Compte rendu de Projet du statistique spaciale

29/12/2020

Géostatistique

Dans cette partie on dispose de deux fichiers, le premier fichier parana.txt contient un tableau formé des colonnes suivantes :
— colonne 1 : abscisses des stations (Est).
— colonne 2 : ordonnées des stations (Nord).
— colonne 3 : mesures aux stations (Data).
Et chaque ligne correspond à une mesure des précipitations effectuées en chaque stations durant la période d'étude.
Le deuxième fichier parana.borders.txt contient deux colonnes qui représentent les coordonnées géographiques (Est et Nord) des frontières de l'état du Parana, on dispose de 369 données.

Chargement de fichiers de données

```
parana= read.csv("parana.txt", sep="")
parana.borders = read.csv("parana-borders.txt", sep="")
```

Analyse exploratoire

Analyse decriptive

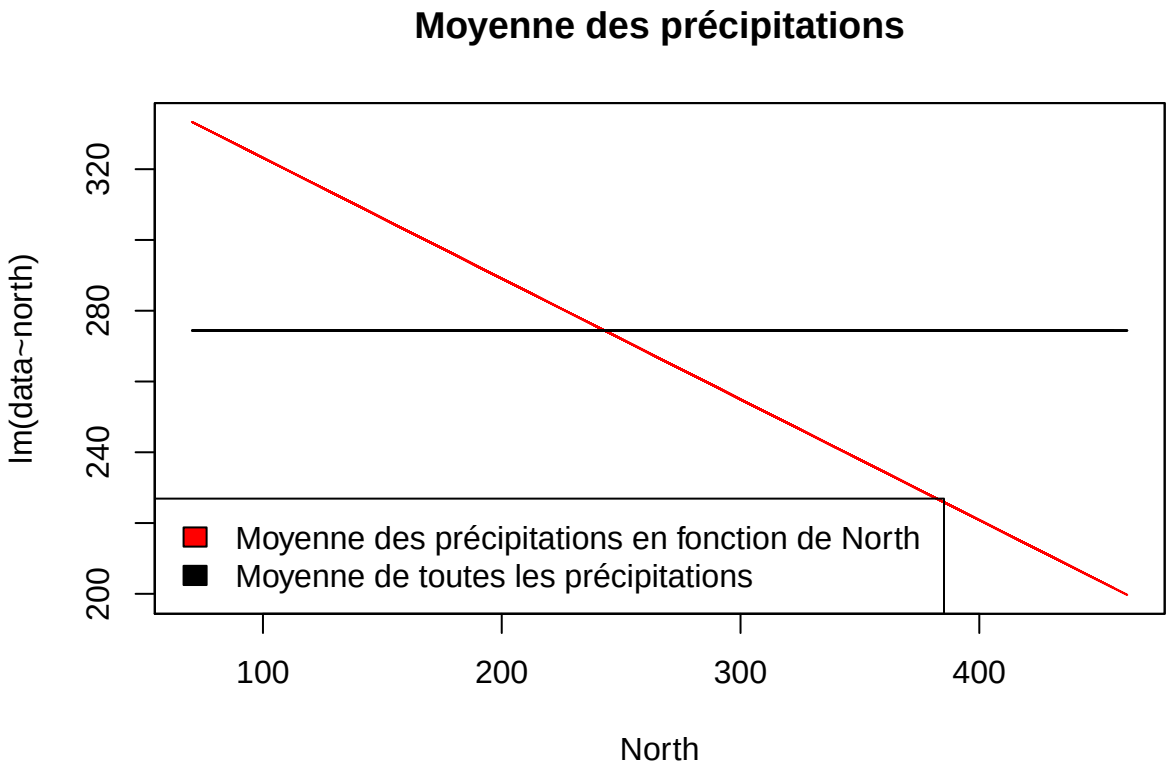
```
summary(parana)
```

```
##       east           north             data
##  Min.   :150.1   Min.   : 70.36   Min.   :162.8
##  1st Qu.:263.9   1st Qu.:163.33   1st Qu.:234.2
##  Median :366.9   Median :223.70   Median :269.9
##  Mean   :399.2   Mean   :243.01   Mean   :274.4
##  3rd Qu.:512.7   3rd Qu.:319.76   3rd Qu.:318.2
##  Max.   :768.5   Max.   :461.97   Max.   :413.7
```

On remarque que les données de précipitations mésurées sont comprises entre 162.8 et 413.7 avec une moyenne de 274.4.

Analyse de la moyenne des précipitations

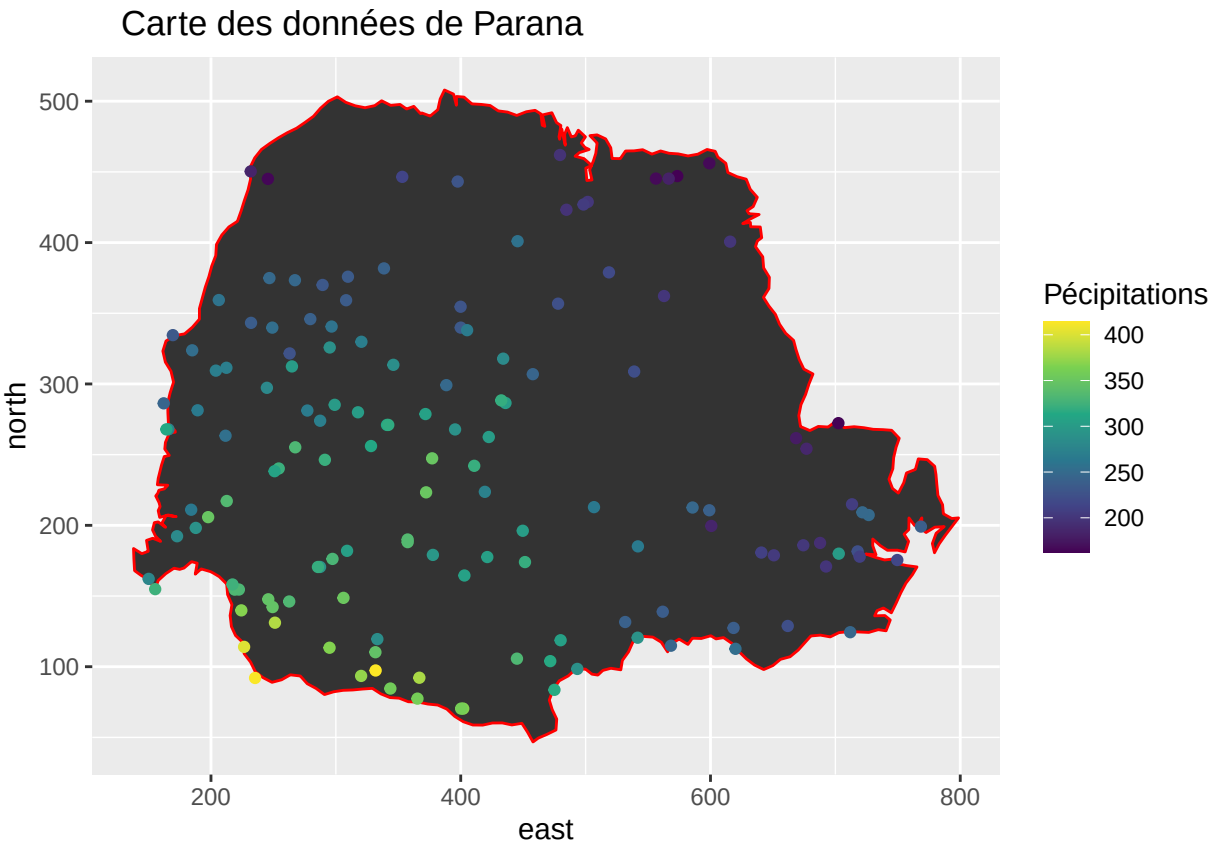
```
data_rl=lm(data~north,data=parana)
plot(parana$north,data_rl$fitted.values,type='l',col="red",xlab="North",ylab="lm(data~north)",
     main="Moyenne des précipitations")
lines(parana$north,rep(mean(parana$data),nrow(parana)))
box()
legend("bottomleft", legend = c("Moyenne des précipitations en fonction de North ",
                                "Moyenne de toutes les précipitations"),
           fill=c("red","black"))
```



On remarque que plus qu'on monte vers le nord du Parana plus la moyenne des précipitations diminue, donc la moyenne des précipitations n'est pas constante donc le processus n'est pas stationnaire, ce premier résulat nous permet à penser d'utiliser un modèle de krigeage universel, pour bien visualiser ce résultat on va regarder la dispersion des précipitations sur la carte du Parana.

La carte des données avec les frontières de l'état.

```
ggplot(parana.borders)+
  geom_polygon(aes(x=east,y=north),col="red")+
  geom_point(data=parana,aes(x = east, y = north, color = data))+
  scale_color_viridis_c("Pécipitations")+
  ggtitle("\t\t Carte des données de Parana")
```



On peut voir que la plupart des stations de mesure sont situées à l'ouest du Parana, et que les fortes précipitations sont concentrées dans la partie sud-ouest du Parana.\ La régression linéaire des précipitations et cette visualisation nous montrent qu'on doit utiliser le modèle de krigeage universel pour obtenir de bons résultats de prévision. Dans un premier temps on va réaliser un modèle de krigeage ordinaire et le comparer avec le modèle de krigeage universel.

Modèle de kigeage ordinaire

Étude variographique des précipitations

Dans cette section, on va visualiser le variogramme empirique avec la fonction variog. On ferra varier le nombre d'intervalles, les largeurs des intervalles et la distance maximale pour éstimer les paramètres necessiares afin d'ajuster un variogramme.

Le graphe ci-dessous représente le nuage variographique (les points $(||s_j - s_i||, \frac{(Z_j - Z_i)^2}{2})$) .

Il est important de noter qu'on doit tout d'abord transformer l'ensemble de données aux données de types "geodata".

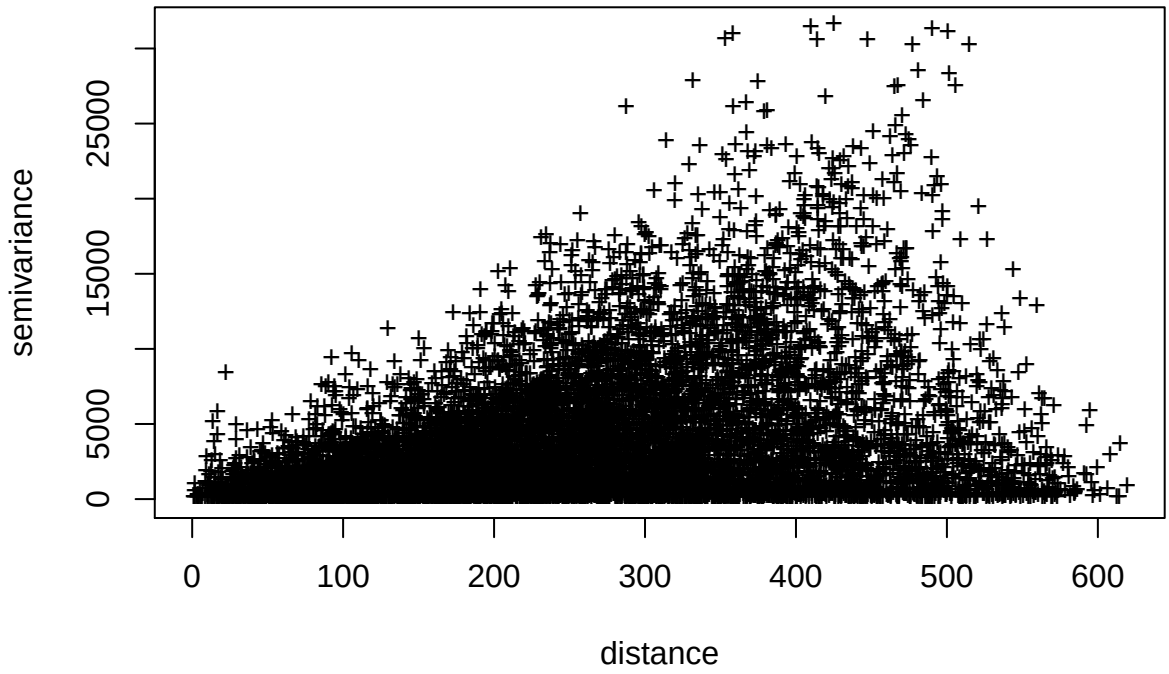
```
geodata = as.geodata(parana)#convertir les observations en geodata
```

Nuage variographique

```
vario.c = variog(geodata,op="cloud")
```

```
## variog: computing omnidirectional variogram
```

```r
plot(vario.c,main = "",pch='+')
```



Ce nuage n'est pas très lisible, il ne suffit pas pour avoir une idée sur les caractéristiques de la fonction variogramme comme la portée, le palier ou la pépite. On utilisera alors le variogramme expérimental pour une représentation de la variabilité spatiale plus visible.

— On remarque que le semi-variogramme précédent n'est pas symétrique, donc on va prendre 600 comme distance maximale.

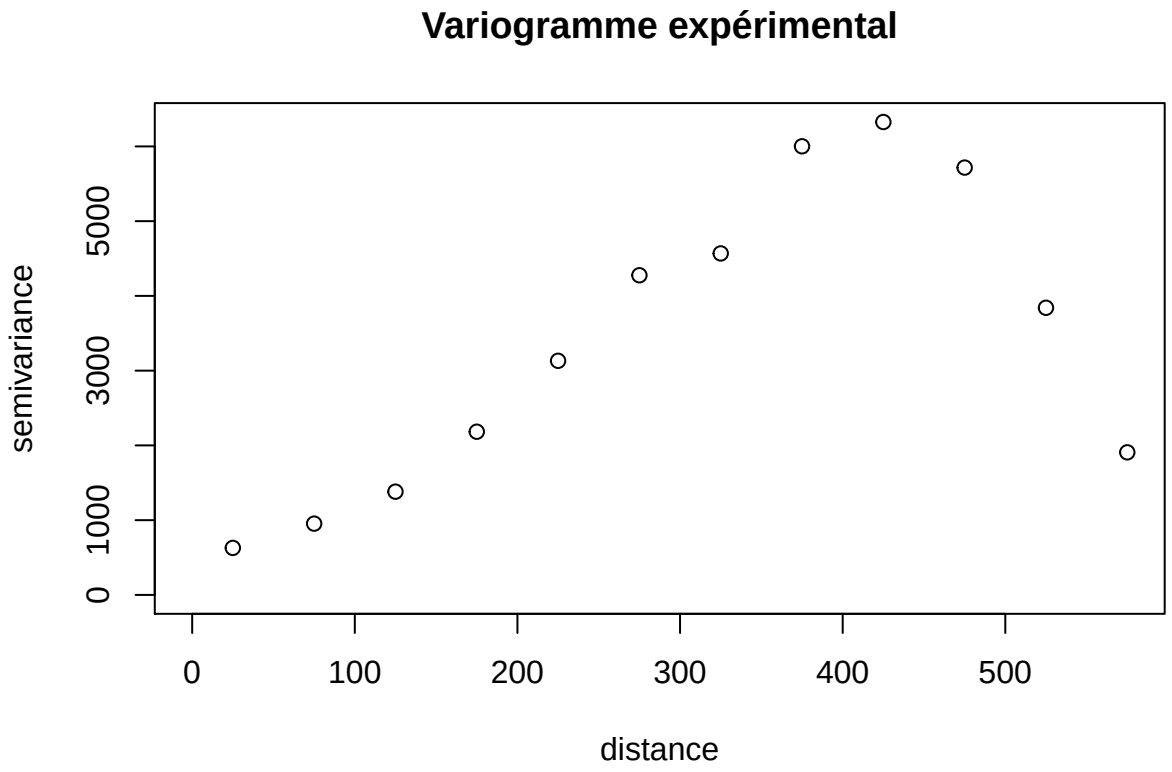
— On peut choisir des intervalles de taille égale à 50

Variogramme expérimental

```r
m.d = 600  # distance maximale
interv = seq(0,m.d,by=50)    # intervalles
p.m = 20             # nombre minimal de paire
vario.b = variog(geodata,max.dist=m.d,pairs.min=p.m,
                 breaks=interv)
```

```
## variog: computing omnidirectional variogram
```

```r
plot(vario.b,main = "Variogramme expérimental")
```



On constate que au dela d'une distance de 340, la dépendance devient faible, donc on peut estimer une portée de 340, un palier de 5000 et une pépite de 500.

Variogramme ajusté

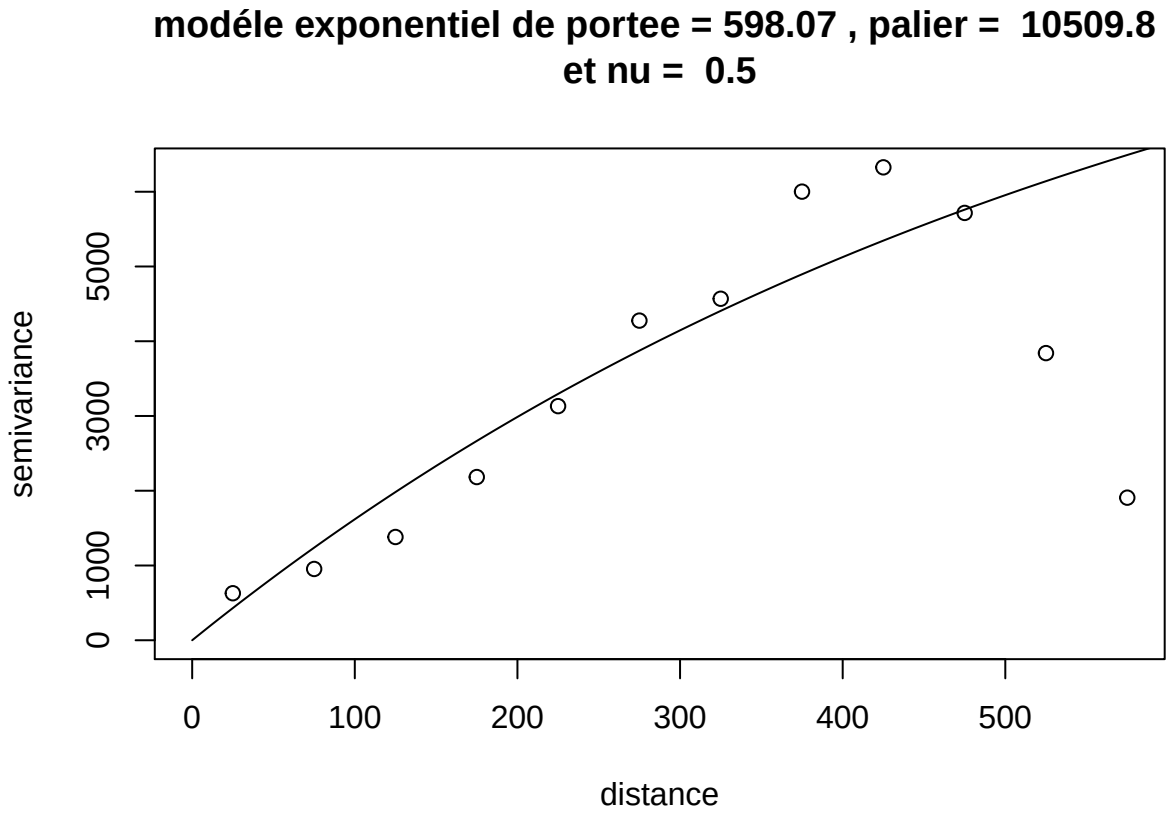
On ajuste ici deux modèles exponentiel et sphérique en donnant des paramètres initiaux pour portée (340), palier (5000) (on suppose qu'à la distance 340 le variogramme converge vers le palier et la covariance est proche de 0) déduits du variogramme experimental. On suppose que les erreurs de mésures sont négligeables donc on suppose qu'il n'y a pas un effet pépite.

Ajustument du variogramme exponentiel

```
c.m_ex = "exponential"
i.c = c(5000,340)
varioest_ex = variofit(vario.b,cov.model = c.m_ex,fix.kappa=TRUE,
ini.cov.pars=i.c,fix.nugget=T)
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```
titre = paste("modéle exponentiel de portee =",round(varioest_ex$cov.pars[2],2),
", palier = ",round(varioest_ex$cov.pars[1],2),
"\n et nu = ",round(varioest_ex$kappa,2),"\n")
plot(vario.b, main=titre)
lines(varioest_ex)
```



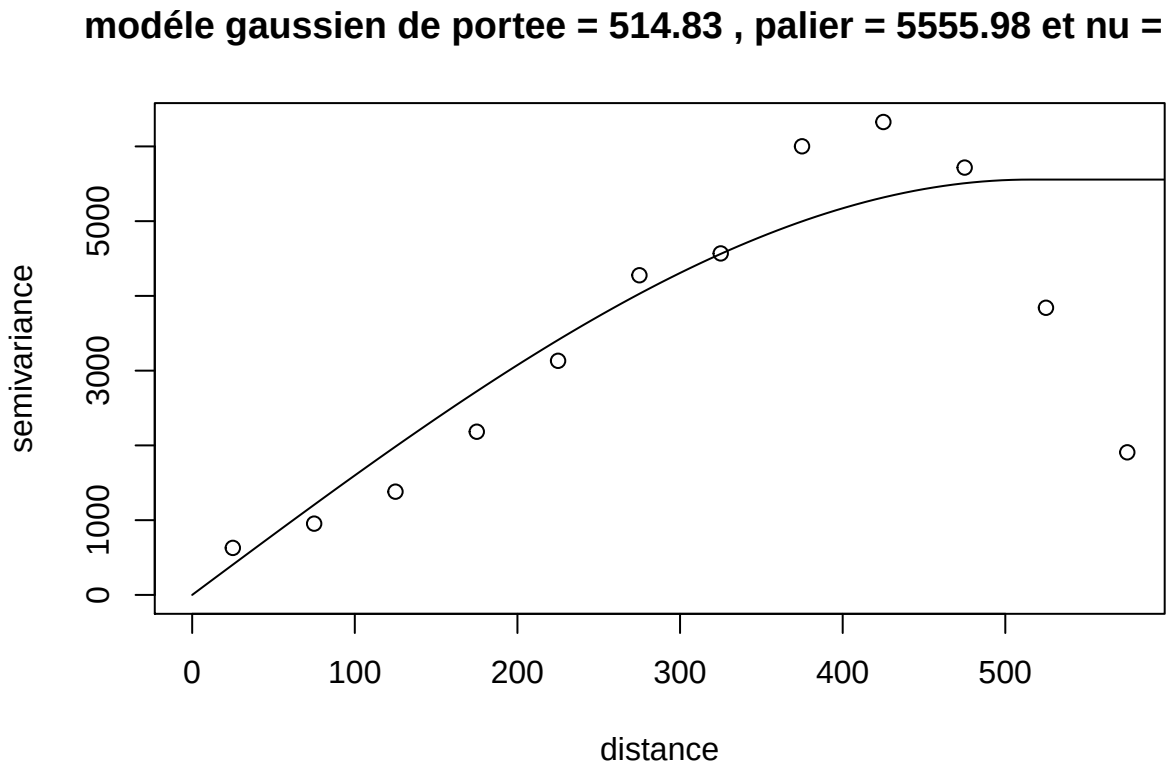
Le résultat de cet ajustument exponeniel n'est pas tout à fait compatible avec l'étude expérimentale, vu que à la distance 598 le variogramme converge vers un palier de 10509.

Ajustument du variogramme sphirique

```
c.m_sp = "spherical"
i.c = c(5000,340)
varioest_sp = variofit(vario.b,cov.model = c.m_sp,fix.kappa=TRUE,
ini.cov.pars=i.c,fix.nugget=T)
```

```
## variofit: covariance model used is spherical
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```
titre = paste("modéle gaussien de portee =",round(varioest_sp$cov.pars[2],2),
", palier =",round(varioest_sp$cov.pars[1],2),
"et nu = ",round(varioest_sp$kappa,2))
plot(vario.b, main=titre)
lines(varioest_sp)
```



Le résultat de cette étude variographique nous dit que la variabilité spatiale du processus est modélisée par un modèle sphirique de portée 514.83, un palier de 5556. On va donc utiliser ce variogramme avec ces paramétres estimés dans le krigeage.

Prédiction spatiale

Construction d'une grille sur la carte de l'état Parana

```
summary(parana.borders)
```

```
##       east           north
##  Min.   :138.0   Min.   : 46.77
##  1st Qu.:247.1   1st Qu.:138.11
##  Median :492.0   Median :232.67
##  Mean   :472.0   Mean   :273.12
##  3rd Qu.:663.7   3rd Qu.:444.31
##  Max.   :798.6   Max.   :507.93
```

On a la variable east varie entre 138 et 800, et la variable north varie entre 45 et 510. Donc on va construire une grille rectangulaire qui recouvre la carte du Parana.

```
grx = seq(138,800,by=5)
gry = seq(45,510,by=5)
grille = expand.grid(grx,gry) # l'ensemble S
```

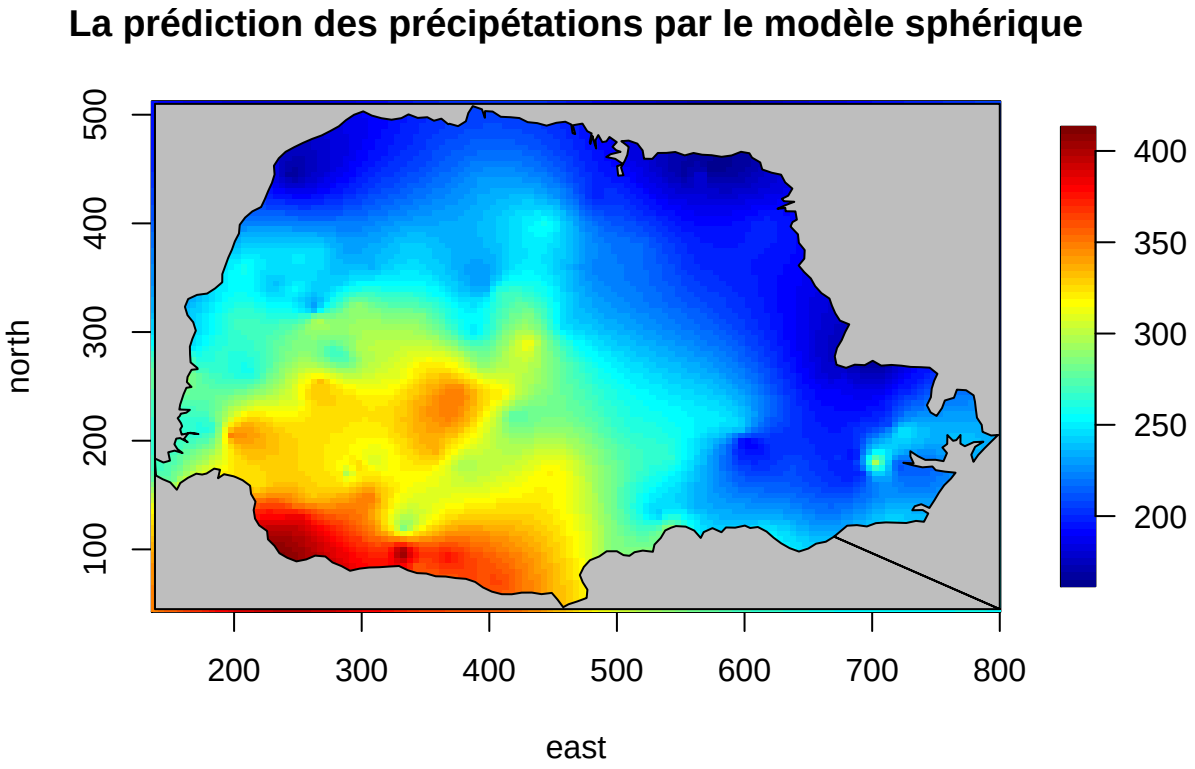
Krigeage en chaque point d'une grille

Krigeage par le modèle de sphérique

```
Kcontrol_sp = krige.control(type.krige="ok",obj.model=varioest_sp)
Ocontrol_sp = output.control(n.pred=143,simul=TRUE,thres=2)
K_sp = krige.conv(geodata,loc=grille,krige=Kcontrol_sp)
```

```
## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood
```

```
# le resultat du krigeage

Zkrige_sp = matrix(K_sp$predict,nrow=length(grx),ncol=length(gry),byrow=F)
titre = paste("La prédiction des précipétations par le modèle sphérique")
image.plot(grx,gry,Zkrige_sp,main=titre,xlab="east",ylab="north")
polygon(contour2,col = "grey")
```



On remarque alors par ce modèle de krigeage on utilisant la fonction variogramme sphérique, une forte précipétation dans le sud-ouest et des précipitations moyenne dans le centre, plus qu'on monte vers nord ou le sud-est plus que les précipitations deviennent faibles.

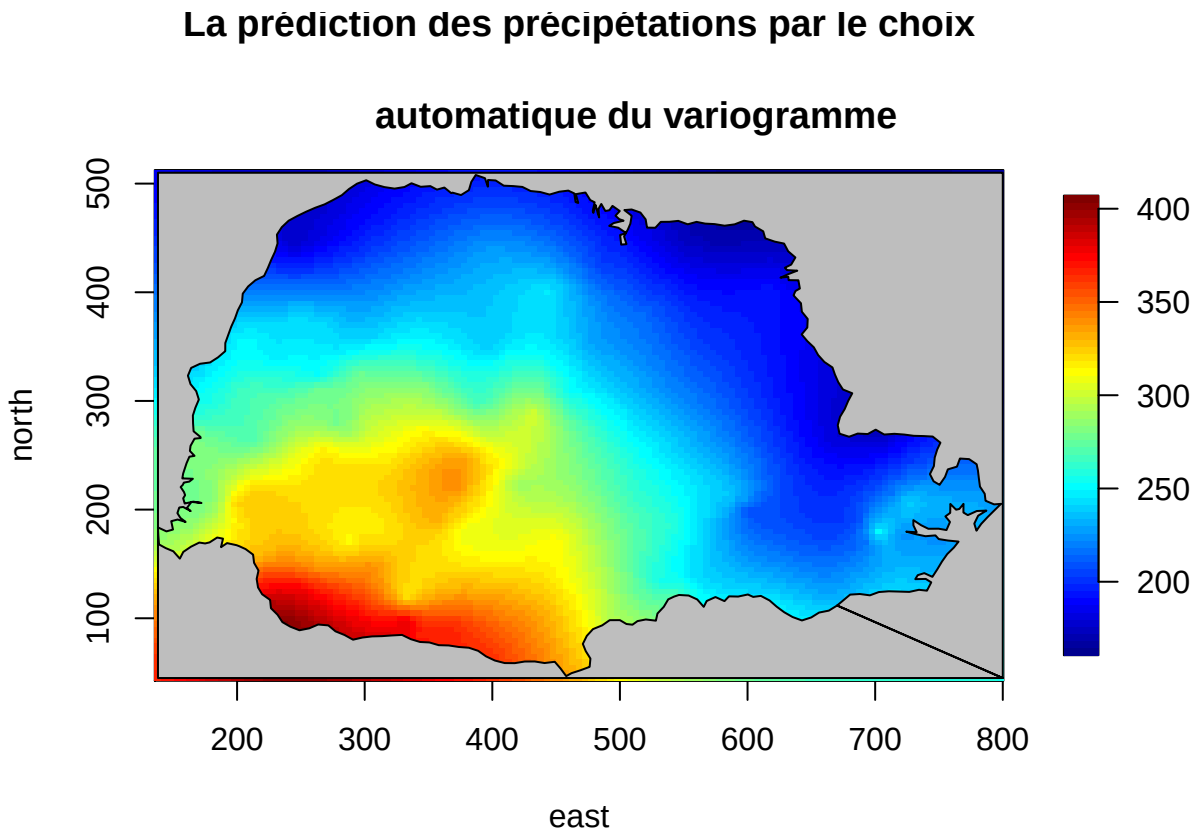
Krigeage avec choix automatique de variogramme

```
data=parana
colnames(data)=c("x","y","z")
coordinates(data) = ~x+y
###Krigeage avec choix automatique du variogramme
grille2=as.data.frame(grille)
```

```
colnames(grille2)=c("x","y")
coords=SpatialPoints(grille2)
kriging_result = autoKrige(z~1,data, coords)
```

```
## [using ordinary kriging]
```

```
Zkrige_auto=kriging_result$krige_output$var1.pred
Zkrige_auto_mat = matrix(Zkrige_auto,nrow=length(grx),ncol=length(gry),byrow=F)
image.plot(grx,gry,Zkrige_auto_mat,main="La prédiction des précipétations par le choix\n
           automatique du variogramme\n",xlab="east",ylab="north")
polygon(contour2,col = "grey")
```



On remarque que par ce krigeage automatique on a obtenu les mˆeme résultats de précipétation comme dans le modèle précédent, sauf au centre de la carte on visualise une petite différence au niveau de krigeage automatique les prévisions des précipétations sont un peu augmentées.

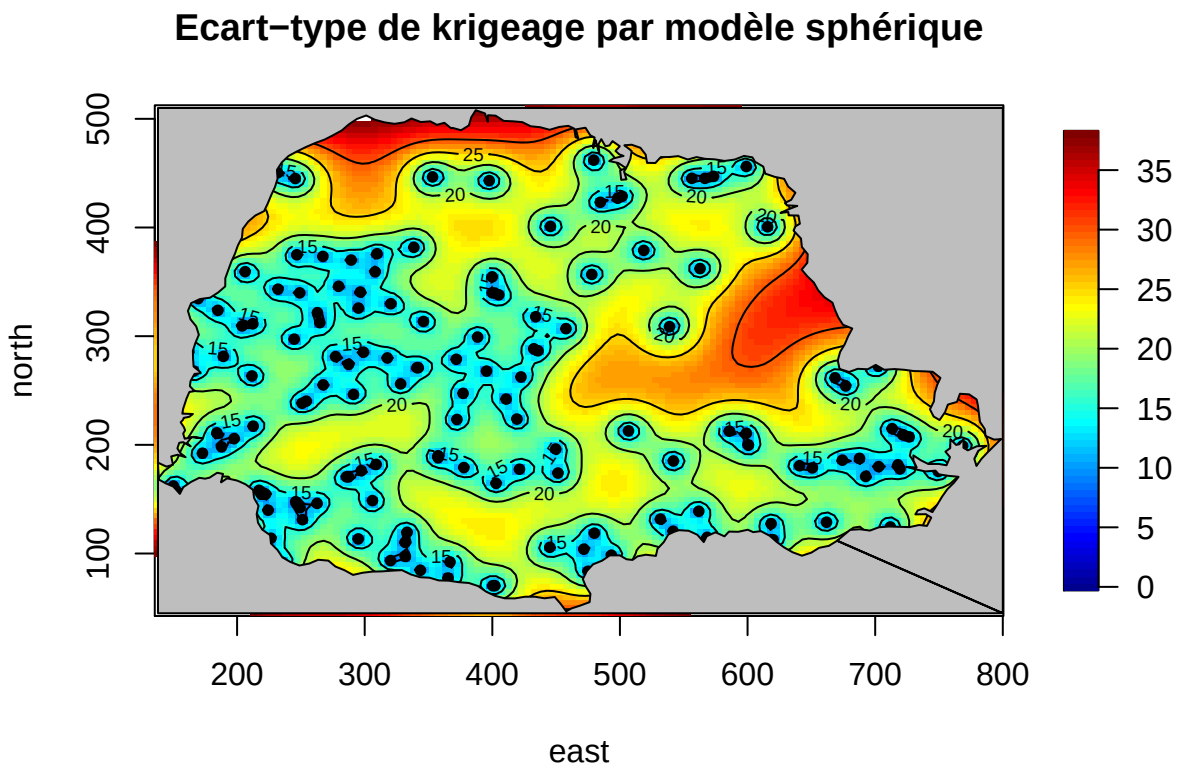
La comparaison des résultats

Pour comparer ces résultats on va comparer les cartes obtenues par krigeage selon l'erreur de prévision.

Variance de l'erreur de prévision du modèle sphérique

```
s=apply(cbind(K_sp$krige.var,rep(0,length(K_sp$krige.var))),1,max)
Sigma = sqrt(matrix(s,nrow=length(grx),ncol=length(gry),
                    byrow=F))
titre = "Ecart-type de krigeage par modèle sphérique"
```

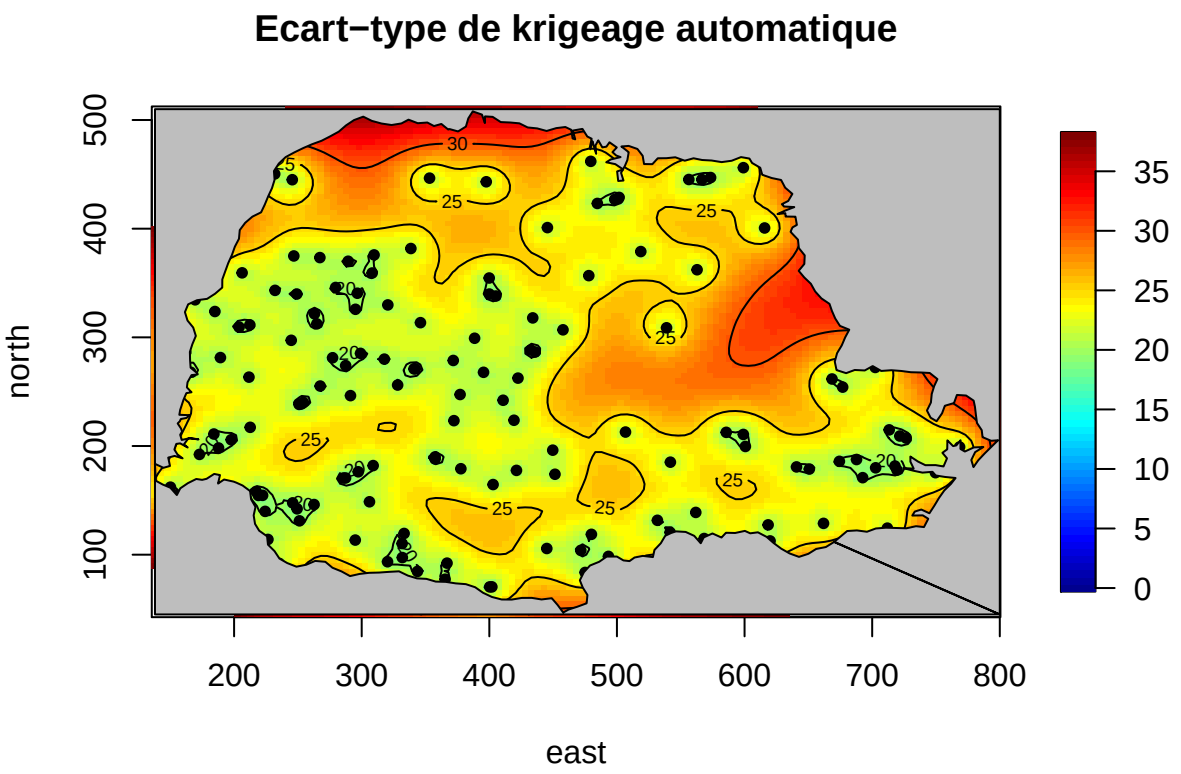
```
image.plot(grx,gry,Sigma,zlim=c(0,38),main=titre,xlab="east",ylab="north")
contour(grx,gry,Sigma,levels=seq(0,30,5),add=TRUE,pch=15)
points(parana$east,parana$north,pch=20)
polygon(contour2,col = "grey")
```



D'après ce graphique on remarque que l'erreur de prévision est un peu petite autours des stations de mésures (entre 10 et 15), et plus on s'éloigne des stations de mésures plus que l'erreur augmente, à cause d'une manque de données.

Variance de l'erreur de prévision du modèle automatique du variogramme

```
s=apply(cbind(kriging_result$krige_output$var1.var,rep(0,length(kriging_result$krige_output$var1.var)))
Sigma = sqrt(matrix(s,nrow=length(grx),ncol=length(gry),
                    byrow=F))
titre = "Ecart-type de krigeage automatique"
image.plot(grx,gry,Sigma,zlim=c(0,38),main=titre,xlab="east",ylab="north")
contour(grx,gry,Sigma,levels=seq(0,30,5),add=TRUE,pch=15)
points(parana$east,parana$north,pch=20)
polygon(contour2,col = "grey")
```



On visualise que l'erreur du modèle automatique du variogramme est grande par rapport au modèle sphérique, on trouve qu'il donne pas des bonnes prévisions mˆeme autours des stations (écart type varie entre 18 et 20). Pour les deux modèle de plus qu'on s'éloigne des stations plus que l'erreur augmente. les prévisions trouvées sont pas bonnes avec le krigeage ordinaire meme avec le choix automatique qui choisit le variogramme le plus adapter aux données. Ces résultats sont logique, vu qu'on veut appliquer le krigeage ordinaire sur un processus qui n'est pas stationnaire.

Krigeage universel

Pour éffectuer le krigeage universel on va faire le krigeage ordinaire par le choix automatique sur les résidus de la régression linéaire entre la variable north et la valeur des précipitations.

Régression linéaire

```
reg=lm(data~north,data=parana)
resdata=parana
res=residuals(reg)
resdata$data=res
fit=fitted(reg)
```

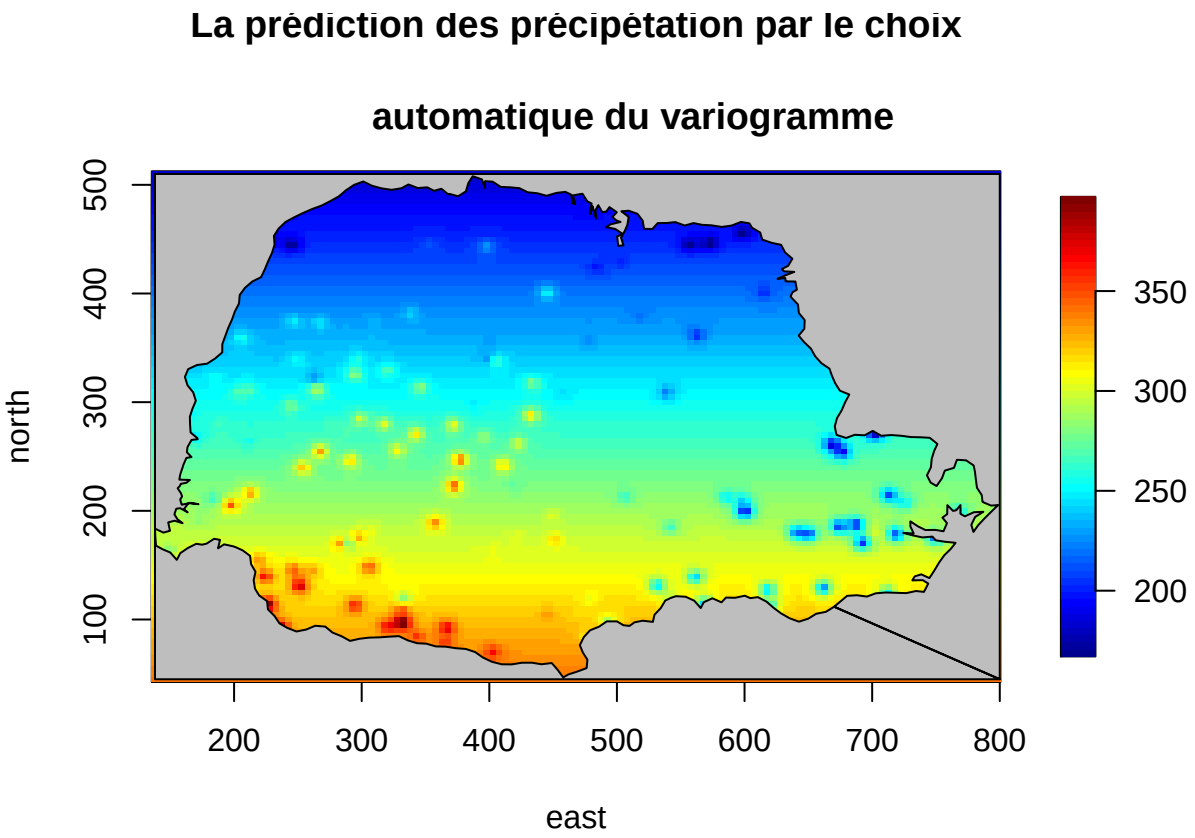
Krigeage ordinaire des résidus

```
data=resdata
colnames(data)=c("x","y","z")
coordinates(data) = ~x+y
###Krigeage ordinaire des résidus avec choix automatique du variogramme
grille2=as.data.frame(grille)
colnames(grille2)=c("x","y")
coords=SpatialPoints(grille2)
kriging_result = autoKrige(z~1,data, coords)
```

```
## [using ordinary kriging]
Zkrige_auto=kriging_result$krige_output$var1.pred
```

Le variogramme le plus adapter aux résidus par le choix automatique est le variogramme de Matern avec une portée de 7 et un palier de 750.

```
###Tendance dans la grille ##
Zreg=coef(reg)[1]+grille[,2]*coef(reg)[2]
### krigeage ordinaire
Zkrige_auto_mat = matrix(Zkrige_auto+Zreg,nrow=length(grx),ncol=length(gry),byrow=F)
image.plot(grx,gry,Zkrige_auto_mat,main="La prédiction des précipétation par le choix\n
           automatique du variogramme\n",xlab="east",ylab="north")
polygon(contour2,col = "grey")
```



D'après ce graphique, on remarque une différence entre ce modèle de krigeage universel et le modèle de krigeage ordinaire par le modèle sphérique, dans ce modèle les précipitations sont fortes dans la partie inférieure de la carte et plus qu'on monte plus que les précipétations deviennent faibles.

Conclusion :

Dans cette partie on a fait deux études sur un processus non stationnaire (précipitations), le krigeage ordinaire nous a pas donné de bons résultats de prévision la variance d'erreur est plus élevée. En revanche le modèle de krigeage universel est le plus adapter à notre étude.

Econométrie spatiale

Analyse exploratoire de données spatiales

La base de données qu'on va utliser dans cette partie est intitulée "yields", la base est constituée des fichiers suivants :
— rosas1999.shp informations de géométrie (coordonnées).
— rosas1999.shx indice de position des géométries.
— rosas1999.dbf table des attributs.
Cette base de données concerne un champ en Argentine qui comprend 4 types de sols différents (Slope W, Hilltop, Slope E et Low E) et elle contient des indications sur la production de maïs de 1738 parcelles de culture de maïs. On va analyser l'impact de l'utilisation de nitrogène comme engrais sur les rendements du maïs (mesurés en nombre de quintaux par hectare).

Chargement de fichiers de données

Pour importer les données geodata, on va utliser la fonction readOGR qui importe tous les fichiers "rosa1999".

```
yields <- readOGR(dsn='/home/samnouni/Bureau/M2 ISN/statistique spatiale/projet', layer = 'rosas1999' )
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/home/samnouni/Bureau/M2 ISN/statistique spatiale/projet", layer: "rosas1999"
## with 1738 features
## It has 34 fields
```

La Table ci-dessous présente les variables incluses dans notre base de données :

Libellé	Description
TOP2	variable muette : 1 si parcelle est de type Slope E (terrain pentu orienté Est) et 0 sinon
TOP3	variable muette : 1 si parcelle est de type Hilltop (terrain au sommet de la colline) et 0 sinon
TOP4	variable muette : 1 si parcelle est de type Slope W (terrain pentu orienté Ouest) et 0 sinon
NXTOP2 à NXTOP4	interaction Nitrogène - zone topographique
N2XTOP2 à N2XTOP4	interaction Nitrogène au carré - zone topographique
LONGITUDE	longitude
LATITUDE	latitude
OBS	numéro d'observation
YIELD100	rendement du maïs (Kg par hectare)
N	quantité de nitrogène utilisée (kg par hectare)
N2	quantité de nitrogène au carré
TOPO	zone : Low E (1), Slope E (2), Hilltop (3), Slope W (4). Low E représente un terrain en vallée exposé à l'Est
BV	lunimosité (proxy pour une faible teneur en matière organique)
BV2	luminosité au carré
NXBV	interaction Nitrogène - luminosité
BVXT2 à BVXT4	interaction luminosité -zome topographique
BV2XT2 à BV2XT4	interaction luminosité au carré et zone topographique

Libellé	Description
SAT	rayonnement (proxy pour une faible teneur en matière organique)
SAT2	rayonnement au carré
NXSAT	interaction Nitrogène -rayonnement
SATXT2 à SATXT4	interaction rayonnement - zone topographique
SAT2XT2 à SAT2XT4	interaction carré du rayonnement - zone topographique

Afin d'obtenir des résultats plus lisibles, on va tout d'abord multiplier la variable YIELD par 100, pour obtenir un rendement exprimé en kg par hectare, plutôt qu'en quintaux par hectare. On va nommer cette nouvelle variable YIELD100.

```
data=yields@data
data=renomme.variable(data, "YIELD", "YIELD100")
data$YIELD100=data$YIELD100*100
```

Analyse decriptive

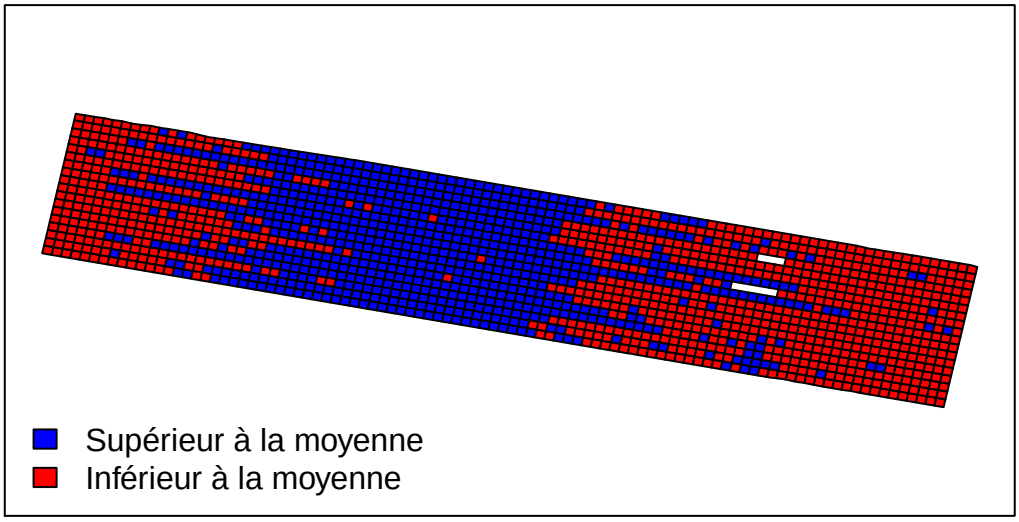
```
summary(data1[,c(1:3,13,14)])
```

```
##  TOP2     TOP3     TOP4        YIELD100          N
##  0:1370   0:1384   0:1183   Min.   :3123   Min.   :  0.00
##  1: 368   1: 354   1: 555   1st Qu.:5863   1st Qu.: 29.00
##                             Median :6554   Median : 66.00
##                             Mean   :6456   Mean   : 64.21
##                             3rd Qu.:7055   3rd Qu.:106.00
##                             Max.   :9038   Max.   :131.50
```

Les rendements du mais YILED100 en kg par hectare sont comprises entre 3123 et 9038 avec une moyenne de 6456, et la quantité du nitrogéne utilisée pour les parcelles varie entre 0 et 131.5 pour une moyenne de 64.21. On remarque aussi dans cette étude on a 1370 parcelles de type orienté Est, et 368 parcelles de type slop E, ainsi que 354 parcelles de type Hilltop et 555 parcelle de type Slop W.

Représentation graphique des données

```
quadrant <- vector(mode="numeric",length=nrow(data))

quadrant[data$YIELD100>mean(data$YIELD100) ] <- 1
quadrant[data$YIELD100<mean(data$YIELD100) ] <- 0
brks <- c(0,1)
colors <- c("blue","red")
plot(yields,col=colors[findInterval(quadrant,brks,all.inside=FALSE)],
     main="Les parcelles selon la quantité du nitrogène")
box()
legend("bottomleft", legend = c("Supérieur à la moyenne","Inférieur à la moyenne"),fill=colors,bty="n")
```



Le graphique ci-dessus représente des parcelles ayant une valeur du rendement du maïs (YILED100) plus grande que la moyenne (en rouge), et des parcelles avec une valeur du rendement du maïs plus petite que la moyenne (en bleu).

Analyse exploratoire de données spatiales

On veut construire une matrice de poids basée sur la contiguïté au sens de la reine à l'ordre 1. Pour cela on va utliser la fonction "poly2nb" avec le choix queen.

```
# une liste de voisins basée sur les régions ayant des frontières contiguës au sens de la reine
queen=poly2nb(yields, queen = T)
# matrice de poids des voisinages
mat.vois = nb2mat(queen,style ="W")
```

L'autocorrélation spatiale mesure la corrélation d'une variable avec elle-même, lorsque les observations sont décalées dans l'espace.
On va calculer la statistique du test de Moran pour analyser la dépendance spatiale globale de la variable YIELD100 dans chaque différent parcelles du champ.

L'analyse et l'interprétation de l'autocorrélation spatiale globale dans la variable YIELD100

```
poids.vois = nb2listw(queen,style="W")
print(moran.test(data$YIELD100,poids.vois)) ## randomisation
```

```
##
##  Moran I test under randomisation
##
## data:  data$YIELD100
## weights: poids.vois
##
## Moran I statistic standard deviate = 56.5, p-value < 2.2e-16
## alternative hypothesis: greater
```

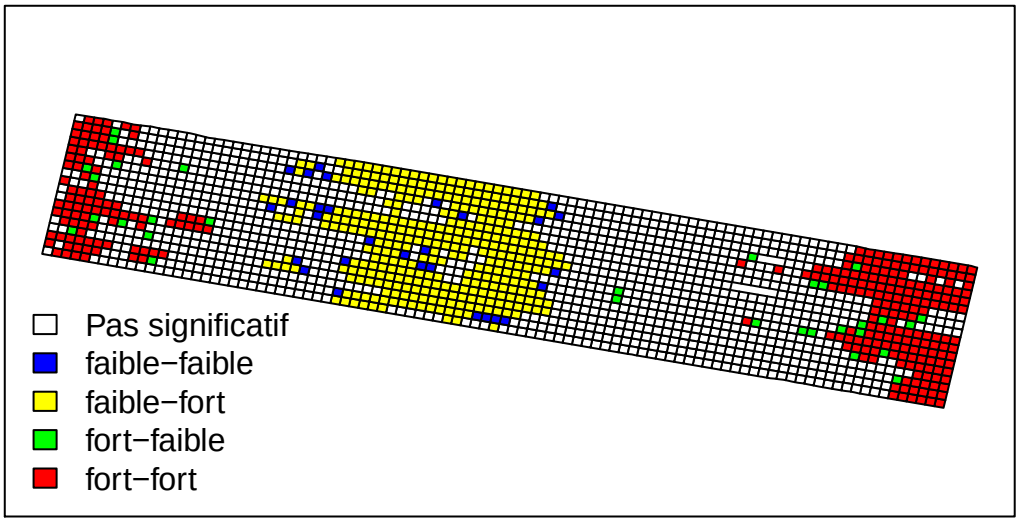
```
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.7008644728     -0.0005757052      0.0001541267
```

La statistique du test de Moran nous indique qu'il existe autocorrélation spatiale globale positive dans la variable YIELD100. Donc il existe des clusters, les parcelles de chaque cluster ont des caratéristiques communes concernant les rendements du maïs.

L'analyse de la présence d'autocorrélation spatiale locale dans la variable YIELDS100

On va effectuer l'analyse avec les I de Moran locaux permettent de mesurer pour la variable YIELD100 la dépendance locale entre une unité spatiale et les unités spatiales voisines. Ils permettent d'identifier les regroupements similaires autour d'un site donné et les zones de non-stationnarité spatiale locale.
$I_i = \sum_j w_{ij}(Z_j - \overline{Z})(Z_i - \overline{Z})$

```
moran.local = localmoran(data$YIELD100,poids.vois,zero.policy=TRUE)
#print(moran.local)

YIELD100red <- scale(data$YIELD100)
YIELD100red=as.data.frame(YIELD100red)
colnames(YIELD100red)=c("YIELD100red")
moran.local <- localmoran(YIELD100red$YIELD100red,poids.vois,zero.policy=TRUE)

quadrant <- vector(mode="numeric",length=nrow(moran.local))
# Centrer/reduire l'indice locam
    moran.localred <- scale(moran.local[,1])

# significance threshold
    signif <- 0.05

#builds a data quadrant
quadrant[YIELD100red$YIELD100red >0 & moran.localred>0] <- 4
quadrant[YIELD100red$YIELD100red <0 & moran.localred<0] <- 1
quadrant[YIELD100red$YIELD100red <0 & moran.localred>0] <- 2
quadrant[YIELD100red$YIELD100red >0 & moran.localred<0] <- 3
quadrant[moran.local[,5]>signif] <- 0
brks <- c(0,1,2,3,4)
colors <- c("white","blue","yellow","green","red")
plot(yields,col=colors[findInterval(quadrant,brks,all.inside=FALSE)])
box()
legend("bottomleft", legend = c("Pas significatif","faible-faible",
                                "faible-fort","fort-faible","fort-fort"),
           fill=colors,bty="n")
```

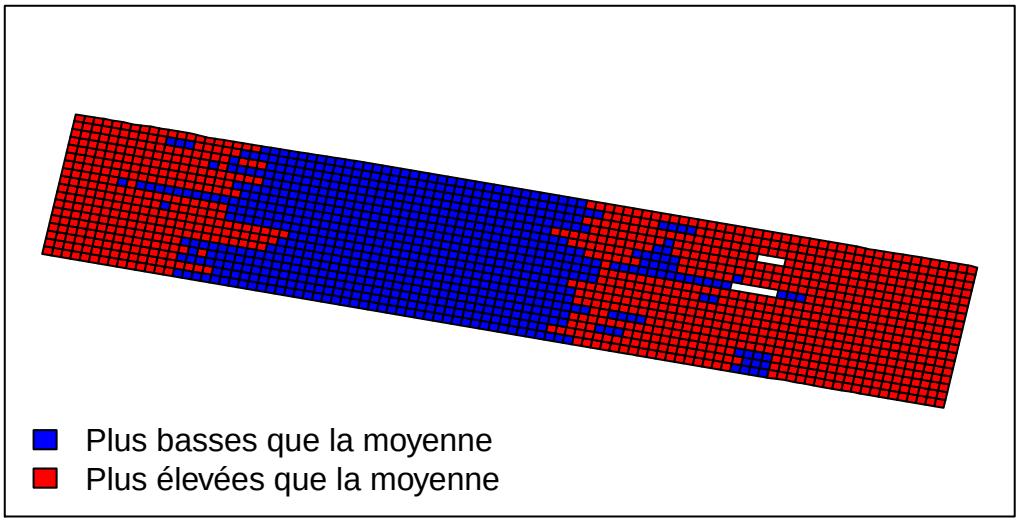


D'après ce graphique :
— les parcelles en rouge dans les extrémités du champ siginfient qu'il existe une forte autocorrélation spatiale locale dans la variable YIELDS100 avec des valeurs importantes.
— Les percelles en bleu au milieur du champ signifient que l'autocorrélation spatiale locale est faible dans la variable YIELDS100 avec des valeurs petites.
— Les percelles en jaune vers le milieur du champ signifient que l'autocorrélation spatiale locale est forte dans la variable YIELDS100 avec des valeurs petites.
— Les percelles en vert à l'extrémité du champ signifient que l'autocorrélation spatiale locale est faible dans la variable YIELDS100 avec des valeurs importante.

L'analyse de l'autocorrélation spatiale locale dans la variable YIELDS100 à l'aide des statistiques de Getis

Getis propose un indicateur permettant de détecter les dépendances spatiales locales qui n'apparaissent pas dans l'analyse globale.
Indicateur de Getis :
$G_i = \frac{\sum_j w_{ij} Z_{s_i}}{\sum_j w_{ij}}$
$G_i > 0$ indique un regroupement de valeurs plus élevées que la moyenne.
$G_i < 0$ indique un regroupement de valeurs plus basses que la moyenne.
La fonction "localG" permet d'utiliser cet indicateur.

```
local.get=localG(data$YIELD100,poids.vois)
quadrant <- vector(mode="numeric",length(local.get))
quadrant[local.get<0] <- 0
quadrant[local.get>0] <- 1
brks <- c(0,1)
colors <- c("blue","red")
plot(yields,col=colors[findInterval(quadrant,brks,all.inside=FALSE)])
box()
legend("bottomleft", legend = c("Plus basses que la moyenne","Plus élevées que la moyenne"),
           fill=colors,bty="n")
```



On remarque d'après ce graphique la statistique de Getis a identifié deux regreoupements comme suit :
— Le groupe des parcelles en bleu au milieu du champ signifient un regroupement de valeurs plus basses que la moyenne.
— Le groupe des parcelles en rouge vers l'extrémité du champ signifient un regroupement de valeurs plus élevées que la moyenne.

La différence avec l'analyse basée sur le I de Moran local et la statistique de Getis locale

Avec la statistique de Getis on a identifié deux groupes avec deux caractéristiques différentes; un groupe avec des valeurs du rendement du maïs plus basses que la moyenne et un autre groupe avec des valeurs du rendement du maïs plus élevées que la moyenne. Tandis que par le test de Moran on a identifié cinq groupes, dont un groupe contient des parcelles avec une valeur de la statistique de Moran qui n'est pas significative (moins de 5%). Avec le test de Getis on n'a pa pu identifié les autocorrélations spatiales qui ne sont pas signifatives.

Econométrie spatiale

Dans cette partie, on considére un seuil de significativité à 10%, pour répondre aux questions.

La Construction de la spécification économétrique présentée ci-dessous

Régression spatiale

Considérons le modèle linéaire classique suivant :

$$YIELD100 = \beta_0 + \beta_1 N + \beta_2 N_2 + \beta_3 TOP_2 + \beta_4 TOP_3 + \beta_5 TOP_4 + \varepsilon$$

On va déterminer les coefficients de ce modèle après on va réaliser un test d'indépendance pour les résidus

```
YIELD100.lm <- lm(YIELD100 ~ N + N2 + TOP2 + TOP3 +TOP4, data=data)
summary(YIELD100.lm)
```

```
##
## Call:
## lm(formula = YIELD100 ~ N + N2 + TOP2 + TOP3 + TOP4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2212.93  -374.39     8.52   364.02  2827.22
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.597e+03  3.786e+01 174.254  < 2e-16 ***
## N            1.144e+01  1.067e+00  10.722  < 2e-16 ***
## N2          -3.510e-02  7.627e-03  -4.602 4.49e-06 ***
## TOP2        -6.316e+02  3.922e+01 -16.105  < 2e-16 ***
## TOP3        -1.769e+03  3.965e+01 -44.604  < 2e-16 ***
## TOP4        -5.265e+02  3.535e+01 -14.893  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 560.9 on 1732 degrees of freedom
## Multiple R-squared:  0.6004, Adjusted R-squared:  0.5993
## F-statistic: 520.5 on 5 and 1732 DF,  p-value: < 2.2e-16
```

```
anova(YIELD100.lm)
```

```
## Analysis of Variance Table
##
## Response: YIELD100
##             Df    Sum Sq   Mean Sq   F value    Pr(>F)
## N            1 167614773 167614773  532.7014 < 2.2e-16 ***
## N2           1   5553377   5553377   17.6493 2.791e-05 ***
## TOP2         1    442021    442021    1.4048    0.2361
## TOP3         1 575458327 575458327 1828.8808 < 2.2e-16 ***
## TOP4         1  69792868  69792868  221.8107 < 2.2e-16 ***
## Residuals 1732 544974741    314651
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Selon la fonction “ANOVA” la variable TOP2 n’est pas significative, donc on peut la retirer de notre modèle. On s’intéresse maintenant à faire un test sur les résidus pour le nouveau modèle, on propose un test de Moran

```
YIELD100.lm = lm(YIELD100 ~ N + N2 +TOP3 +TOP4, data=data)
res.lm=residuals(YIELD100.lm)
moran.test(res.lm,poids.vois)
```

```
##
##  Moran I test under randomisation
##
## data:  res.lm
## weights: poids.vois
##
## Moran I statistic standard deviate = 49.591, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.6149254488     -0.0005757052      0.0001540466
```

On a la statistique du test de Moran est plus grand que 0, donc on rejette l'hypothése d'indépendance des résidus. Alors, on a une dépendance des résidus, le modèle classique ne suffit pas pour expliquer la variable YIELD100. Donc, on doit chercher o'u se trouve l'autocorrélation spatiale, est ce que dans les variables explicatives ou dans la variable YIELD100 (On a déjà trouvé qu'il existe des autocorrélations spatiales dans cette variable).
Pour cela, on va utiliser d'autres modèles adéquats pour modéliser l'autocorrélation spatiale en utilisant la matrice de connectivité (La matrice des voisinages)

la spécification spatiale la plus adéquate pour modéliser l'autocorrélation spatiale

On propose le modèle SAR, pour modéliser l'autocorrélation spatiale dans la variable YIELD100, et le modèle SLX pour modéliser l'autocorrélation spatiale dans les variables explicatives. Ces deux modèles s'écrivent sous la forme suivante :

SAR

$$YIELD100 = \rho WYIELD100 + \beta_0 + \beta_1 N + \beta_2 N2 + \beta_4 TOP3 + \beta_5 TOP4 + \varepsilon$$

SLX

$$YIELD100 = \rho W(1, N, N2, TOP3, TOP4) + \beta_0 + \beta_1 N + \beta_2 N2 + \beta_4 TOP3 + \beta_5 TOP4 + \varepsilon$$

```
#### SAR ####
YIELD100.lagsarlm=lagsarlm(YIELD100.lm,listw=poids.vois,type="lag",method="eigen",data=data)
summary(YIELD100.lagsarlm)
```

```
##
## Call:lagsarlm(formula = YIELD100.lm, data = data, listw = poids.vois,
##     type = "lag", method = "eigen")
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -2248.356  -249.586   -14.902   246.006  2713.845
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                Estimate  Std. Error  z value  Pr(>|z|)
## (Intercept)  1.3594e+03  1.2438e+02  10.9294 < 2.2e-16
## N            1.1184e+01  7.9521e-01  14.0640 < 2.2e-16
## N2          -3.1006e-02  5.6846e-03  -5.4544 4.915e-08
## TOP3        -4.0884e+02  3.6924e+01 -11.0725 < 2.2e-16
## TOP4        -7.0771e+01  2.3405e+01  -3.0237  0.002497
##
## Rho: 0.72406, LR test value: 1091.4, p-value: < 2.22e-16
## Asymptotic standard error: 0.018429
##     z-value: 39.29, p-value: < 2.22e-16
## Wald statistic: 1543.7, p-value: < 2.22e-16
##
## Log likelihood: -13039.54 for lag model
## ML residual variance (sigma squared): 174400, (sigma: 417.62)
## Number of observations: 1738
## Number of parameters estimated: 7
```

```
## AIC: 26093, (AIC for lm: 27182)
## LM test for residual autocorrelation
## test value: 6.376, p-value: 0.011568
```

```
res.sar <- residuals(YIELD100.lagsarlm)## résidus non-autocorrélés
moran.test(res.sar,poids.vois)
```

```
##
##  Moran I test under randomisation
##
## data:  res.sar
## weights: poids.vois
##
## Moran I statistic standard deviate = 1.7626, p-value = 0.03898
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.0212859829     -0.0005757052      0.0001538308
```

D'après ces résultats, on a trouvé que les résidus du modèle SAR sont faiblements autocorrélés puisque le test de Moran nous donne une statistique du test de 0.02 pas très proche de 0. On a aussi le paramètre autorégressif (Rho) est siginificativement différent de zéro, la p-valeur du test est très petite et selon le critère d'AIC le modèle SAR est le plus pertinent que le modèle de regréssion classique puisque la valeur d'AIC est petite. On remarque aussi que tous les coefficients liés aux variables explicatives sont siginificatifs.

```
### SLX ###
# 1) Création des décalages spatiaux des X
# Utilisation de la fonction lag.listw()
data$WN=lag.listw(poids.vois,data$N)
data$WN2=lag.listw(poids.vois,data$N2)
data$WTOP3=lag.listw(poids.vois,data$TOP3)
data$WTOP4=lag.listw(poids.vois,data$TOP4)
# Estimation par MCO du modèle de base en incluant les décalages spatiaux des X
YIELD100.slx<- lm(YIELD100~N+WN+N2+WN2+TOP3+WTOP3+TOP4+WTOP4 , data=data)
summary(YIELD100.slx)
```

```
##
## Call:
## lm(formula = YIELD100 ~ N + WN + N2 + WN2 + TOP3 + WTOP3 + TOP4 +
##     WTOP4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2622.53  -367.04   -34.73   355.55  2953.16
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.791e+03  1.306e+02  44.326  < 2e-16 ***
## N            1.493e+01  1.264e+00  11.813  < 2e-16 ***
## WN           2.922e+01  5.726e+00   5.103 3.72e-07 ***
## N2          -5.821e-02  9.098e-03  -6.399 2.01e-10 ***
## WN2         -2.353e-01  4.251e-02  -5.535 3.59e-08 ***
## TOP3         2.313e+02  2.537e+02   0.912    0.362
## WTOP3       -1.811e+03  2.583e+02  -7.012 3.36e-12 ***
## TOP4         1.425e+01  3.490e+02   0.041    0.967
## WTOP4       -2.557e+02  3.514e+02  -0.728    0.467
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 579.6 on 1729 degrees of freedom
## Multiple R-squared:  0.5741, Adjusted R-squared:  0.5721
## F-statistic: 291.3 on 8 and 1729 DF,  p-value: < 2.2e-16
```

```
res.slx <- residuals(YIELD100.slx)## résidus non-autocorrélés
moran.test(res.slx,poids.vois)
```

```
##
##  Moran I test under randomisation
##
## data:  res.slx
## weights: poids.vois
##
## Moran I statistic standard deviate = 49.912, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.6189053459     -0.0005757052      0.0001540438
```

D'après ces résultats, on a trouvé que les résidus du modèle SLX sont fortement autocorrélés puisque le test de Moran nous donne une statistique du test de 0.62 très grande que 0. Donc ce modèle n'est pas adéquat pour notre étude. Alors, on propose de choisir le modèle SAR pour la modélisation de l'autocorrélation spatiale.

L'impact d'une variation de la quantité d'engrais, évaluée à la valeur moyenne de l'échantillon, sur le rendement du maïs

```
rh_sar=YIELD100.lagsarlm$rho
betarm=YIELD100.lagsarlm$coefficients[2]
## La fonction invIrW calcule l'inverse de (I - rho*W)
Sw=invIrW(poids.vois, rh_sar)*betarm ##
n=length(YIELD100.lagsarlm$residuals)
i_n=matrix(1,nrow=n)
direct=diag(Sw)
indirect=Sw-diag(diag(Sw))
ADE=mean(direct) ## impact direct moyen
AIE=1/n*t(i_n)%*%indirect%*%i_n ## impact indirect moyen
ADE
```

```
## [1] 12.62562
```

```
AIE
```

```
##          [,1]
## [1,] 27.90454
```

On remarque que l'impact direct moyen d'une variation de la quantité d'engrais est significatif (la somme moyenne des coefficients liès à chaque observation du nitrogène utilisé pour chaque parcelle est différente de 0). On a l'impact direct moyen de l'utilisation du nitrogène comme engrais sur le rendement du maïs est positif 12.62, et l'impact indirect moyen est aussi positif 27.9, donc le rendement du maïs de chaque percelle est impacté par l'utilisation du nitrogène de ses voisins.

Estimation du modèle (1) en intégrant le décalage spatial de la variable dépendante.

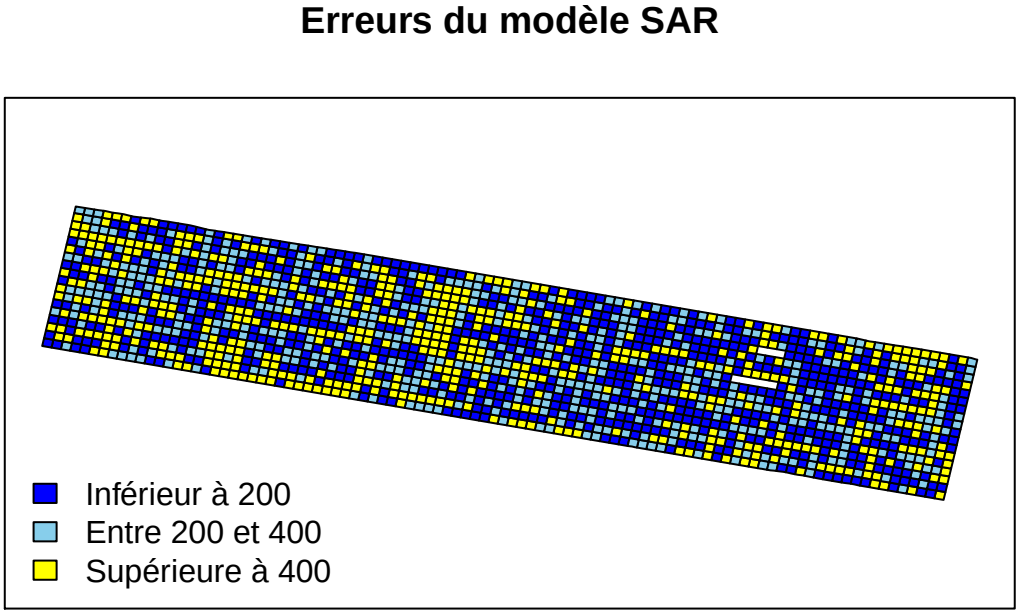
Dans la question 2, on déjà estimé le modèle SAR (en intégrant le décalage spatial de la variable dépendante) le modèle s'écrit sous la forme suivante :

$$YIELD100 = \rho WYIELD100 + \beta_0 + \beta_1 N + \beta_2 N2 + \beta_4 TOP3 + \beta_5 TOP4 + \varepsilon$$

Comparaison des résultats obtenus avec ceux des MCO

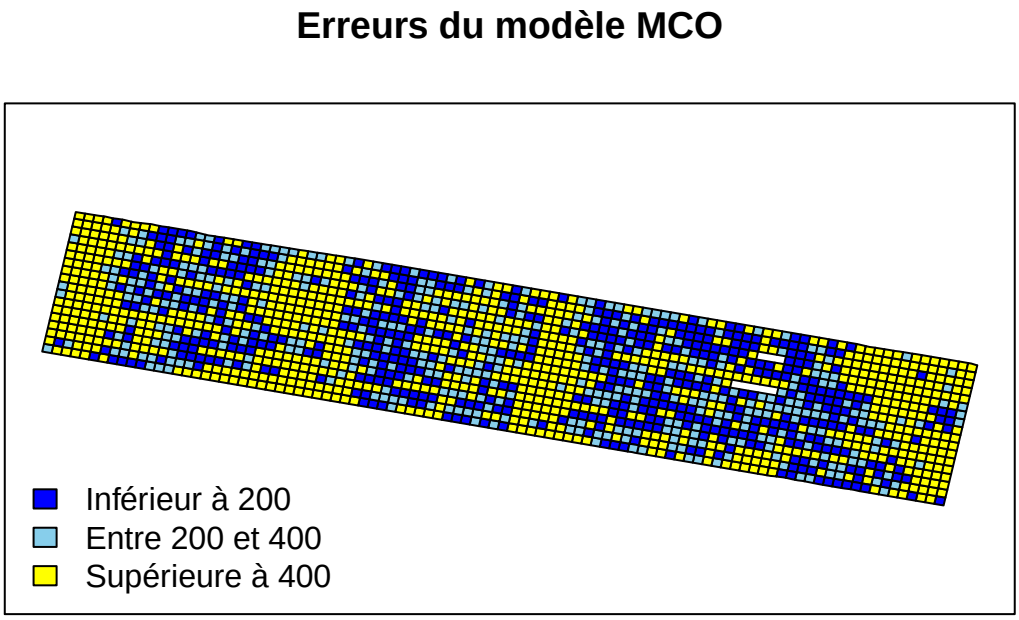
On a déjà trouvé que le modèle SAR est mieux que le modèle obtenu par MCO selon le critère AIC. Maintenant, on va faire une étude sur les résidus de chacun de ces modèles.

```
quadrant1 <- vector(mode="numeric",length=length(res.sar))
quadrant2 <- vector(mode="numeric",length=length(res.lm))

quadrant1[abs(res.sar)<200] <- 0
quadrant1[abs(res.sar)>=200 & abs(res.sar)<400] <- 1
quadrant1[abs(res.sar)>=400] <- 2

quadrant2[abs(res.lm)<200] <- 0
quadrant2[abs(res.lm)>=200 & abs(res.lm)<400] <- 1
quadrant2[abs(res.lm)>=400] <- 2

brks <- c(0,1,2)
colors = c("blue","skyblue","yellow")
plot(yields,col=colors[findInterval(quadrant1,brks,all.inside=FALSE)],
     main="Erreurs du modèle SAR")
box()
legend("bottomleft", legend = c("Inférieur à 200","Entre 200 et 400","Supérieure à 400"),
       fill=colors,bty="n")
```




```
plot(yields,col=colors[findInterval(quadrant2,brks,all.inside=FALSE)],
     main="Erreurs du modèle MCO")
```

```
box()
legend("bottomleft", legend = c("Inférieur à 200","Entre 200 et 400","Supérieure à 400"),
       fill=colors,bty="n")
```

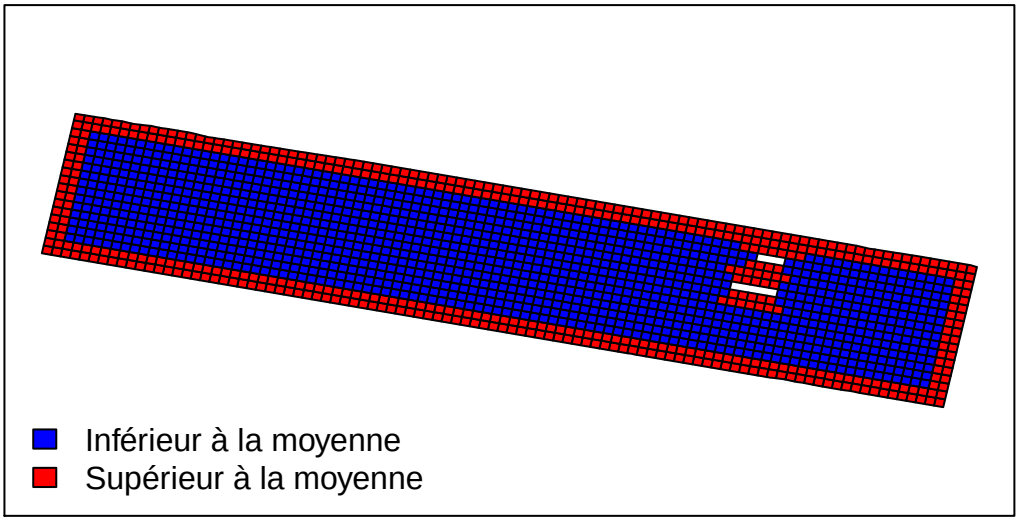


Par le premier modèle de regression linéaire classique on a obtenu des erreurs plus grandes avec une forte autocorrélation positive, elles forment des clusters, comme montré ci-dessus dans le graphique des erreurs du modèle MCO. En revanche, les erreurs du modèle SAR sont pas très grandes avec une faible autocorrélation positive. Donc le modèle SAR est mieux que le modèle obtenu par MCO.

Les parcelles pour lesquelles l'effet direct est maximal et la valeur de l'effet

```
quadrant <- vector(mode="numeric",length=length(direct))

quadrant[direct<mean(direct)] <- 0
quadrant[direct>mean(direct)] <- 1

brks <- c(0,1)
colors = c("blue","red")
plot(yields,col=colors[findInterval(quadrant,brks,all.inside=FALSE)],main="Les parcelles selon la valeu
box()
legend("bottomleft", legend = c("Inférieur à la moyenne","Supérieur à la moyenne"), fill=colors,bty="n"
```

Les parcelles selon la valeur de l'effet direct pour modèle SAR



On remarque que les parcelles qui ont un effet direct maximal (plus grand que la moyenne 12.62) se trouvent dans les frontières du champ.

Comparaison des modèles

Dans cette partie on va comparer le modèle SEM avec autocorrélation spatiale des erreurs, et le modèle SAR avec variable endogène spatialement décalée.
On a le modèle SEM s'écrit sous la forme suivante :

$$YIELD100 = \beta_0 + \beta_1 * N + \beta_2 * N_2 + \beta_4 * TOP_3 + \beta_5 * TOP_4 + \nu$$

Avec

$$\nu = \lambda W \nu + \varepsilon$$

```
YIELD100.errorsarlm <- errorsarlm(YIELD100~N+N2+TOP3+TOP4,data=data,poids.vois)
summary(YIELD100.errorsarlm)
```

```
##
## Call:errorsarlm(formula = YIELD100 ~ N + N2 + TOP3 + TOP4, data = data,
##     listw = poids.vois)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2095.12  -228.99    -8.05   207.80  2713.21
##
## Type: error
## Coefficients: (asymptotic standard errors)
##                Estimate  Std. Error z value  Pr(>|z|)
## (Intercept)  6.1041e+03  8.5869e+01 71.0860 < 2.2e-16
## N            1.0817e+01  6.4445e-01 16.7855 < 2.2e-16
## N2          -2.2216e-02  4.6094e-03 -4.8197 1.438e-06
## TOP3        -6.1810e+02  1.0816e+02 -5.7146 1.099e-08
## TOP4        -2.4268e+02  1.2209e+02 -1.9877   0.04684
##
```

```
## Lambda: 0.86111, LR test value: 1365.5, p-value: < 2.22e-16
## Asymptotic standard error: 0.0151
##     z-value: 57.028, p-value: < 2.22e-16
## Wald statistic: 3252.2, p-value: < 2.22e-16
##
## Log likelihood: -12902.47 for error model
## ML residual variance (sigma squared): 139920, (sigma: 374.06)
## Number of observations: 1738
## Number of parameters estimated: 7
## AIC: 25819, (AIC for lm: 27182)
```

```
res1 <- residuals(YIELD100.errorsarlm)
moran.test(res1,poids.vois) ## les résidus ne sont auto-corrélés
```

```
##
##  Moran I test under randomisation
##
## data:  res1
## weights: poids.vois
##
## Moran I statistic standard deviate = -9.6624, p-value = 1
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##     -0.1203655606     -0.0005757052      0.0001536972
```

```
### comparaison des AIC
print(AIC(YIELD100.errorsarlm,YIELD100.lagsarlm))
```

```
##                      df      AIC
## YIELD100.errorsarlm  7 25818.94
## YIELD100.lagsarlm    7 26093.07
```

Mˆeme si le modèle SEM est mieux que le modèle SAR selon le critère AIC, on trouve que la statistique du test de Moran pour le modèle SEM est de -0.12 plus grand que la statistique du test du modèle SAR en valeur absolue, donc il reste une grande autocorrélation spatiale dans les résidus du modèle SEM. Pour cette raison on garde le modèle SAR.

Conclusion :

Dans cette étude d'autocorrélation spatiale, on a construit un modèle de regression linèaire classique par MCO, mais on a trouvé une autocorrélation spatiale forte dans les résidus du modèle. Après on s'est interéssé à chercher o'u se trouve l'autocorrélation est ce que dans la variable indépendante ou dans les variables explicatives. Pour cela, on a utlisé deux modèles SAR et SLX, le test de Moran nous a indiqué que pour l'autocorrélation spatiale dans les résidus du modèle SLX est forte 0.62, et pour le modèle SAR on a trouvé 0.02 proche de 0. Alors, on a gardé ce dernier modèle pour modéliser l'autocorrélation spatiale. En comparant le modèle SAR avec le modèle SEM et en se basant sur le test de Moran on a trouvé que le modèle SAR est le plus pertinent.