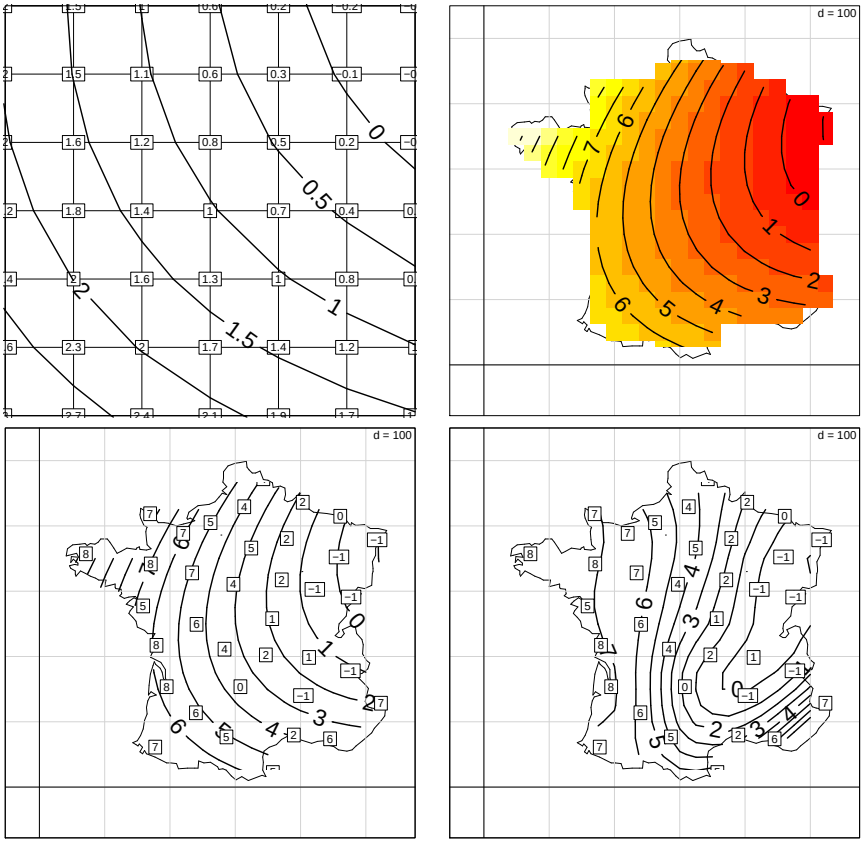


FIG. 5 – Restitution des modèles par courbes de niveaux.