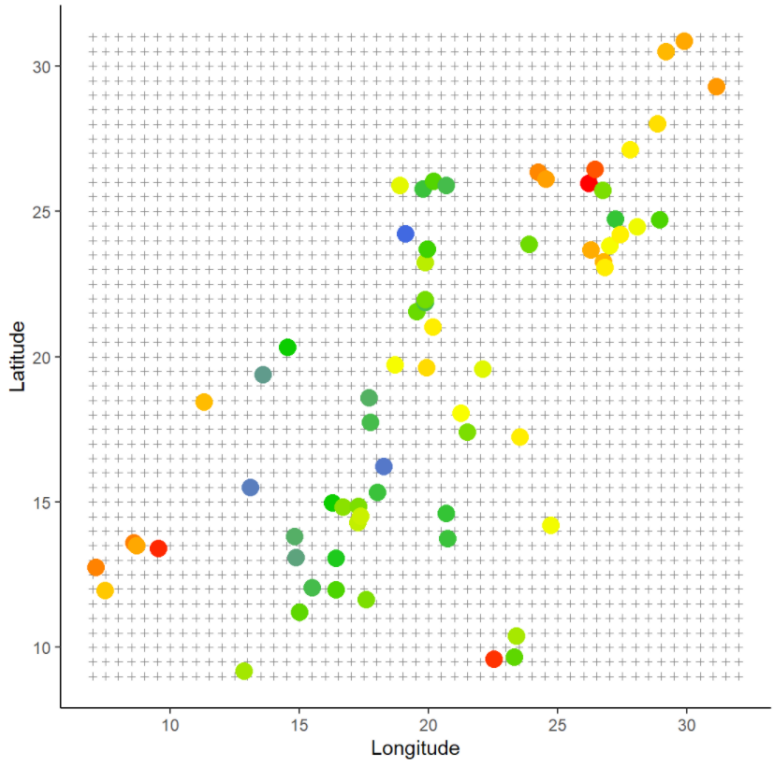
LBRTI2101A
DATA SCIENCE IN BIOSCIENCE ENGINEERING- PARTIM A

Projet d'analyse de données spatiales



HEROUFOSSE Gauthier
JUSHPE Louna
LAFFINEUR Brieuc
RIGO Maxime

Professeurs :
BOGAERT Patrick,
TOUSSAINT François

Année académique 2021 - 2022

Table des matières

1 Introduction

Ce travail a pour objectif de traiter les données tirées de la cartographie des concentrations en éléments traces métalliques (ETM) en Wallonie.

La notion d'éléments traces métalliques est de plus en plus utilisées pour remplacer le concept de métaux lourds. Ce sont des polluants qui se retrouvent aujourd'hui fréquemment dans les sols wallons, à cause des secteurs de la métallurgie, du textile, des intrants agricoles ou encore à cause du passé de la Wallonie dans les activités industrielles d'activités minières (Naert, 2016).
Ces polluants s'accumulent dans le sol et cette acumulation représente des risques environnementaux comme la perte de services écosystémiques liés au sol (dégradation de la matière organique, filtration de l'eau...) ou des dangers liés à la santé humaine car certains d'entre-eux sont toxiques à très faibles concentrations (Bliefert Perraud, 2011).

Nos données, récoltées par plusieurs laboratoires wallons et homogénéisées par un laboratoire de l'UCLouvain, nous donnent les concentrations de nickel (Ni), de zinc (Zn) et de chrome (Cr) en fonction de la position géographique (X et Y en Lambert 1972), ainsi que le sigle pédologique du point mesuré et la région concernée. Les concentrations ainsi que les prédictions que nous réaliserons seront toutes exprimées en mg/kg de sol sec.

Au cours de ce rapport, ces données de concentrations en seront tout d'abord affichées telles quelles, sans modification du jeu de données afin d'avoir un premier aperçu de la région concernée par celui-ci. Cette analyse exploratoire permettra à la fois de déterminer la zone géographique qui nous intéresse mais aussi de faire ressortir les principales statistiques des données (distribution, variance, covariance...).
Bien sûr, sur base des statistiques précédentes, il sera possible de déceler les éventuels valeurs abberantes et ainsi de corriger tout cela pour obtenir un jeu de données plus cohérent.
L'étape suivante consistera en la modélisation de la dépendance spatiale de nos données via les variogrammes les plus pertinents par rapport à nos concentrations de Ni, Zn et Cr.
Ensuite, il sera très intéressant de faire plusieurs prédictions statistiques des ETM sur la province en question. Ces prédictions seront faites sur base de différentes méthodes (IDW, krigeage, co-krigeage), ce qui permettra d'en évaluer la qualité et de discuter les différences de celles-ci.
Enfin, une carte des risques de dépassement d'un seuil pour un des trois éléments traces métalliques pourra s'avérer très utile afin d'identifier les zones spatiales de la province qui sont le plus à risque. Cette carte servira par exemple à mettre en place des actions de terrain plus concrètes pour préserver ces zones d'une pollution trop importante.

2 Analyse exploratoire des données

2.1 Position des points de mesure des ETM

Les 3 figures ci-dessous représentent la position des points de mesure des ETM dans la province du Hainaut. Les points non-colorés représentent les NAN [1].

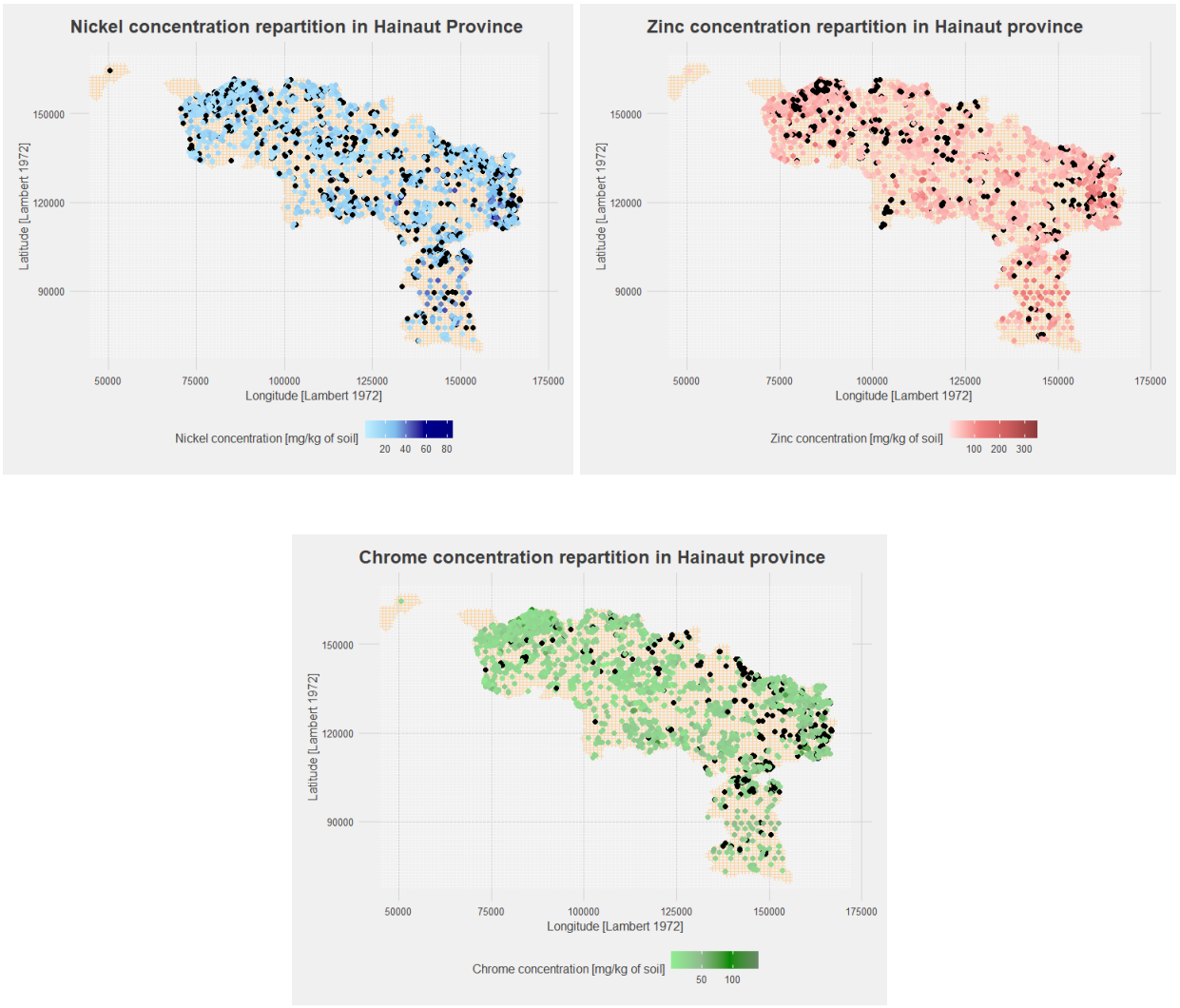


FIGURE 1 – Position des points de mesure des différents ETM

Sur base de la figure ci-dessus, il est aisé de constater que les mesures n'ont pas toutes été prises de la même manière à chaque position.
En effet, les points noirs, auxquels correspondent des valeurs non-chiffrées, sont localisés différemment. Cela signifie que, pour tous ces points en question, un (ou 2) des 3 éléments trace métalliques en question dans ce projet n'a pas été analysé.
Aussi, on remarque que ces points noirs sont présents en plus grand nombre dans les analyses de nickel que dans les analyses de zinc, qui présentent elles-même davantage de valeurs non-chiffrées que dans les analyses de chrome.

1. Not A Number : Les points qui n'ont été mesuré que pour un ou 2 éléments traces. La localisation du 3e élément n'affiche donc pas de valeur

Il faudra donc être vigilant au moment de supprimer les NaN, de manière à ne pas supprimer toutes les positions de mesure contenant ne fut-ce qu'une seule valeur non-chiffrée, et ainsi perdre un grand nombre de données chiffrées associées à ces positions.

En ce qui concerne les valeurs de concentration en ETM, mesurées en milligrammes par kilogrammes de sol, quelques échelles peuvent être mises en évidences en un coup d'oeil :

- Les concentrations en Ni sont majoritairement comprises entre 0 et 85 mg/kg.
- Les concentrations en Zn sont principalement comprises entre 0 et 350 mg/kg.
- Les concentrations en Cr sont en grande partie comprise entre 0 et 150 mg/kg.

Bien sûr, tout ceci n'est utile qu'à titre indicatif et sera bien plus développé et analysé dans la suite de ce rapport.

2.2 Province étudiée

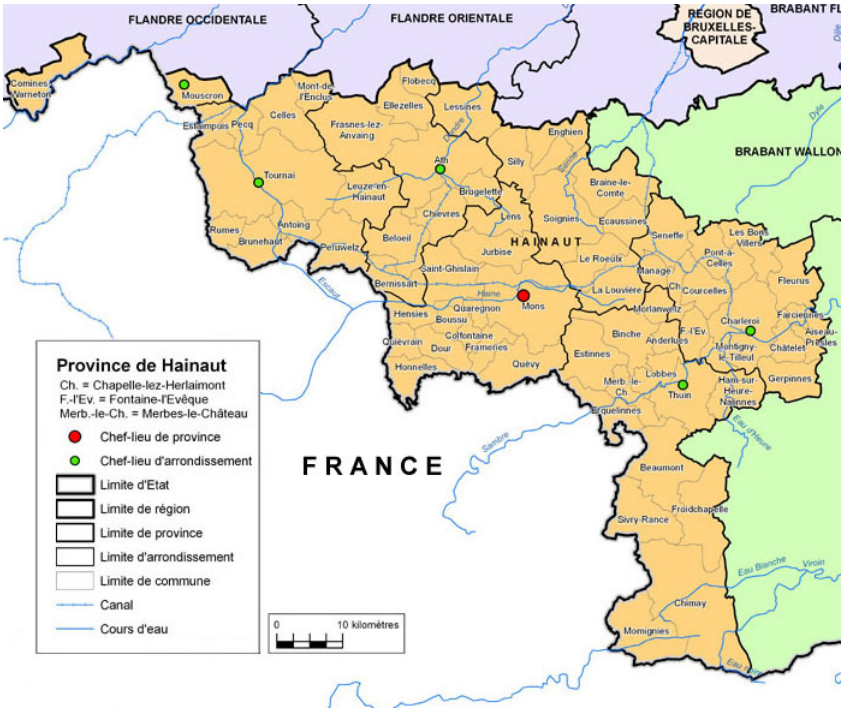


FIGURE 2 – Province du Hainaut

Sur base de la distribution des données (figure 1), il est possible de définir les limites de la province étudiée. La province du Hainaut est ainsi aisément distinguable sur base de cette répartition spatiale. Les limites spatiale de ce rapport seront donc déterminées par cette province.
A noter que la petite partie de province esseulée à l'Ouest à côté de la Flandre occidentale sera bien prise en compte dans ce travail. En effet, en comparant la figure 2 avec la figure 1, on constate que les mesures de concentrations en ETM ont été réalisées aussi bien sur cette zone détachée que sur le reste de la province. Il serait donc incohérent de ne pas prendre cette zone en considération dans notre jeu de données.

2.3 Histogrammes et statistiques des ETM

Les principales statistiques des ETM sont obtenues grâce à la commande "summary ()" sur *R Studio* une fois le jeu de données chargé. Bien sûr, on parle dans cette section du jeu de données brutes, qui ne sera corrigé qu'à partir de la prochaine section.
Ainsi, les concentration de nickel s'étendent de 1,00 à 85,37 mg/kg de sol. La médiane et la moyenne de concentrations en Ni sont respectivement de 15,16 et 16,54 mg/kg. Le Ni contient 821 NaNs sur 3801 mesures.
Les concentrations de zinc s'étendent de 2,00 à 350,40 mg/kg de sol. La médiane et la moyenne de concentrations en Zn sont respectivement de 54,92 et 59,22 mg/kg. Le Zn contient 514 NaNs sur 3801 mesures.
Les concentrations de chrome s'étendent de 2,00 à 141,89 mg/kg de sol. La médiane et la moyenne de concentrations en Cr sont respectivement de 25,74 et 27,00 mg/kg. Le Cr contient 475 NaNs sur 3801 mesures.

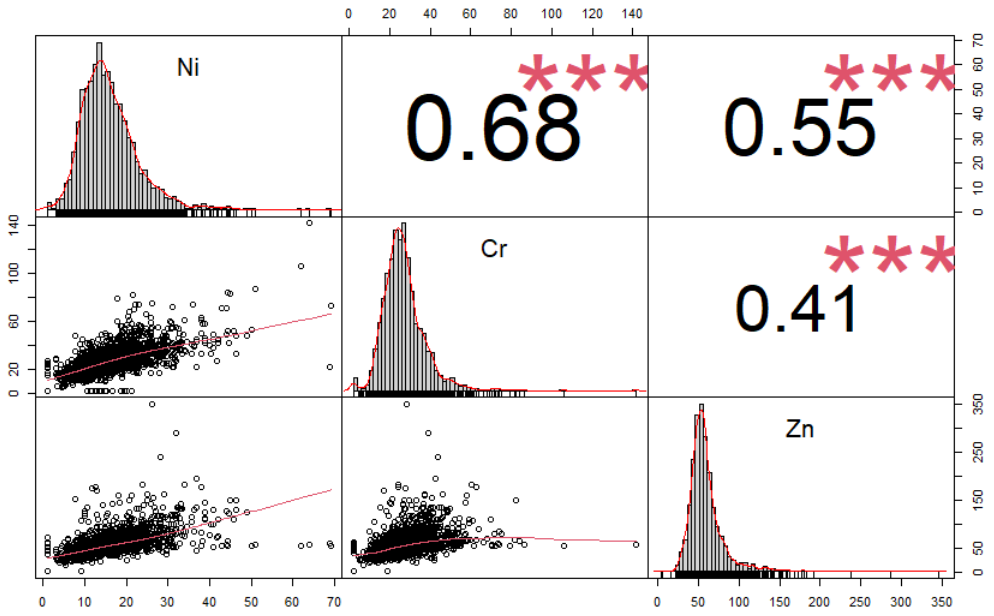


FIGURE 3 – Scatter plots, Histogrammes et covariance des ETM après retrait des points qui avaient au moin une valeur NAN

En plus des histogrammes qui illustrent le nombre de données en fonction des concentration, la figure ci-dessus permet surtout de visualiser la corrélation qui existe entre les trois variables continues étudiées.
Les diagrammes de dispersion ou scatterplots montrent rapidement la corrélation qui existe entre les 3 ETM, pris 2 par 2. Ainsi, on voit que c'est le nickel et le chrome qui possèdent la plus grande corrélation spatiale (corrélation de 0,68), le maximum possible étant une valeur de 1. On constate donc par ailleurs que le Cr et le Zn ont la plus faible corrélation spatiale (0,41) et que le zinc et le nicke ont une corrélation de 0,55.

3 Ajustement des données

Le jeu de données contient beaucoup de valeurs vides (des NAN). Sur les 3801 points de données de départ, 1433 points manquent au moins 1 des 3 concentrations des ETM. En retirant ces valeurs non chiffrées grâce à la fonction *na.omit*, la base de données passe ainsi à 2368 points de mesure.

Cette opération n'est pas la plus pertinente car elle retire toutes les concentrations en ETM d'une ligne, même si 2 concentrations sur 3 n'étaient pas des NaN.
Pour contourner cette perte de données excessive, il a fallu travailler différemment pour trier ces données, en sélectionnant chaque fois le *na.omit* sur une seule colonne de concentration en ETM et non pas sur les trois.
En parallèle de cette opération, il sera intéressant de retirer les valeurs aberrantes du jeu de données. Cette opération sera décrite par la suite dans la section 3.2.

D'autre part, vu que le jeu de données comprenait également des valeurs de concentration d'un même élément mesurées plusieurs fois au même endroit, il a aussi été nécessaire de régler ce détail. En effet, étant donné l'utilisation de certaines méthodes de prédiction exacte, l'existence de plusieurs valeurs à une même position aurait grandement faussé les estimations. C'est pourquoi, via la commande *group by*, tous les doublons de localisation ont pu être supprimés et remplacés par une seule valeur moyenne de la concentration en ETM de ces doublons. Une fois cette opération réalisée, ce sont 457 données doublons qui ont été éliminées du jeu de données.

3.1 Distribution des concentrations

La figure ci-dessous intègre les histogrammes des 3 ETM étudiés. Sur base de cette distribution des concentrations, il sera possible de voir si celles-ci sont normales ou s'il faut leur appliquer une transformation.

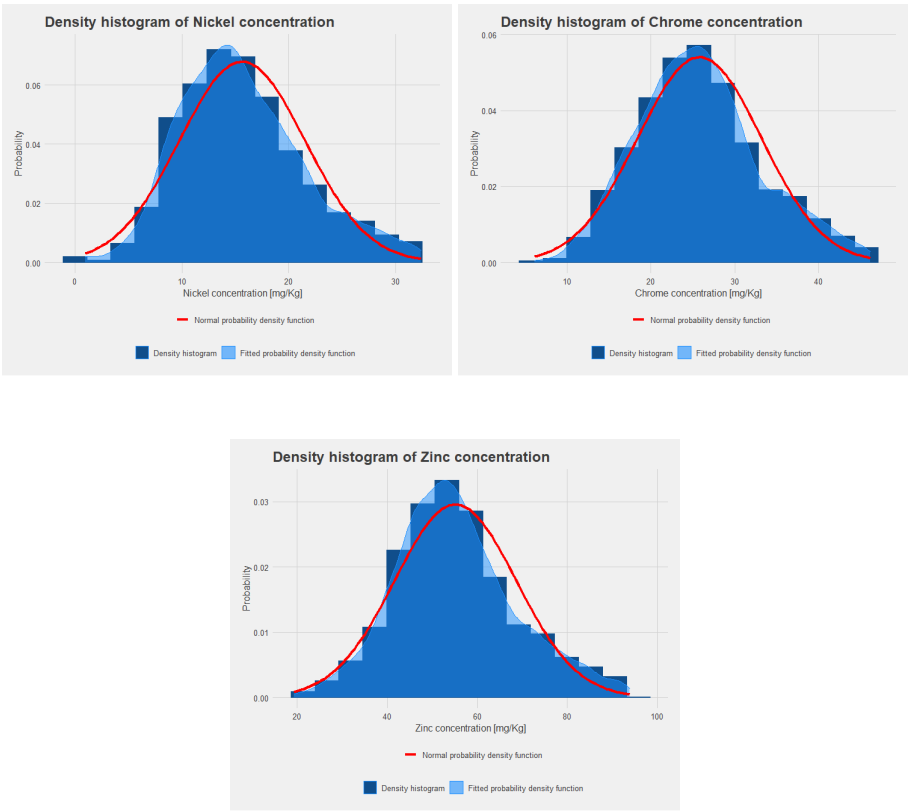


Figure 4 – Distribution des concentrations des différents ETM

On constate sur base de la figure 4 que les concentrations des 3 ETM suivent toutes une loi normale de distribution. En effet, la ligne bleue (associée à l'histogramme) suit de très près la ligne rouge qui correspond à la fonction de densité normale. Il n'y aura donc pas lieu d'appliquer une transformation (*box cox* par exemple) afin de rendre la distribution des données normale. A noter que la distribution des concentrations du Cr est la plus fidèle à la loi normale, comparée aux deux autres ETM. D'autre part, d'autres analyses statistiques telles que qqplot, présentes en annexe (figures 23 et 24), ont permis de confirmer le premier constat de la distribution normale.

3.2 Valeurs aberrantes (outliers)

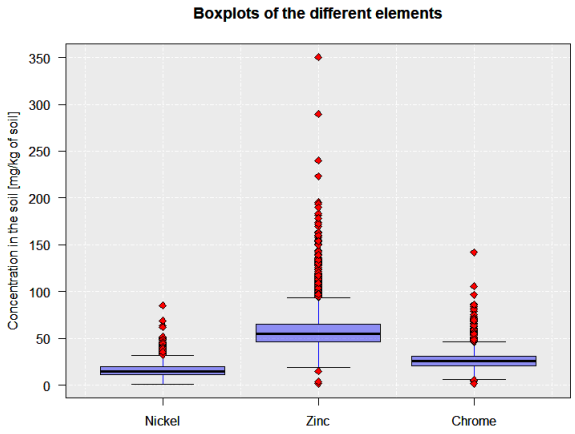


FIGURE 5 – Box plot des ETM

La figure ci-dessus comprend les boxplots des 3 ETM étudiés. Ces boxplots illustrent la médiane, le premier quartile, le troisième quartile, le minimum, le maximum et les outliers ou valeurs aberrantes [2] de chaque échantillon de données. Il est aisé de constater que les 3 ETM comprennent des outliers . En particulier, le zinc présente un plus grand nombre d'outliers et ceux-ci ont des valeurs très élevées comparées aux valeurs des autres ETM.

Les valeurs aberrantes sont problématiques car elles peuvent affecter les résultats d'une analyse. Il semble donc plus que nécessaire de corriger les données en retirant celles-ci pour chaque ETM.
Pour ce faire, une façon courante de trouver des valeurs aberrantes dans unjeu de données est d'utiliser l'écart interquartile.
L'écart interquartile, souvent abrégé en IQR (Inter Quartile Range), est la différence entre le premier quartile (Q1) et le troisième quartile (Q3) d'un ensemble de données. Il mesure la dispersion de la moitié des valeurs.
La méthode consiste à déclarer qu'une observation est aberrante si elle a une valeur 1,5 fois supérieure à l'IQR ou 1,5 fois inférieure à l'IQR (voir figure ci-dessous).

2. Une valeur aberrante est une observation qui se situe anormalement loin des autres valeurs d'un ensemble de données.

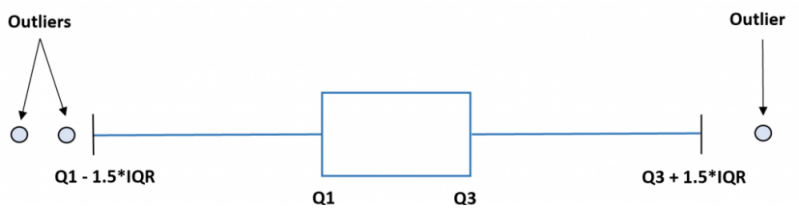


FIGURE 6 – Illustration de la méthode de calcul des valeurs aberrantes[1]

Cette méthode permet de corriger au mieux le jeu de données pour ensuite refaire les box plots pour chaque ETM et avoir ainsi une meilleure vision, non biaisée de leur distribution (voir figure 7)

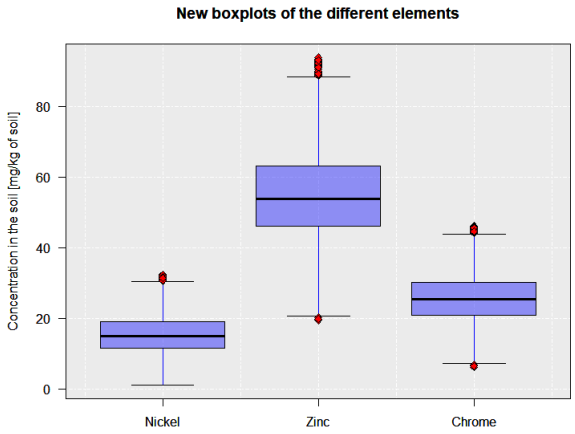


FIGURE 7 – Box plot des ETM après retrait des valeurs aberrantes

La figure ci-dessus renseigne du fait que les données sont désormais bien mieux réparties pour chaque ETM, en témoigne l'échelle de concentration qui est passée d'une valeur maximale de 350 mg/kg à une valeur maximale de 100 mg/kg.

Néanmoins, il semble pertinent de se demander pourquoi des outliers existent encore en dehors des valeurs limites définies une fois la méthode de l'IQR appliquée. Cela vient du choix de prendre une méthode moins restrictive dans l'affichage de nos données.
Ce choix s'explique tout simplement par la nature intrinséque des ETM. Il s'avère que ceux-ci peuvent être de nature anthropique, c'est-à-dire qu'ils sont parfois présents dans le sol à cause de la pollution liée aux activités humaines comme la métallurgie ou l'agriculture. Il n'est donc pas impossible de trouver des valeurs de concentrations en ETM beaucoup plus élevées que la moyenne dans des zones de prélèvement proches d'activités humaines de ce genre.
Ainsi, la méthode IQR à été préférée à celle des distances de 3 écarts-types à la moyenne, plus restrictive dans notre cas. La visualisation que vous observez est celle qui correspond à l'utilisation des écarts-types, ce qui explique que des valeurs extérieures aux quartiles subsistent.

4 Analyse de la dépendance spatiale

4.1 Hypothèses et variogrammes initiaux

Pour analyser de la dépendance spatiale du jeu de données, il faut procéder à l'élaboration des semi-variogrammes des différents ETM. Le semi-variogramme est un outil très utile en analyse de dépendance spatiale car il rend compte de la dissimilitude entre deux valeurs en fonction de la distance. Normalement, on s'attend à ce que cette fonction soit monotone croissante avec la distance.

Plusieurs hypothèses doivent cependant être émises au préalable. Tout d'abord, il faut faire l'hypothèse que la stationnarité d'ordre 1 est respectée, c'est-à-dire que l'espérance du champ de données ne dépend pas de la position, est la même partout et est supposée inconnue.
Ensuite, on assume la stationnarité intrinsèque du champ aléatoire, ce qui implique que la fonction $\gamma(h)$ ne dépend que de la distance h, donc que la variance de l'incrément $Z(x+h) - Z(x)$ ne dépend que de la distance entre les deux valeurs et vaut $2\gamma(h)$. Il faut poser cette hypothèse car elle est nécessaire pour définir le variogramme.
Enfin, nos variogrammes sont considérés comme isotropiques, c'est-à-dire que que ceux-ci ne dépendent que de $||h||$ et uniquement de $||h||$, et non de la direction du vecteur h. Il faut poser cette hypothèse car il n'y a aucune information quant à une éventuelle anisotropie.

Les hypothèses étant maintenant posées, il est possible dans un premier temps de présenter les premiers semi-variogrammes obtenus pour chaque ETM ainsi que les paramètres optimisés des modèles appliqués. Les données utilisées sont celles corrigées dans la section précédentes et le résultat permettra de déterminer si d'autres manipulations doivent être appliquées sur celles-ci.

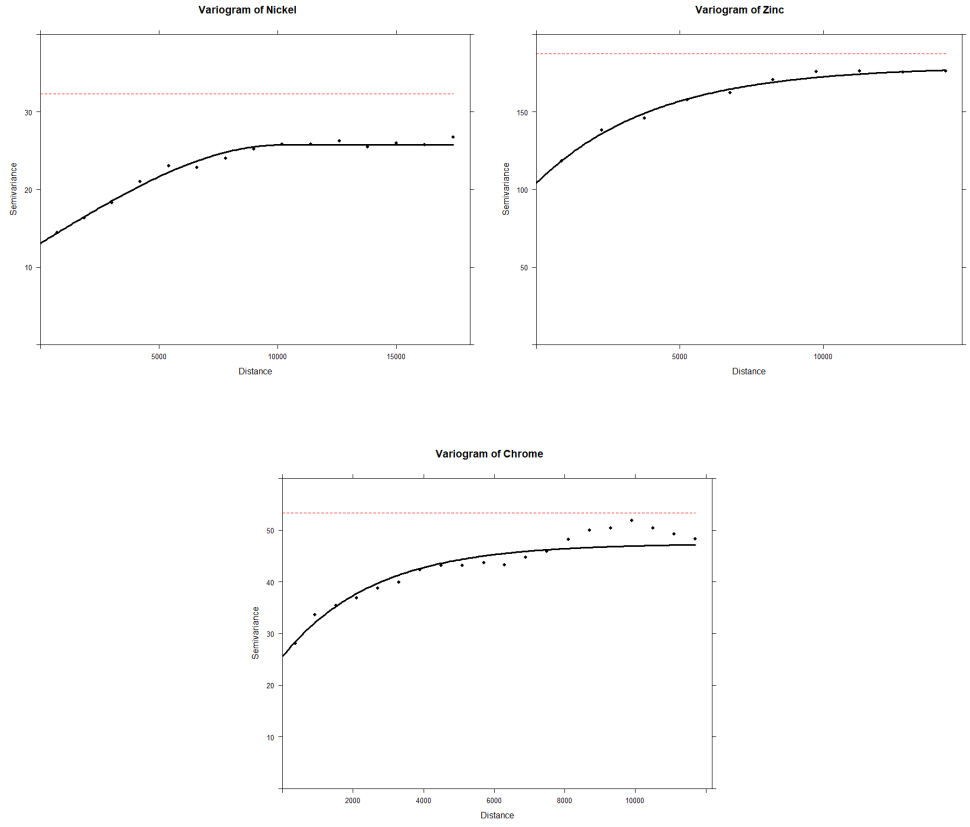


FIGURE 8 – Variogrammes des trois différents ETM

```
> Ni.vg.fit
  model    psill    range
1   Nug 13.07779     0.00
2   Sph 12.67709 10188.31
> Zn.vg.fit
  model     psill    range
1   Nug 103.95720    0.000
2   Exp  75.31396 4111.516
> Cr.vg.fit
  model    psill    range
1   Nug 25.50404    0.000
2   Exp 21.88083 2575.286
```

FIGURE 9 – Paramètres optimisés pour les premiers semi-variogrammes

Le semi-variogramme du nickel est obtenu avec un modèle de fitting de type sphérique. Ceux des deux autres éléments ont été obtenus avec un modèle exponentiel. Ces modèles ont été déterminés comme étant les plus adaptés par R lors du fitting.

Tout d'abord, il est intéressant de noter la présence d'un effet de pépite sur chaque semi-variogramme, leurs valeurs étant reprises dans la figure 7. Ceci indique que même deux valeurs proches n'ont pas une très forte ressemblance.
La figure 8 illustre le fait que pour l'instant, aucun de ces semi-variogrammes ne tend vers la valeur de variance attendue. On peut donc supposer que les différents champs ne respectent pas la stationnarité d'ordre 1. Par conséquent, il est nécessaire de procéder au retrait de l'influence de la moyenne. La visualisation de ces données est présentée dans la section suivante.

4.2 Visualisation des champs de données

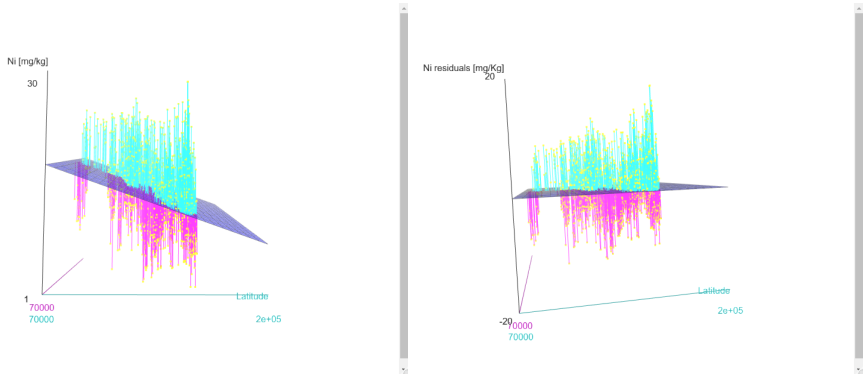


FIGURE 10 – Visualisation des données du Nickel avant et après le retrait de l'influence de la moyenne

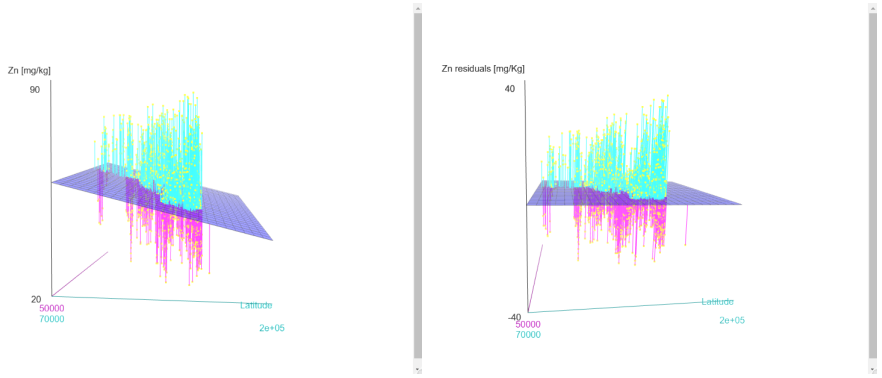


FIGURE 11 – Visualisation des données du Zinc avant et après le retrait de l'influence de la moyenne

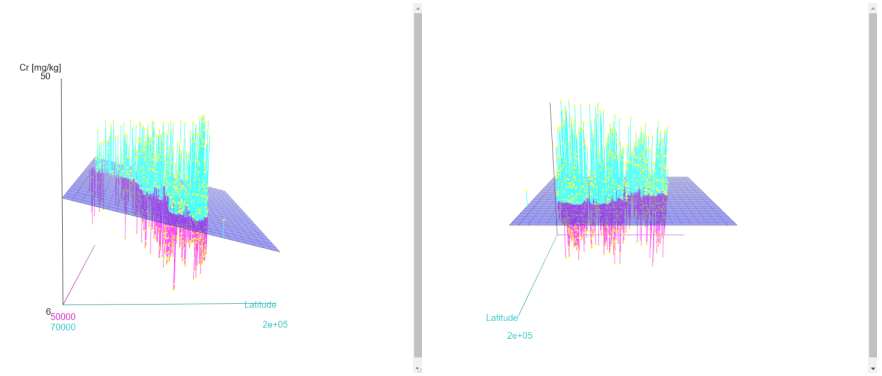


FIGURE 12 – Visualisation des données du Chrome avant et après le retrait de l'influence de la moyenne

L'allure des champs avant le retrait de l'influence de la moyenne semble confirmer qu'ils ne respectaient pas la stationnarité d'ordre 1 mais que la correction qui leur est appliquée rétablit celle-ci. Après inspection des différentes visualisations, aucune valeur anormalement faible ou élevée ne ressort. Les semi-variogrammes des résidus peuvent donc être produits et sont présentés dans la section suivante avec les différents paramètres optimisés.

4.3 Variogrammes des résidus

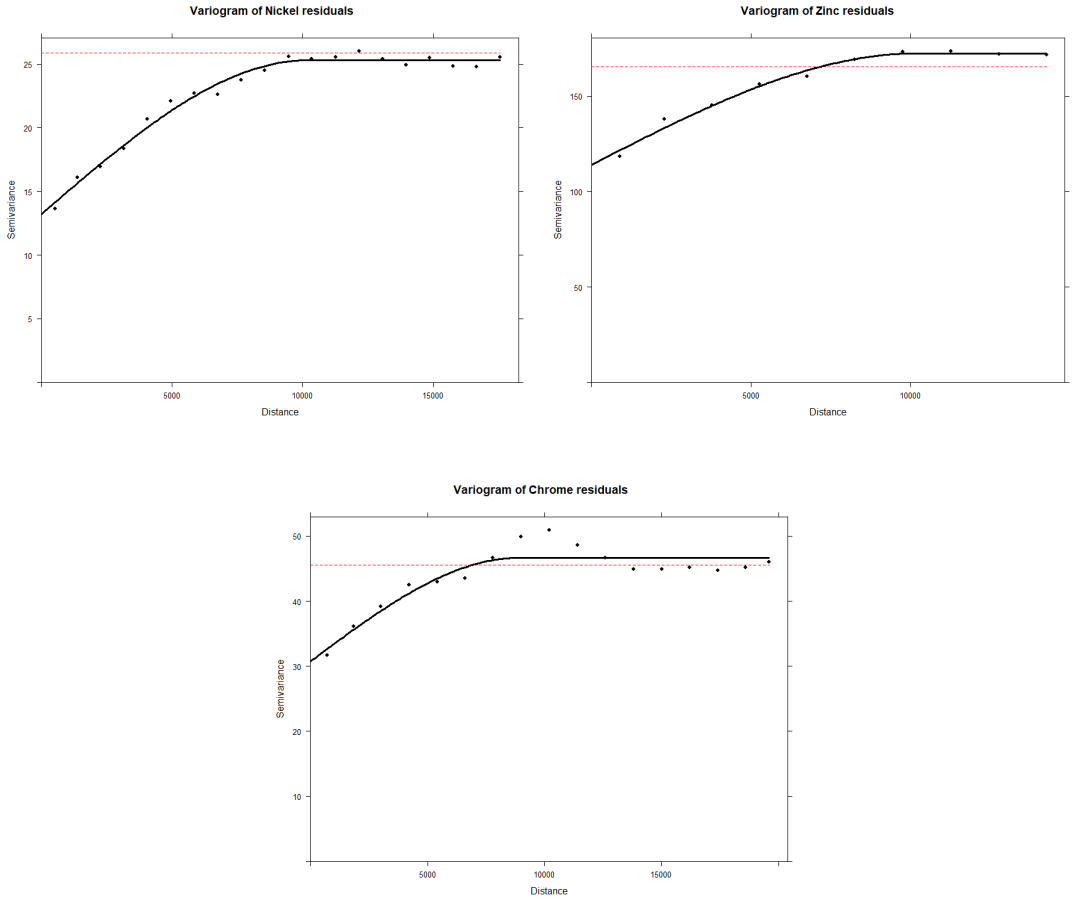


FIGURE 13 – Variogrammes des résidus des trois différents ETM

```
> clean.fit.ni.exp
  model    psill    range
1   Nug 13.21445     0.00
2   Sph 12.10030 10223.94
> clean.fit.zn.exp
  model     psill    range
1   Nug 114.12678     0.00
2   Sph  58.30283 10219.71
> clean.fit.cr.exp
  model    psill    range
1   Nug 30.74043    0.000
2   Sph 15.92876 8905.193
```

FIGURE 14 – Paramètres optimisés pour les modèles utilisés pour les variogrammes des résidus

Il est intéressant de constater que tous les modèles sont désormais sphériques pour procéder au fitting, tandis que les premiers semi-variogrammes du chrome et du zinc avant le retrait de l'influence de la moyenne avaient été obtenus avec des modèles exponentiels.
Les trois semi-variogrammes des différents ETM tendent désormais vers leur variance respective.

On note également que les valeurs de range sont assez proches pour les trois éléments, le chrome étant un peu plus bas. Puisque la range indique la distance à laquelle le modèle commence à s'aplatir, et donc le moment où la dissimilitude entre les valeurs commence à atteindre son maximum, on peut établir que nos trois ETM réagissent de manière assez similaire avec une augmentation de la distance.
On remarque que cette range n'a que très peu changé entre le premier et le second semi-variogramme dans le cas du nickel. Cependant, celle-ci a fortement changé pour le zinc et le chrome, on peut faire l'hypothèse que ceci est surtout dû au changement de modèle par R. Un modèle sphérique a été préféré dans les trois cas pour les seconds semi-variogrammes.

A propos de l'effet de pépite, on remarque qu'il est présent dans les trois semi-variogrammes. Au plus celui-ci est élevé, au moins deux valeurs voisines seront corrélées. Dans le cas des données d'ETM, cet effet semble plutôt prononcé.
Plusieurs explications peuvent être à l'origine de cet effet de pépite. La première serait que les données n'ont pas toutes été prises au même moment. Ceci impliquerait qu'entre deux campagnes d'échantillonage, les valeurs de deux mesures très proches aient changé.
La seconde explication serait la variabilité de l'instrument de mesure. La prise de mesure est toujours accompagnée d'incertitude, celle-ci étant dûe à l'instrument en lui-même, à la personne prenant les mesures, ou à l'utilisation de plusieurs instruments.
Enfin, la dernière hypothèse est simplement un réel effet de pépite, c'est-à-dire une variation marquée du paramètre mesuré. Ceci pourrait être dû au rejet industriel de ces éléments dans la nature et impliquerait une concentration plus forte au voisinnage de concentrations naturelles dans le sol. D'autre part, un lien peut être avec la section 3.2 où les outliers proches des limites de la méthode de l'IQR ont été conservé.

5 Présentation des résultats

Cette partie s'intéresse à la prédiction des valeurs de concentration des ETM sur l'ensemble du territoire étudié. Pour ce faire, il faut effectuer des interpolations avec différentes méthodes de prédictions. L'ensemble des méthodes que présentées ci-dessous donnent lieu à des prédicteurs linéaires. Cette section a pour but de présenter l'implémentation des différentes méthodes suivie des résultats, pour finir avec une discussion autour de ces résultats obtenus dans la section suivante.

5.1 Méthode déterministe : IDW

La première méthode utilisée est une méthode déterministe nommée prédiction par distance inverse. La méthode suppose que tout point x_i possède un poids proportionnel à l'inverse de la distance h_i entre ce point et x_0. De plus, un paramètre θ détermine la vitesse à laquelle le poids d'un point diminue en fonction de $||h||$ [2]. L'enjeu sera de trouver la valeur optimale que représente θ pour chaque élément trace métallique par cross-validation (*"leave-one-out cross-validation"*) en minisant la somme des carrés des écarts entre la prédiction en un point x_i et la valeur mesurée en ce point. Une telle méthode d'optimisation est nécessaire dans le cas où le prédicteur est exact, ce qui est le cas de la méthode de distance inverse.

Après implémentation de la méthode, il faut faire la recherche de ce paramètre θ optimal dans un intervalle compris entre 0.5 et 3.5, par incrément de 0.25. Ces calculs donne des valeurs de θ égales à **1.25** dans le cas des trois ETM. Un exemple d'optimisation du paramètre est disponible en Annexe.

Dû à l'agglomération des points de mesure, la décision a été prise d'utiliser un nombre maximal de points dans le voisinage pour la prédiction par distance inverse égal à **20**. Ainsi, cela permet

de se concentrer sur une prédiction à un niveau local plutôt qu'un niveau global. Dû à une forte hétérogénéité, nous n'avons pas établi de distance maximale dans laquelle les points devraient être piochés .

Avec cette paramétrisation, voici les différents résultats obtenus :

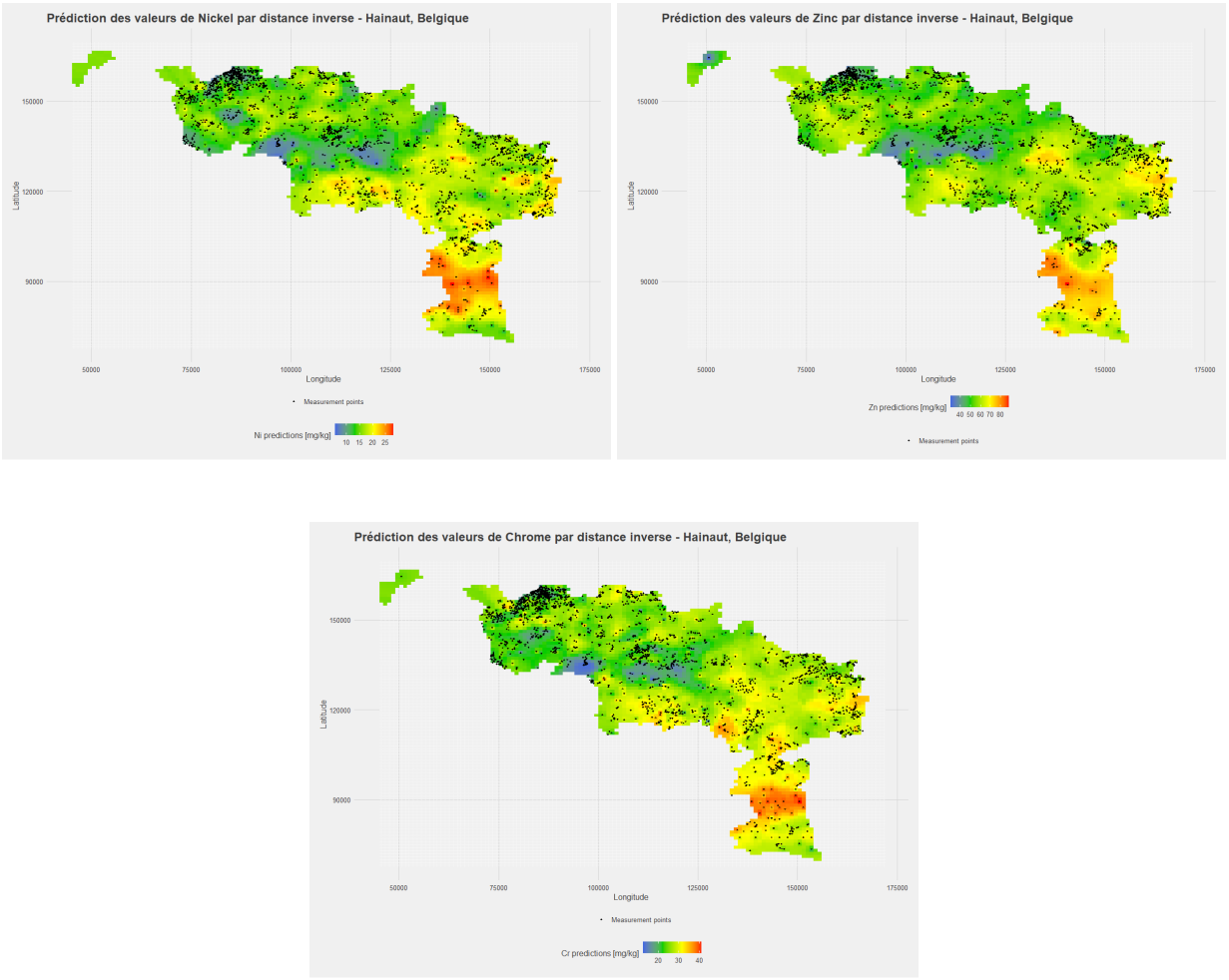


FIGURE 15 – Prédictions par distance inverse des 3 ETM

5.2 Méthode stochastique : Krigeage

Ensuite, une seconde prédiction sur les concentrations en ETM dans la zone d'étude a été réalisée, mais cette fois sur base d'un modèle stochastique.

Le krigeage est une méthode de prédiction linéaire réalisant l'interpolation spatiale de notre variable en utilisant l'interprétation d'un variogramme expérimental. À la différence d'un modèle de régression locale, le krigeage suppose que les erreurs sont maintenant dépendantes spatialement. Il s'énonce comme suit :

$$Z(s) = \mu(s) + \delta(s), \quad s \in D$$

où $\mu(s)$ est la structure déterministe du modèle, et $\delta(s)$ une fonction aléatoire stationnaire, d'espérance nulle et de structure de dépendance connue [3].

Ce type de modèle offre la prédiction la plus optimale qui soit : l'espérance de son erreur est nulle et la variance de cette erreur est la plus petite possible. L'hypothèse posée dans ce cas était une

stationnarité d'ordre 1 des données. Il faut donc considérer la moyenne $\mu(s)$ comme étant constante et inconnue. Par conséquent, le kriegage effectué est de type ordinaire. Concernant la structure de dépendance spatiale de la fonction aléatoire $\delta(s)$, le semi-variogramme dû à l'hypothèse de stationnarité intrinsèque sera utilisé.

Ce sont les semi-variogrammes construis préalablement sur base des résidus pour chaque ETM qui ont été utilisés commme base pour cette partie. Ceux-ci servent donc à exprimer la structure de la dépendance spatiale des résidus. Les points étant fortement agglomérés, le choix a été fait d'inclure uniquement les 20 plus proches voisins pour toutes les prédictions par krigeage, comme c'était le cas pour la méthode par distance inverse. Cela signifie que le krigeage en question met l'accent sur les prédictions d'un point de vue local, au lieu de chercher une tendance dans la globalité du territoire.

Voici les différents résultats obtenus :

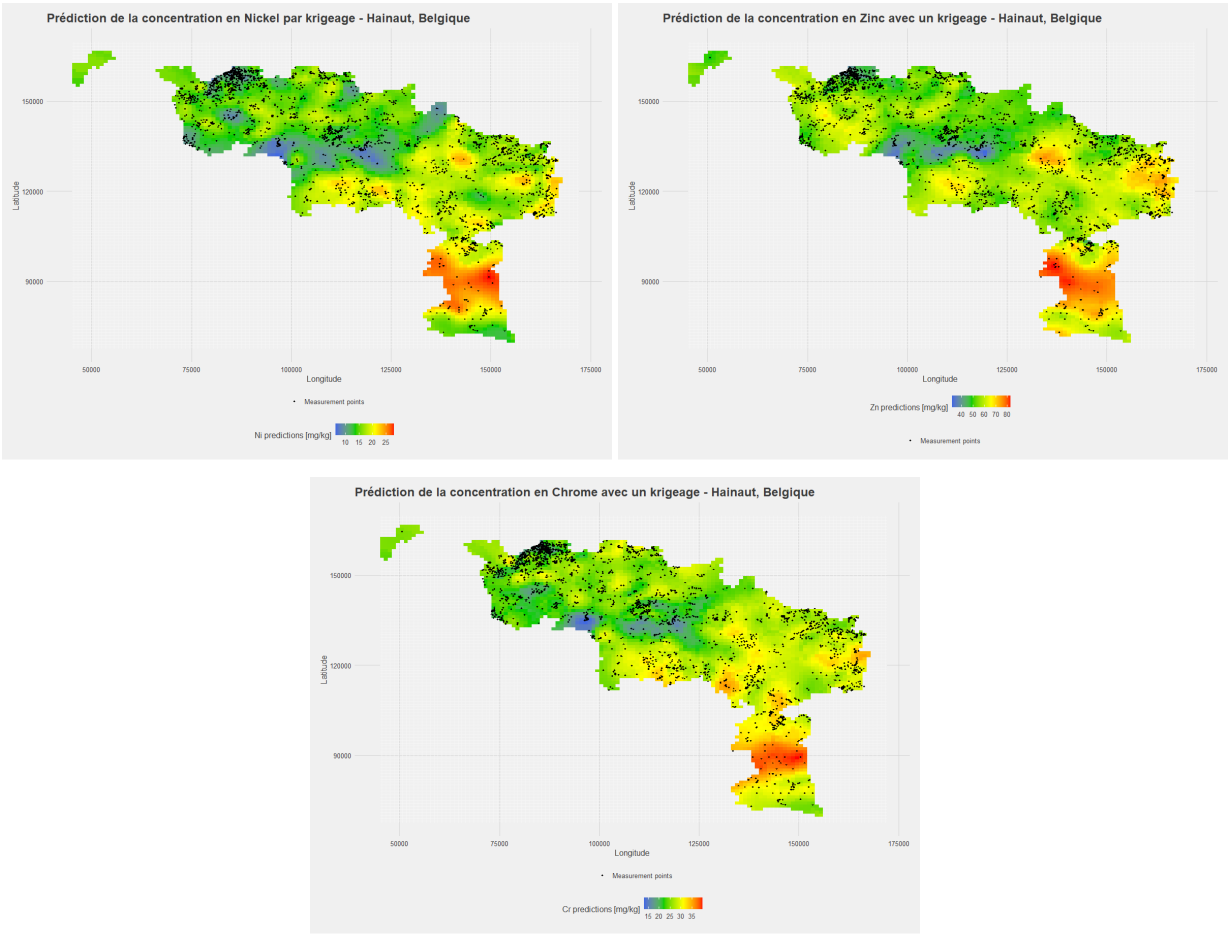


FIGURE 16 – Prédictions par krigeage des 3 ETM

Concernant la variance associée à cette méthode de prédiction, les résultats seront présentés dans la section *Discussion* afin d'apposer les résultats du krigeage et du co-krigeage côte-à-côte pour faciliter la visualisation des différences et ainsi la discussion.

5.3 Co-krigeage

Enfin, la dernière méthode employée pour réaliser la prédiction dérive en fait du krigeage. Le prédicteur estimé par cokrigeage prend la forme d'une combinaison linéaire pondérée des observations :

- De la variable régionalisée à interpoler ;
- D'une variable régionalisée auxiliaire.

Nous nous sommes basés sur l'analyse de correlation entre les variables réalisée lors de l'analyse exploratoire des données afin de choisir les variables présentant le plus de similitudes pour minimiser la variance des prédictions. Ainsi, le choix a été fait de réaliser la prédiction par cokrigeage du nickel avec le chrome comme variable auxiliaire.

Ci-après se trouve la carte de prédiction des concentrations du Nickel, avec visualisation des points de mesures du nickel ainsi que ceux du chrome.

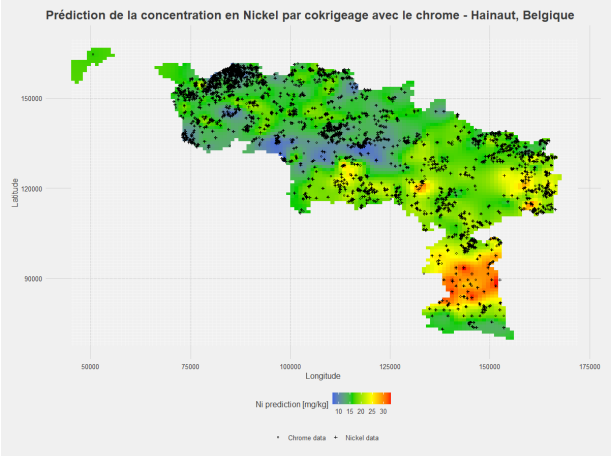


FIGURE 17 – Cokrigeage

6 Discussion

Nous allons maintenant procéder à des analyses de nos résultats obtenus via les prédictions détaillées dans la section précedente.

6.1 Prédiction

Les 3 méthodes de prédiction employées donnent des cartes fortement similaires, prédisant globalement des valeurs de concentration élevées ou faibles dans les mêmes zones.
Ceci est une bonne nouvelle car différentes méthodes présentant des résultats similaires attestent d'une plus grande robustesse de ces résultats.

On dénote cependant quelques différences en y regardant de plus près. En effet, le krigeage semble prédire des valeurs de concentration de manière plus lisse que la méthode IDW, où les démarcations entre les carrés de prédiction sont plus nettes (ceci est évident lorsqu'on observe les valeurs de concentrations élevées).
La méthode IDW prédit des valeurs légèrement à la baisse au voisinnage des pics de concentration

par rapport au krigeage. On remarque également que le co-krigeage semble prédire des valeurs plus faibles sur l'ensemble du territoire avec quelques exceptions.

On peut également se questionner sur la fiabilité des résultats dans les zones où les prises de mesure sont quasi-inexistantes. Il aurait peut-être été plus judicieux de prendre des mesures de façon plus homogène dans le but d'analyser toute la province du Hainaut. La zone à l'extrême nord-ouest est un bon exemple : elle ne comprend qu'un point de mesure de chrome et de zinc, et pourtant toutes les méthodes de prédictions y ont été appliquées. Par conséquent, on ne peut pas affirmer que les prédictions dans cette zone soient fiables.

Il est intéressant de mettre ces différents résultats en lien avec les cartes des sols de la Wallonie :

Avant cela, il faut noter que l'enclave hennuyère en territoire flandrien ne sera pas prise en compte dans l'analyse car elle n'a pas assez de données mesurées pour produire une bonne éstimation, comme cela a été expliqué auparavant.
Ainsi, les zones intéressantes à analyser sont (a) la zone où les concentrations des 3 ETM sont les plus élevée se situant en Fagne, (b) la zone relativement fort concentrée a l'est, et (c) les zones à concentrations faibles se trouvant au centre-ouest et nord-ouest de la province.

Lors de la comparaison de nos résultats avec la partie qui nous intéresse de la Carte de la localisation des 34 Régions-ETM en Région Wallonne (voir figure ci-dessous), on distingue aisément que la zone de concentrations les plus élevées (a) se trouve dans la Région-ETM "Fagne". Cette région correspond aux sols sur schistes, psammites, calcaires et fonds de vallées. L'autre zone à concentration plus élevée (b) correspond à des sols sur schistes, psammites et calcaires également. Les zones à faibles concentrations (c) sont le nord de la Plaine de l'Escaut et le bassin de la Haine. Ces régions correspondent à des sols sur sable tertiaire et loess.

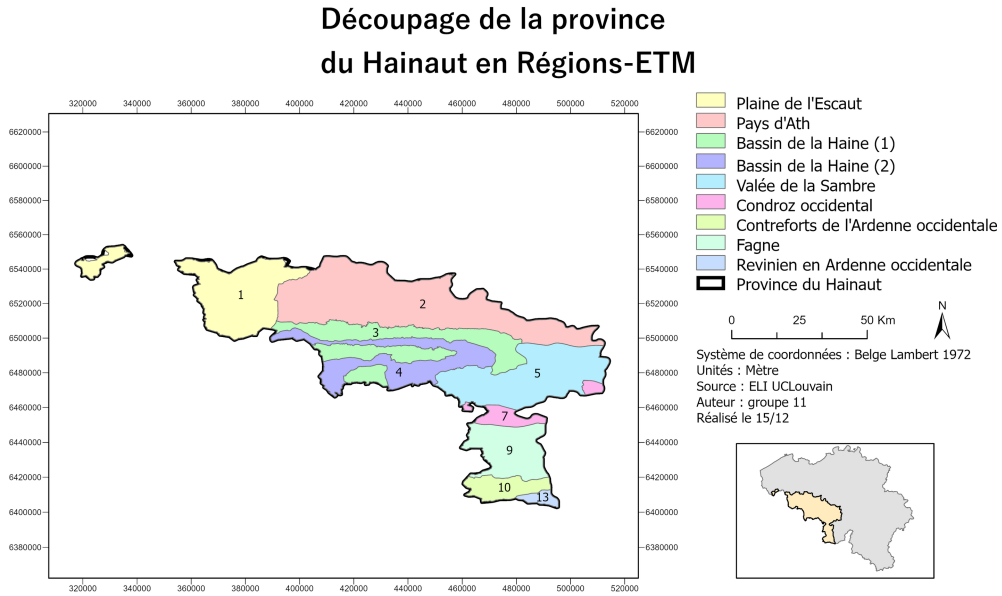


Figure 18 – Carte des différents ETM dans la province du Hainaut

D'après la Carte des Principaux Types de Sols de Wallonie, la région de Fagne (a) est principalement composée d'un sol limono-caillouteux à charge psammitique ou schisto-psammitique (plus vers le haut de cette zone, sur la bosse qui dépasse à gauche) et d'un sol limono-caillouteux à charge schisteuse (qui est localisé plus vers le centre de la région, dans le creux).

D'après la méthode déterministe IDW, les valeurs de zinc sont les plus concentrées sur les sols limono-caillouteux à charge psammitique, et le chrome est le plus concentré sur les sols limono-caillouteux à charge schisteuse. Le nickel quant à lui présente une zone de concentration qui recouvre de façon plus homogène les 2 types de sols. Dans la prédiction de la méthode stochastique krigeage, il ressort que les prédictions de concentrations ressemblent fortement à celles de l'IDW. Les sols à charge psammitique et schisteuse sont riches en quartz et en muscovite, qui sont tous les deux des minéraux contenant des éléments traces, notamment le chrome et le zinc. Nous pouvons donc expliquer ces hautes concentrations de métaux à la charge caillouteuse de cette zone là.

Quant à la seconde zone identifiée comme portant des concentrations prédites assez élevées (b), il est difficile d'établir des hypothèses avec certitude en se basant sur la Carte des Sols. En effet, le sol majoritaire dans cette zone n'est pas particulièrement défini, mais c'est un mélange de sols caillouteux, qui comme expliqué précedemment, apporte des concentrations en métaux plus élevées.

Les zones à faibles concentrations (c) se trouvent sur des sols artificiels autour des grandes villes comme Mons et Tournai.

6.2 Variance et différence

La prochaine analyse va maintenant se restreindre à la comparaison des méthodes stochastique du krigeage et de sa variation, le cokrigeage. Pour ce faire, les variances engendrées par l'utilisation de ces méthodes vont être comparées et ensuite discutées.

Voici ci-dessous les résultats correspondant :

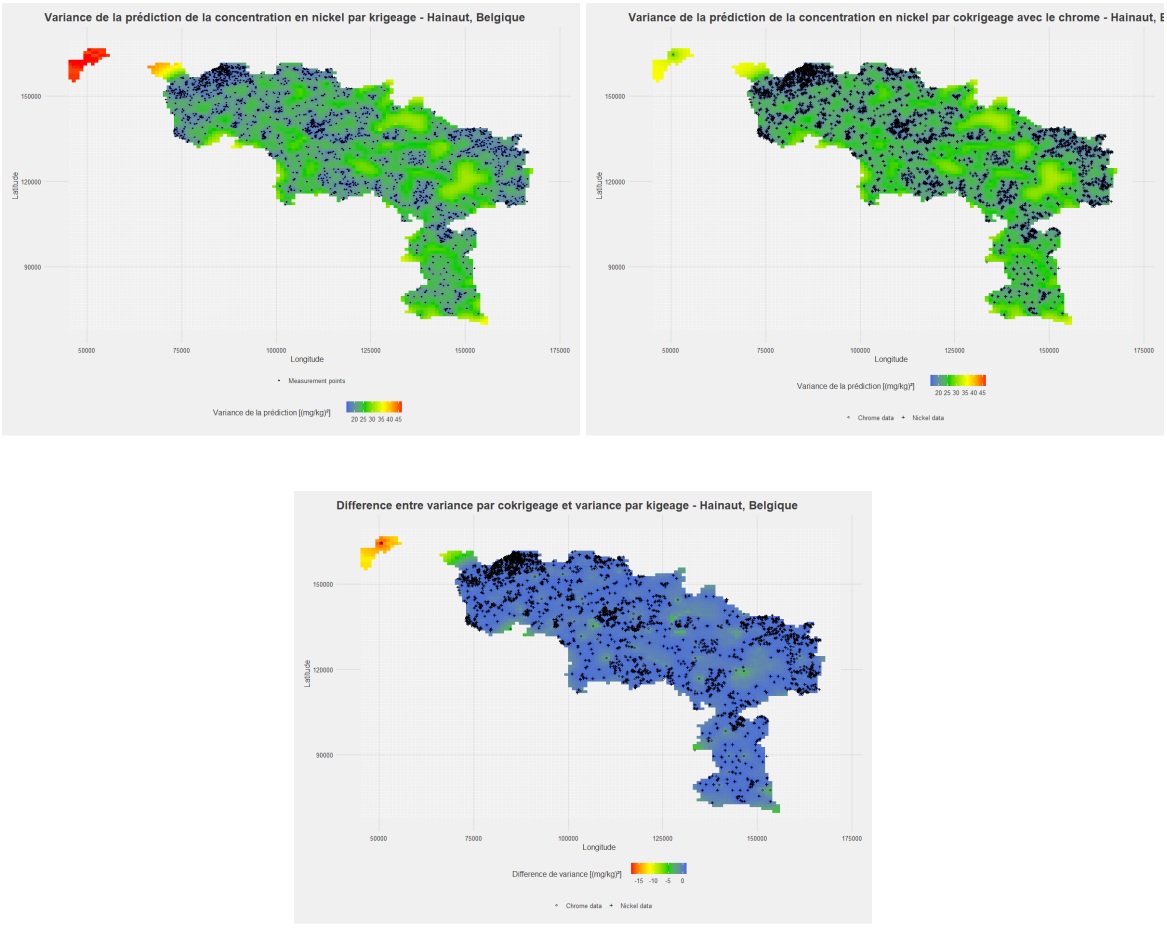


FIGURE 19 – Variance krigeage, cokrigeage et différence

On constate tout d'abord à première vue un grande similitude entre les variances sur la majorité du territoire. Lorsqu'une prédiction est réalisée loin d'un point de mesure, la variance de cette prédiction tend vers la variance du nickel. Lorsqu'on se trouve sur un point de mesure, la variance de prédiction est égale à 0 et ce, pour les deux méthodes.

Concernant les différences entre les méthodes de krigeage et co-krigeage, on remarque que l'énorme majorité du territoire se trouve en zone où cette différence est minime. Néanmoins, il apparait qu'en certains points précis, lorsqu'une donnée pour le chrome est disponible au contraire du nickel, la différence se veut plus importante, témoignant d'une qualité de prédiction meilleure pour le co-krigeage. Ainsi de manière très représentative, l'enclave hennuyère en territoire flandrien ne comporte aucun point de mesure pour le nickel et une unique mesure de chrome. On remarque que la différence de variance y est logiquement bien plus importante que sur le reste du territoire.
Cependant, concernant cette enclave bien précise, il faut rester prudent vu le faible nombre de mesure réalisé à cet emplacement, fortement éloigné du reste de la province.

Comme expliqué précédemment, puisque le cokrigeage tient compte également des points de mesure du chrome, le nombre de points disponible pour la prédiction est plus important que dans le cas du nickel seul. De ce fait, la variance résultant de la prédiction du co-krigeage se veut forcément plus faible que celle résultant du krigeage ordinaire, et ce plus l'échantillonnage de la variable auxiliaire sera important. Dans notre cas, nous utilisons 2406 observations de nickel contre 2792 observations de chrome. Une manière d'augmenter la fiabilité de la prédiction serait d'augmenter les points d'échantillonnage en nickel et en chrome, en particulier là où les observations se font plus rares. Un compromis entre fiabilité et ressources allouées à la prise de mesure doit donc être fait. Une meilleure répartition de cette prise de mesure sur la surface étudiée aurait été bénéfique afin d'affiner les prédictions.

7 Approfondissement : Cartes de risque de dépassement d'un seuil de Nickel

La dernière opération mise en place est la création de cartes de risque de dépassement d'un seuil pour le nickel. Le choix du nickel nous est paru tout naturel après avoir déjà effecuté la thématique du co-krigeage avec celui-ci.

En faisant des recherches, nous avons constaté que la valeur seuil permissible de la concentration du nickel dans plusieurs références sont plus élevées que toutes nos données. D'après le ministère de l'environnement de Finlande, la valeur de seuil permissible de la concentration du nickel dans les sols est de 50 mg/kg [4]. Dans le guide de référence pour l'étude de risques du département du sol et des déchets en wallonie, la valeur seuil relative à la protection des écosystèmes vaut 58 mg/kg [5]. Par ailleurs, dans le même guide, la valeur seuil pour la santé humaine des sols à usage agricole est encore plus élevée, avec comme valeurs 265 mg/kg [6].

Afin de pouvoir évaluer des cartes de risques intéréssantes, nous avons pris comme seuil la valeur 22 mg/kg, qui est la déviation standard supérieure de la moyenne. Mille simulations conditionnelles ont ensuite été engendrées pour obtenir une moyenne crédible. La première d'entre elles est représentée sur la carte à la figure 20.

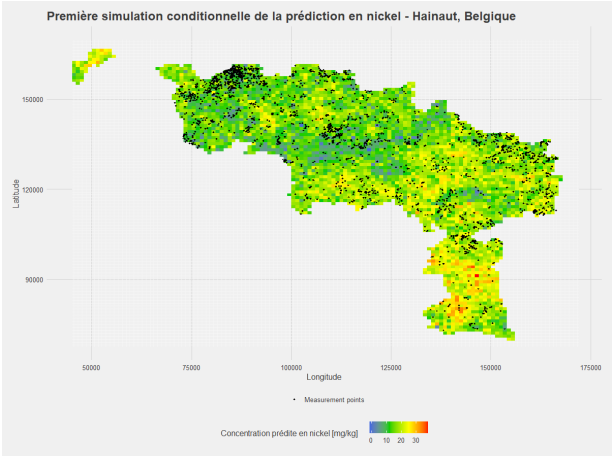


FIGURE 20 – Simulation conditionnelle des concentrations de Nickel avec les localisations des données en noir

Les zones à haut risque ont ensuite été déterminées, qui sont des zones où la probabilité que la concentration en Nickel soit plus élevées que 22mg/kg est de plus de 0.75. Vous pouvez retrouver la probabilité pour chaque prédiction à la figure 21.Ensuite, ces zones sont mises en évidence à la figure 22.

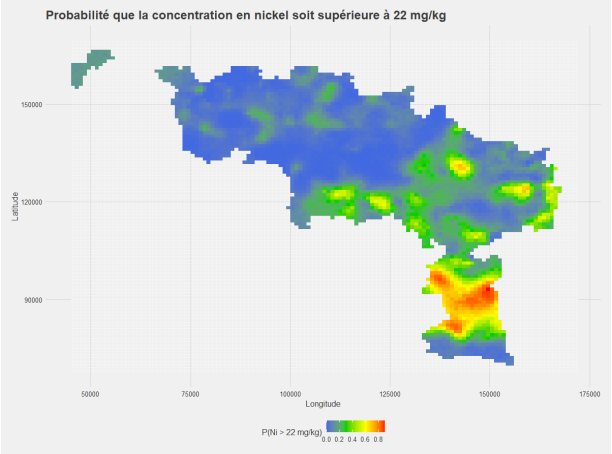


FIGURE 21 – Probabilité que Ni > 22mg/kg

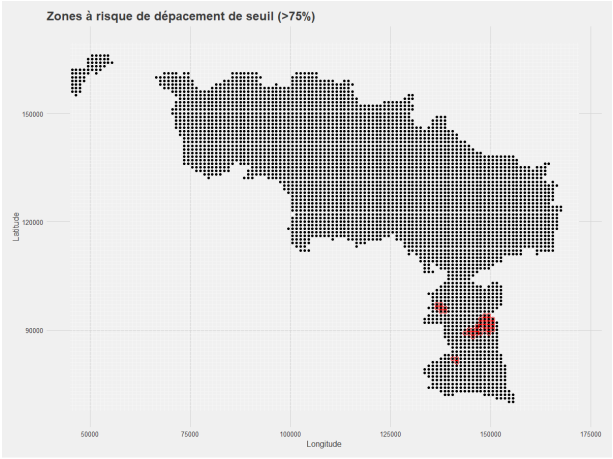


Figure 22 – Zones où la probabilité d'avoir une concentration plus élevée que 22mg/kg est plus élevée que 0.75.

Sans surprise, nous remarquons qu'ils sont concentrés dans la Fagne. Cependant, le choix d'un seuil largement inférieur à ceux trouvés dans la littérature permet d'assurer qu'il y a peu de risque de trouver des valeurs dangereuses pour la santé dans quelque région du Hainaut.

8 Conclusion

L'étape la plus importante du travail fut sans conteste la manipulation des données. Il a paru évident dès le début que la manipulation du jeu de données, son ajustement et ses différentes corrections constituraient une étape essentielle dans l'obtention de résultats fiables.

Grâce au regroupement des mesures prises en une même localisation, différentes prédictions ont pu être réalisées sur l'ensemble du territoire. On remarque notamment que la rétention d'éléments traces métalliques est plus favorisée dans la région de la Fagne que partout ailleurs.
Différentes hypothèses ont été soulevées pour permettre de justifier la répartition des concentrations, se basant sur la répartition des régions-ETM ainsi que sur les différents types de sols présents au sein de ces régions.

La méthode s'avérant être la plus fiable est le co-krigeage, présentant une somme des carrés des écarts ainsi qu'une variance toutes deux minimales. Sur cette base, des mesures d'amélioration de ces prédictions ont étés proposées, comme notamment la prise de mesure sur un territoire plus homogène, ou la prise en compte des affectations du sol à différentes activités anthropiques.

Enfin, nous avons développé sur base du krigeage un carte de risque de dépassement de seuil pour le nickel. Malgré quelques localisations à risque, les seuils choisis étant bien inférieurs à ceux trouvés dans la littérature, cela nous permet d'avancer qu'il est peu probable de dépasser un seuil de nickel réellement dangereux pour la santé dans la province du Hainaut.

9 Bibliographie

Références

[1] Statology, "How to Find Outliers Using the Interquartile Range", janvier 2021, consulté le 16 décembre 2021.
https ://www.statology.org/find-outliers-with-iqr/

[2] Bogaert P., "LBRTI 2101A Analyse statistique de données spatiales temporelles".

[3] SOPHIE BAILLARGEON, "Le krigeage : revue de la théorie et application à l'interpolation spatiale de données de précipitations", Avril 2005.
https ://www.mat.ulaval.ca/fileadmin/mat/documents/lrivest/EtudesGraduees/SBaillargeon.pdf

[4] Ministry of the Environment, Finland, "Government Decree on the Assessment of Soil Contamination and Remediation Needs", 1 March 2007.
https ://www.finlex.fi/en/laki/kaannokset/2007/en20070214.pdf ?fbclid=IwAR1X3kzjcrJjypBlFG5QWW1Iz
_wfUJMwhgBKgxz4dy-u1_8vXBn82s

[5] Direction générale operationnelle de l'agriculture, des ressources naturelles et de l'environnement "Annexe D : Liste des valeurs seuil génériques partielles et d'intervention relatives à la protection des écosystèmes", *Code Wallon de bonnes pratiques : Guide de référence pour l'étude de risques*, consulté le 15 décembre 2021.
https ://sol.environnement.wallonie.be/files/Document/CWBP/V03/GRER/PARTIE%20D/GRER_PARTIE%

[6] Direction générale operationnelle de l'agriculture, des ressources naturelles et de l'environnement "Annexe B1 : Liste des valeurs limites relatives à la protection de la santé humaine", *Code Wallon de bonnes pratiques : Guide de référence pour l'étude de risques*, consulté le 15 décembre 2021.
https ://sol.environnement.wallonie.be/files/Document/CWBP/V03/GRER/PARTIE%20B/GRER_PARTIEB%

[7] Tótha, G., Hermannb, T., Szatmáric, G., Pásztorc, L., "Maps of heavy metals in the soils of the European Union and proposed priority areas for detailed assessment" *Science of the Total Environment, Volume 565, Pages 1054-1062*, 15 September 2016.
https ://www.sciencedirect.com/science/article/pii/S0048969716310452 ?fbclid=IwAR0vWoOAEmZiKLDM
2vW0SEPFFWY3oygSRwJbiBIHI

[8] GISGeography, "Semi-Variogram : Nugget, Range and Sill,", Octobre 29, 2021, consulté le 15 décembre 2021.
https ://gisgeography.com/semi-variogram-nugget-range-sill/

[9] Aspexit, "Hypothèses fondamentales du variogramme : stationnarité d'ordre 2, stationnarité intrinsèque", 14 Aôut 2018, consulté le 15 décembre 2021.
https ://www.aspexit.com/hypotheses-fondamentales-du-variogramme-stationnarite-dordre-2-stationnarite-intrinseque-quest-ce-que-tout-cela-signifie-vraiment/

10 Annexe

10.1 Graphes auxiliaires

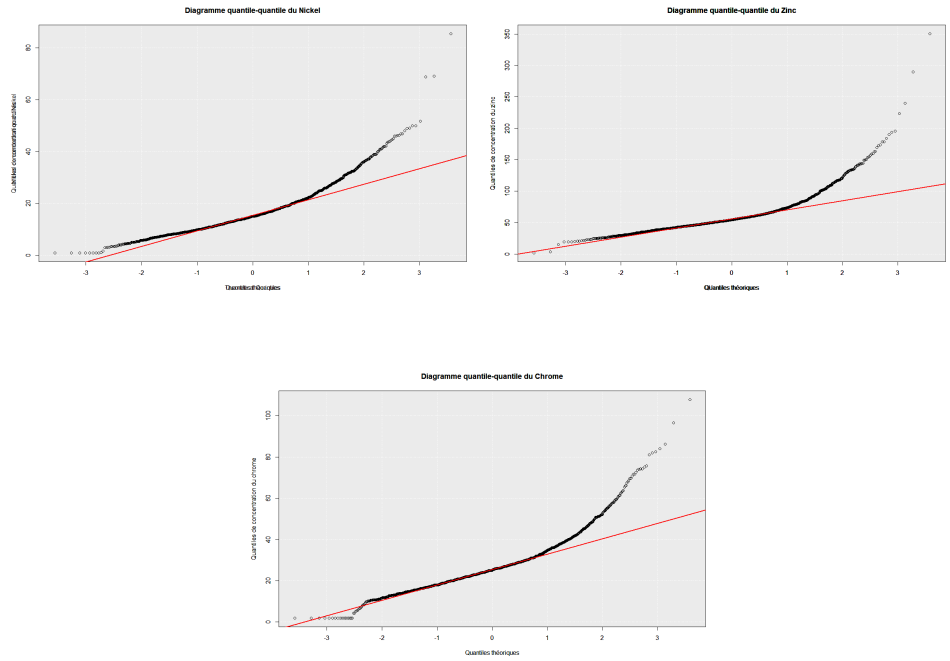


FIGURE 23 – QQplots des différents ETM avant retrait des valeurs aberrantes

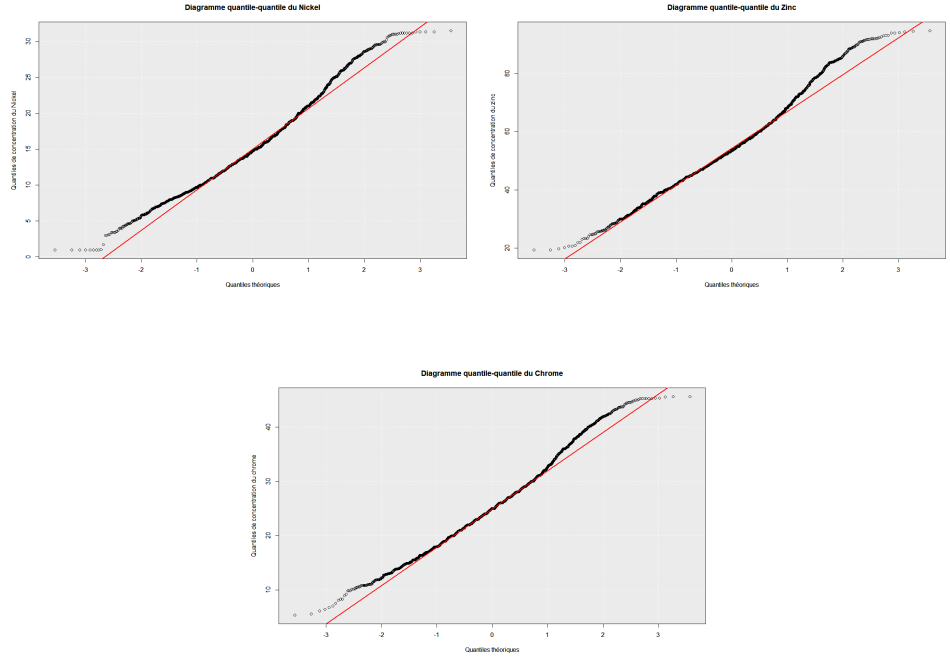


FIGURE 24 – QQplots des différents ETM après retrait des valeurs aberrantes

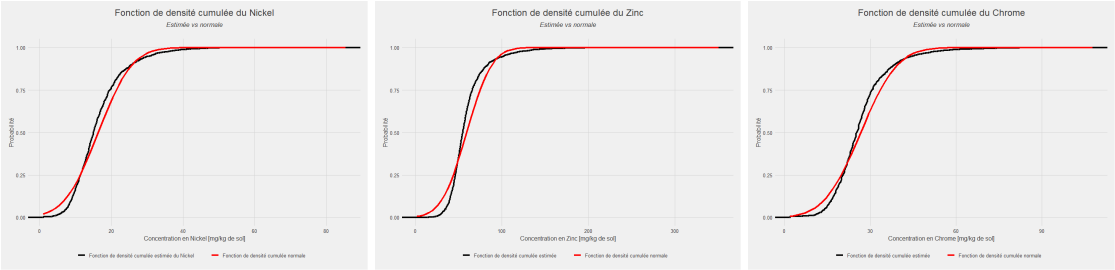


FIGURE 25 – Fonctions de densité cumulée estimées pour chacun des ETM

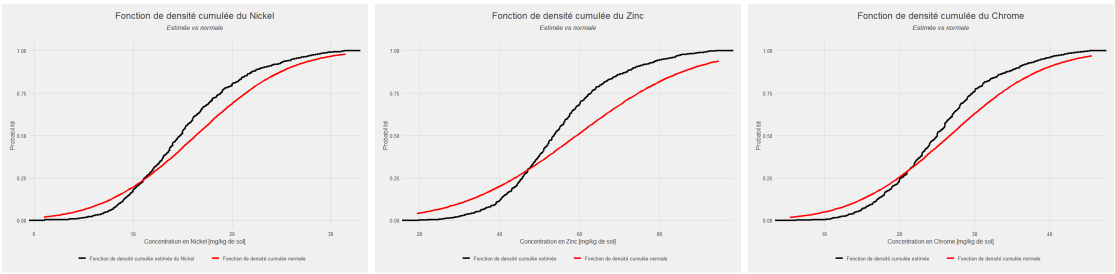


FIGURE 26 – Fonctions de densité cumulée estimées pour chacun des ETM après le nettoyage des données

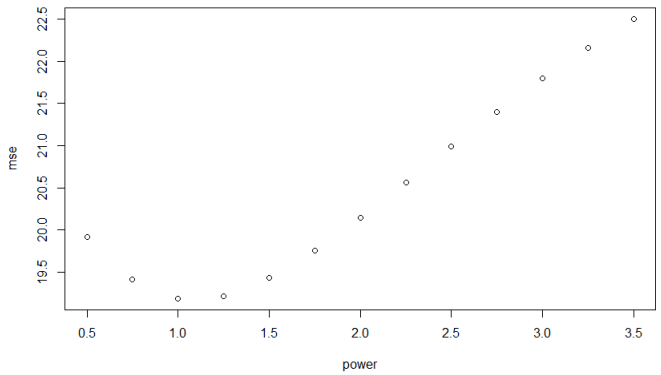


FIGURE 27 – Visualisation des sommes des carrés des écarts en fonction de θ

10.2 Listing des programmes complets

```
# Script final Projet Analyse de donn es spatiales - LBRTI2101(A) -------
# Import des donn es & Diff rents Packages -----


warning("Dont forget to change the working directory ! ")
setwd(dir = "/Users/gauthierheroufosse/Documents/Studies/MA 1/Q1/DataSciences - Partim A/Projet/")


rm(list=ls())


library(data.table)
library(ggplot2)
theme_set(theme_classic())
library(gridExtra)
library("PerformanceAnalytics")
library(car)
library(rgl)
library(gstat)
library(rgdal)
library(AID)
library(dplyr)
library(latticeExtra)
library(ggthemes)
library(tidyr)


Donnees <- fread("groupe11.csv")



# Analyse exploratoire des donn es -----


## Localisation de nos donn es -----

# On commence par renommer les colonnes de localisation
Donnees <- Donnees %>%
  rename(
    x = X,
    y = Y
  )


# Retrait de la double localisation

Donnees <- Donnees  %>%
  group_by(x,y) %>%
  summarise(Ni = mean(na.omit(Ni)), Zn = mean(na.omit(Zn)), Cr = mean(na.omit(Cr)))

Donnees <- data.table(Donnees)


##  Grillage de la province du Hainaut

gridsize = 1000 # Taille du maillage [m]
margin = 5000 # Ajout d'une marge pour  tre  s r d'englober toutes les donn es

x <- seq(floor(min(Donnees$x)-margin), # from minimum longitude
         ceiling(max(Donnees$x+margin)), # to maximum longitude
         by=gridsize)
y <- seq(floor(min(Donnees$y)-margin), # from minimum latitude
         ceiling(max(Donnees$y)+margin), # to maximum latitude
         by=gridsize)
hainaut.grid <- as.data.table(expand.grid(x=x, y=y))

## Create a spatial version of the grid
ETMdataSpatial <- copy(hainaut.grid)
coordinates(ETMdataSpatial) <- ~x+y
proj4string(ETMdataSpatial) <- CRS("+init=epsg:31370") # Specify coordinate system (Lambert belge 1972)

## Load provinces shapefile and specify it's coordinate system

provinces <- readOGR("/Users/gauthierheroufosse/Desktop/province/Hainaut.shp", use_iconv = TRUE, encoding = "UTF-8")
provinces <- spTransform(provinces, CRS("+proj=longlat +datum=WGS84"))
plot(provinces) # Display provinces
provinces$NAME_2 # Display all province names

## Transform grid coordinates system to match provinces
ETMdataSpatial <- spTransform(ETMdataSpatial, CRS("+proj=longlat +datum=WGS84"))

## Get index of points in the province of Hainaut
prov.id <- over(provinces[provinces$NAME_2 == "Hainaut", ],
                ETMdataSpatial,
                returnList = TRUE)

## Select rows in your original grid that are located in Hainaut
prov.grid = hainaut.grid[prov.id[[1]],]

# Spatial Ni repartition in Hainaut province
ggplot() +
  geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_point(data=prov.grid, aes(x=x, y=y), color='burlywood1', shape = 3) +
  geom_point(data=Donnees, aes(x=x, y=y, color=Ni),size=2) +
  labs(fill="", color="Ni [mg/kg]") +
  scale_color_gradientn(name="Nickel concentration [mg/kg of soil]",
                        colors=c('lightblue1','skyblue2', 'blue4', 'navyblue'), na.value = "black") +
  xlab("Longitude [Lambert 1972]") + ylab("Latitude [Lambert 1972]") +
  theme(plot.title=element_text(hjust=0.5)) +
```

```
  ggtitle("Nickel concentration repartition in Hainaut Province")+
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())

# Spatial Zn repartition in Hainaut province
ggplot() +
  geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_point(data=prov.grid, aes(x=x, y=y), color='burlywood1', shape = 3) +
  geom_point(data=Donnees,aes(x=x, y=y, color=Zn),size=2) +
  labs(fill="", color="Zn [mg/kg]") +
  scale_color_gradientn(name="Zinc concentration [mg/kg of soil]",
                        colors=c('mistyrose','lightcoral', 'indianred', 'indianred4'),na.value = "black") +
  xlab("Longitude [Lambert 1972]") + ylab("Latitude [Lambert 1972]") +
  theme(plot.title=element_text(hjust=0.5)) +
  ggtitle("Zinc concentration repartition in Hainaut province")+
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())

# Spatial Cr repartition in Hainaut province
ggplot() + geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_point(data=prov.grid, aes(x=x, y=y), color='burlywood1', shape = 3) +
  geom_point(data=Donnees,aes(x=x, y=y, color=Cr),size=2) +
  labs(fill="", color="Cr [mg/kg]") +
  scale_color_gradientn(name="Chrome concentration [mg/kg of soil]",
                        colors=c('lightgreen','darkseagreen', 'green4', 'darkseagreen4'), na.value = "black") +
  xlab("Longitude [Lambert 1972]") + ylab("Latitude [Lambert 1972]") +
  theme(plot.title=element_text(hjust=0.5)) +
  ggtitle("Chrome concentration repartition in Hainaut province")+
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())

## Statistiques de base ----

summary(Donnees)



# Visualisation de la distribution de nos variables
rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], # Light gray background
     col = "#ebebeb")

grid(nx = NULL, ny = NULL, col = "white", lty = 10, # Add white grid
     lwd = par("lwd"), equilogs = TRUE)

par(new = TRUE) # Boxplot

boxplot(Donnees$Ni,Donnees$Zn,Donnees$Cr,
        names = c("Nickel","Zinc", "Chrome"),
        main = "Boxplots of the different elements",
        col = rgb(0, 0, 1, alpha = 0.4),
        ylab="Concentration in the soil [mg/kg of soil]",
        las=1,
        border = "black",  # Boxplot border color
        outpch =23,      # Outliers symbol
        outbg = "red",   # Outliers color
        whiskcol = "blue", # Whisker color
        whisklty = 1      # Whisker line type
)


# Rel ve de la pr sence de NaN parmi nos donn es
sum(is.na(Donnees$Ni))
sum(is.na(Donnees$Zn))
sum(is.na(Donnees$Cr))


# Correlation entre variables

chart.Correlation(Donnees[,.(Ni,Cr,Zn)])



# Nettoyage des donn es ----

## Distribution des concentrations et corrections ----

### Ni ----

ggplot(Donnees, aes(x = Ni)) + # if you put 'aes' here, all geom_ functions will use Ni as x
  geom_histogram(aes(y = ..density.., # display density function (and not count)
                     fill = "Density histogram"),
                 bins = 15, # number of bins
                 color = "dodgerblue4") +
  geom_density(col = "dodgerblue1", # Bleue curve
               aes(fill = "Fitted probability density function"),
               alpha = 0.5) +
  xlab("Nickel concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle("Density histogram of Nickel concentration") +
  theme(axis.title = element_text())+
  stat_function(fun=dnorm, # display a normal distribution -> red curve
                args=list(mean = mean(na.omit(Donnees$Ni)), sd = sd(na.omit(Donnees$Ni))), # with mean and sd of Ni
                aes(color="Normal probability density function"),
                size= 1.5) + # width of the line
  scale_fill_manual("", values = c("Density histogram"="dodgerblue4",
                                   "Fitted probability density function"=alpha("dodgerblue1",.2))) +
  scale_color_manual("", values = ("Normal probability density function" = "red"))

ggplot(Donnees, aes(x = Ni)) +
  stat_ecdf(geom = "step", aes(color = "Estimated cumulative density function"), size = 1.5) +
  xlab("Nickel concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle(expression(atop("Cumulative density function of the Nickel concentration",
```

```
                        atop(italic("Estimated vs normal"), ""))) +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun = pnorm, # display a normal cumulative distribution
                args = list(mean = mean(na.omit(Donnees$Ni)), sd = sd(na.omit(Donnees$Ni))),
                aes(color = 'Normal cumulative density function'),
                size= 1.5) +
  scale_color_manual("", values = c("Estimated cumulative density function" = "black",
                                    "Normal cumulative density function" = "red"))

car::qqPlot(na.omit(Donnees$Ni)) #Vieux qq plot

#Nouveau qqplot

rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], # Light gray background
     col = "#ebebeb")

grid(nx = NULL, ny = NULL, col = "white", lty = 10, # Add white grid
     lwd = par("lwd"), equilogs = TRUE)

par(new = TRUE)

qqnorm(na.omit(Donnees$Ni), pch = 1, frame = TRUE, grid = TRUE, main="Nickel quantile-quantile diagram",
       ylab ="Nickel concentration quantiles") #Nouveau qqplot
qqline(na.omit(Donnees$Ni), col = "red", lwd = 2)

# Correction via une Box cox transformation

Cdbx.Ni <- boxcoxnc(na.omit(Donnees$Ni), verbose = FALSE) # Find best alpha

Donnees[!is.na(Ni), Cdbx.Ni := Cdbx.Ni$tf.data] # Create column with transformed data
alpha <- Cdbx.Ni$lambda.hat


### Zn ----

ggplot(Donnees, aes(x = Zn)) + # if you put 'aes' here, all geom_ functions will use Zn as x
  geom_histogram(aes(y = ..density..,  # display density function (and not count)
                     fill = "Density histogram"),
                 bins = 15, # number of bins
                 color = "dodgerblue4") +
  geom_density(col = "dodgerblue1", # Bleue curve
               aes(fill = "Fitted probability density function"),
               alpha = 0.5) +
  xlab("Zinc concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle("Density histogram of Zinc concentration") +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun=dnorm, # display a normal distribution -> red curve
                args=list(mean = mean(na.omit(Donnees$Zn)), sd = sd(na.omit(Donnees$Zn))), # with mean and sd of Zn
                aes(color="Normal probability density function"),
                size= 1.5) + # width of the line
  scale_fill_manual("", values = c("Density histogram"="dodgerblue4",
                                   "Fitted probability density function"=alpha("dodgerblue1",.2))) +
  scale_color_manual("", values = ("Normal probability density function" = "red"))

ggplot(Donnees, aes(x = Zn)) +
  stat_ecdf(geom = "step", aes(color = "Estimated cumulative density function"), size = 1.5) +
  xlab("Zinc concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle(expression(atop("Cumulative density function of the Zinc concentration",
                          atop(italic("Estimated vs normal"), "")))) +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun = pnorm, # display a normal cumulative distribution
                args = list(mean = mean(na.omit(Donnees$Zn)), sd = sd(na.omit(Donnees$Zn))),
                aes(color = 'Normal cumulative density function'),
                size= 1.5) +
  scale_color_manual("", values = c("Estimated cumulative density function" = "black",
                                    "Normal cumulative density function" = "red"))

car::qqPlot(na.omit(Donnees$Zn))#Vieux qqplot

rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], # Light gray background
     col = "#ebebeb")

grid(nx = NULL, ny = NULL, col = "white", lty = 10, # Add white grid
     lwd = par("lwd"), equilogs = TRUE)

par(new = TRUE)

qqnorm(na.omit(Donnees$Zn), pch = 1, frame = TRUE, grid = TRUE, main="Zinc quantile-quantile diagram",
       ylab ="Zinc concentration quantiles") #Nouveau qqplot
qqline(na.omit(Donnees$Zn), col = "red", lwd = 2)

# Correction via une Box cox transformation

Cdbx.Zn <- boxcoxnc(na.omit(Donnees$Zn), verbose = FALSE) # Find best alpha

Donnees[!is.na(Zn),Cdbx.Zn := Cdbx.Zn$tf.data] # Create column with transformed data
alpha <- Cdbx.Zn$lambda.hat

### Cr ----

ggplot(Donnees, aes(x = Cr)) + # if you put 'aes' here, all geom_ functions will use Cr as x
  geom_histogram(aes(y = ..density..,  # display density function (and not count)
                     fill = "Density histogram"),
                 bins = 15, # number of bins
                 color = "dodgerblue4") +
  geom_density(col = "dodgerblue1", # Bleue curve
               aes(fill = "Fitted probability density function"),
               alpha = 0.5) +
```

```
  xlab("Chrome concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle("Density histogram of Chrome concentration") +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun=dnorm, # display a normal distribution -> red curve
                args=list(mean = mean(na.omit(Donnees$Cr)), sd = sd(na.omit(Donnees$Cr))), # with mean and sd of rain
                aes(color="Normal probability density function"),
                size= 1.5) + # width of the line
  scale_fill_manual("", values = c("Density histogram"="dodgerblue4",
                                   "Fitted probability density function"=alpha("dodgerblue1",.2))) +
  scale_color_manual("", values = ("Normal probability density function" = "red"))

ggplot(Donnees, aes(x = Cr)) +
  stat_ecdf(geom = "step", aes(color = "Estimated cumulative density function"), size = 1.5) +
  xlab("Chrome concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle(expression(atop("Cumulative density function of the Chrome concentration",
                          atop(italic("Estimated vs normal"), "")))) +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun = pnorm, # display a normal cumulative distribution
                args = list(mean = mean(na.omit(Donnees$Cr)), sd = sd(na.omit(Donnees$Cr))),
                aes(color = 'Normal cumulative density function'),
                size= 1.5) +
  scale_color_manual("", values = c("Estimated cumulative density function" = "black",
                                    "Normal cumulative density function" = "red"))

car::qqPlot(na.omit(Donnees$Cr))#Vieux qqplot

rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], # Light gray background
     col = "#ebebeb")

grid(nx = NULL, ny = NULL, col = "white", lty = 10, # Add white grid
     lwd = par("lwd"), equilogs = TRUE)

par(new = TRUE)

qqnorm(na.omit(Donnees$Cr), pch = 1, frame = TRUE, grid = TRUE, main="Chrome quantile-quantile diagram",
       ylab ="Chrome concentration quantiles") #Nouveau qqplot
qqline(na.omit(Donnees$Cr), col = "red", lwd = 2)

# Correction via une Box cox transformation

Cdbx.Cr <- boxcoxnc(na.omit(Donnees$Cr), verbose = FALSE) # Find best alpha

Donnees[!is.na(Cr),Cdbx.Cr := Cdbx.Cr$tf.data] # Create column with transformed data
alpha <- Cdbx.Cr$lambda.hat



## Retrait des valeurs aberrantes ----
### Ni ----

Q1n <- quantile(na.omit(Donnees$Ni), .25)
Q3n <- quantile(na.omit(Donnees$Ni), .75)
IQRn <- IQR(na.omit(Donnees$Ni))

clean.donnees.Ni <- subset(Donnees, Donnees$Ni > (Q1n - 1.5*IQRn) & Donnees$Ni < (Q3n + 1.5*IQRn))

### Zn ----
Q1z <- quantile(na.omit(Donnees$Zn), .25)
Q3z <- quantile(na.omit(Donnees$Zn), .75)
IQRz <- IQR(na.omit(Donnees$Zn))

clean.donnees.Zn <- subset(Donnees, Donnees$Zn > (Q1z - 1.5*IQRz) & Donnees$Zn < (Q3z + 1.5*IQRz))


### Cr ----
Q1c <- quantile(na.omit(Donnees$Cr), .25)
Q3c <- quantile(na.omit(Donnees$Cr), .75)
IQRc <- IQR(na.omit(Donnees$Cr))

clean.donnees.Cr <- subset(Donnees, Donnees$Cr > (Q1c - 1.5*IQRc) & Donnees$Cr < (Q3c + 1.5*IQRc))




## Visualisation des nouvelles distributions ----


rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4],# Light gray background
     col = "#ebebeb")
grid(nx = NULL, ny = NULL, col = "white", lty = 10,# Add white grid
     lwd = par("lwd"), equilogs = TRUE)
par(new = TRUE)

boxplot(clean.donnees.Ni$Ni,clean.donnees.Zn$Zn, clean.donnees.Cr$Cr,
        names = c("Nickel", "Zinc", "Chrome"),
        main = "New boxplots of the different elements",
        col = rgb(0, 0, 1, alpha = 0.4),
        ylab="Concentration in the soil [mg/kg of soil]",
        las=1,
        border = "black",  # Boxplot border color
        outpch =23,      # Outliers symbol
        outbg = "red",   # Outliers color
        whiskcol = "blue", # Whisker color
        whisklty = 1      # Whisker line type
        )
### Ni ----
```

```
ggplot(clean.donnees.Ni, aes(x = Ni)) + # if you put 'aes' here, all geom_ functions will use rain as x
  geom_histogram(aes(y = ..density..,  # display density function (and not count)
                     fill = "Density histogram"),
                 bins = 15, # number of bins
                 color = "dodgerblue4") +
  geom_density(col = "dodgerblue1", # Bleue curve
               aes(fill = "Fitted probability density function"),
               alpha = 0.5) +
  xlab("Nickel concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle("Density histogram of Nickel concentration") +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun=dnorm, # display a normal distribution -> red curve
                args=list(mean = mean(clean.donnees.Ni$Ni), sd = sd(clean.donnees.Ni$Ni)), # with mean and sd of Ni
                aes(color="Normal probability density function"),
                size= 1.5) + # width of the line
  scale_fill_manual("", values = c("Density histogram"="dodgerblue4",
                                   "Fitted probability density function"=alpha("dodgerblue1",.2))) +
  scale_color_manual("", values = ("Normal probability density function" = "red"))


ggplot(clean.donnees.Ni, aes(x = Ni)) +
  stat_ecdf(geom = "step", aes(color = "Estimated cumulative density function"), size = 1.5) +
  xlab("Nickel concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle(expression(atop("Cumulative density function of the Nickel concentration",
                          atop(italic("Estimated vs normal"), "")))) +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun = pnorm, # display a normal cumulative distribution
                args = list(mean = mean(clean.donnees.Ni$Ni), sd = sd(clean.donnees.Ni$Ni)),
                aes(color = 'Normal cumulative density function'),
                size= 1.5) +
  scale_color_manual("", values = c("Estimated cumulative density function" = "black",
                                    "Normal cumulative density function" = "red"))


car::qqPlot(clean.donnees.Ni$Ni) #Vieux qqplot

#Nouveau QQplot

rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], # Light gray background
     col = "#ebebeb")

grid(nx = NULL, ny = NULL, col = "white", lty = 10, # Add white grid
     lwd = par("lwd"), equilogs = TRUE)

par(new = TRUE)


qqnorm(clean.donnees.Ni$Ni, pch = 1, frame = TRUE, grid = TRUE, main="Nickel quantile-quantile diagram",
       ylab ="Nickel concentration quantiles") #Nouveau qqplot
qqline(clean.donnees.Ni$Ni, col = "red", lwd = 2)

### Zn ----

ggplot(clean.donnees.Zn, aes(x = Zn)) + # if you put 'aes' here, all geom_ functions will use rain as x
  geom_histogram(aes(y = ..density..,  # display density function (and not count)
                     fill = "Density histogram"),
                 bins = 15, # number of bins
                 color = "dodgerblue4") +
  geom_density(col = "dodgerblue1", # Bleue curve
               aes(fill = "Fitted probability density function"),
               alpha = 0.5) +
  xlab("Zinc concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle("Density histogram of Zinc concentration") +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun=dnorm, # display a normal distribution -> red curve
                args=list(mean = mean(clean.donnees.Zn$Zn), sd = sd(clean.donnees.Zn$Zn)), # with mean and sd of Zn
                aes(color="Normal probability density function"),
                size= 1.5) + # width of the line
  scale_fill_manual("", values = c("Density histogram"="dodgerblue4",
                                   "Fitted probability density function"=alpha("dodgerblue1",.2))) +
  scale_color_manual("", values = ("Normal probability density function" = "red"))

ggplot(clean.donnees.Zn, aes(x = Zn)) +
  stat_ecdf(geom = "step", aes(color = "Estimated cumulative density function"), size = 1.5) +
  xlab("Zinc concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle(expression(atop("Cumulative density function of the Zinc concentration",
                          atop(italic("Estimated vs normal"), "")))) +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun = pnorm, # display a normal cumulative distribution
                args = list(mean = mean(clean.donnees.Zn$Zn), sd = sd(clean.donnees.Zn$Zn)),
                aes(color = 'Normal cumulative density function'),
                size= 1.5) +
  scale_color_manual("", values = c("Estimated cumulative density function" = "black",
                                    "Normal cumulative density function" = "red"))

car::qqPlot(clean.donnees.Zn$Zn)#Vieux qqplot

#Nouveau qqplot

rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], # Light gray background
     col = "#ebebeb")

grid(nx = NULL, ny = NULL, col = "white", lty = 10, # Add white grid
     lwd = par("lwd"), equilogs = TRUE)
```

```
par(new = TRUE)

qqnorm(clean.donnees.Zn$Zn, pch = 1, frame = TRUE, grid = TRUE, main="Zinc quantile-quantile diagram",
       ylab ="Zinc concentration quantiles") #Nouveau qqplot
qqline(clean.donnees.Zn$Zn, col = "red", lwd = 2)

### Cr ----

ggplot(clean.donnees.Cr, aes(x = Cr)) + # if you put 'aes' here, all geom_ functions will use rain as x
  geom_histogram(aes(y = ..density.., # display density function (and not count)
                     fill = "Density histogram"),
                 bins = 15, # number of bins
                 color = "dodgerblue4") +
  geom_density(col = "dodgerblue1", # Bleue curve
               aes(fill = "Fitted probability density function"),
               alpha = 0.5) +
  xlab("Chrome concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle("Density histogram of Chrome concentration") +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun=dnorm, # display a normal distribution -> red curve
                args=list(mean = mean(clean.donnees.Cr$Cr), sd = sd(clean.donnees.Cr$Cr)), # with mean and sd of rain
                aes(color="Normal probability density function"),
                size= 1.5) + # width of the line
  scale_fill_manual("", values = c("Density histogram"="dodgerblue4",
                                   "Fitted probability density function"=alpha("dodgerblue1",.2))) +
  scale_color_manual("", values = ("Normal probability density function" = "red"))

ggplot(clean.donnees.Cr, aes(x = Cr)) +
  stat_ecdf(geom = "step", aes(color = "Estimated cumulative density function"), size = 1.5) +
  xlab("Chrome concentration [mg/Kg]") +
  ylab("Probability") +
  ggtitle(expression(atop("Cumulative density function of the Chrome concentration",
                          atop(italic("Estimated vs normal"), "")))) +
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())+
  stat_function(fun = pnorm, # display a normal cumulative distribution
                args = list(mean(clean.donnees.Cr$Cr), sd = sd(clean.donnees.Cr$Cr)),
                aes(color = 'Normal cumulative density function'),
                size= 1.5) +
  scale_color_manual("", values = c("Estimated cumulative density function" = "black",
                                    "Normal cumulative density function" = "red"))

car::qqPlot(clean.donnees.Cr$Cr)#Vieux qqplot

#Nouveau qqplot

rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], # Light gray background
     col = "#ebebeb")

grid(nx = NULL, ny = NULL, col = "white", lty = 10, # Add white grid
     lwd = par("lwd"), equilogs = TRUE)

par(new = TRUE)

qqnorm(clean.donnees.Cr$Cr, pch = 1, frame = TRUE, grid = TRUE, main="Chrome quantile-quantile diagram",
       ylab ="Chrome concentration quantiles") #Nouveau qqplot
qqline(clean.donnees.Cr$Cr, col = "red", lwd = 2)


# Mod lisation spatiale ----

## Fonction esp rance ----



## Variogramme ----

### Variogramme initial apr s correction des donn es ----
#### Ni ----

# First we create a gstat object
Ni.gstat <- gstat(id ="Ni", formula = Ni~1, data = clean.donnees.Ni, locations = ~x+y)

# Then we calculate the semi-variogram values
Ni.vario <- variogram(Ni.gstat, cutoff = 18000, width = 1200)
Ni.vg.fit <- fit.variogram(Ni.vario, model=vgm(c('Exp','Sph','Gau')))
head(Ni.vario)


plot(Ni.vario,  model=Ni.vg.fit, main = "Variogram of Nickel", pch=16,col='black',ylim=c(0,40),
     xlab="Distance", ylab="Semivariance", lty = 1, lwd = 3)

trellis.focus("panel",1,1)
llines(x=c(0,max(Ni.vario$dist)), y=c(var(clean.donnees.Ni$Ni),var(clean.donnees.Ni$Ni)), col="red", lwd=1, lty=2)
trellis.unfocus()

#### Zn ----

# First we create a gstat object
Zn.gstat <- gstat(id ="Zn", formula = Zn~1, data = clean.donnees.Zn, locations = ~x+y)

# Then we calculate the semi-variogrma values
Zn.vario <- variogram(Zn.gstat, cutoff = 20000, width = 1800)
Zn.vg.fit <- fit.variogram(Zn.vario, model=vgm(c('Exp','Sph','Gau')))
head(Zn.vario)


plot(Zn.vario,  model=Zn.vg.fit, main = "Variogram of Zinc", pch=16,col='black', ylim=c(0,200),
     xlab="Distance", ylab="Semivariance", lty = 1, lwd = 3)

trellis.focus("panel",1,1)
```

```
llines(x=c(0,max(Zn.vario$dist)), y=c(var(clean.donnees.Zn$Zn),var(clean.donnees.Zn$Zn)), col="red", lwd=1, lty=2)
trellis.unfocus()

#### Cr ----

# First we create a gstat object
Cr.gstat <- gstat(id ="Cr", formula = Cr~1, data = clean.donnees.Cr, locations = ~x+y)

# Then we calculate the semi-variogram values
Cr.vario <- variogram(Cr.gstat, cutoff = 12000, width = 600)
Cr.vg.fit <- fit.variogram(Cr.vario, model=vgm(c('Exp','Sph','Gau')))
head(Cr.vario)


plot(Cr.vario,  model=Cr.vg.fit, main = "Variogram of Chrome", pch=16,col='black', ylim=c(0,60),
     xlab="Distance", ylab="Semivariance", lty = 1, lwd = 3)

trellis.focus("panel",1,1)
llines(x=c(0,max(Cr.vario$dist)), y=c(var(clean.donnees.Cr$Cr),var(clean.donnees.Cr$Cr)), col="red", lwd=1, lty=2)
trellis.unfocus()

### Variogramme des r sidus apr s retrait de l'influence de la moyenne ----
#### Ni ----

# Visualisation des donn es
Ni.lm <- lm(Ni ~x+y, data=clean.donnees.Ni)

scatter3d(x=clean.donnees.Ni$x, z=clean.donnees.Ni$y, y=clean.donnees.Ni$Ni, xlab="Longitude", zlab ="Latitude",
          ylab='Ni [mg/kg]')
rglwidget(width=600, height=600, reuse=FALSE)


# Retrait de l'influence de la moyenne
Ni.derive <- predict(Ni.lm, clean.donnees.Ni)
clean.donnees.Ni[, Ni.res:=Ni-Ni.derive]

# Visualisation des r sidus
Ni_res.lm <- lm(Ni.res ~x+y, data=clean.donnees.Ni)
scatter3d(x=clean.donnees.Ni$x, z=clean.donnees.Ni$y, y=clean.donnees.Ni$Ni.res,
          xlab="Longitude", zlab ="Latitude", ylab='Ni residuals [mg/Kg]')
rglwidget(width=600, height=600, reuse=FALSE)


#  Variogramme des r sidus
Ni.res.gstat <- gstat(formula = Ni.res~1, data = clean.donnees.Ni, locations = ~x+y)

# Then we calculate the semi-variogrma values
Ni.res.vario <- variogram(Ni.res.gstat, cutoff = 18000, width = 900)
head(Ni.res.vario)

clean.fit.ni <- fit.variogram(Ni.res.vario, model=vgm(c(model='Exp','Sph','Gau'), nugget = 15), fit.method = 6)

plot(Ni.res.vario, model = clean.fit.ni, main = "Variogram of Nickel residuals", pch=16,col='black',
     xlab="Distance", ylab="Semivariance", lty = 1, lwd = 3)

trellis.focus("panel",1,1)
llines(x=c(0,max(Ni.res.vario$dist)), y=c(var(clean.donnees.Ni$Ni.res),var(clean.donnees.Ni$Ni.res)),
       col="red", lwd=1, lty=2)
trellis.unfocus()

# Visualisaion des param tres optimis s du mod le
clean.fit.ni


#### Zn ----

# Visualisation des donn es
Zn.lm <- lm(Zn ~x+y, data=clean.donnees.Zn)
scatter3d(x=clean.donnees.Zn$x, z=clean.donnees.Zn$y, y=clean.donnees.Zn$Zn, xlab="Longitude",
          zlab ="Latitude", ylab='Zn [mg/kg]')
rglwidget(width=600, height=600, reuse=FALSE)

# Retrait de l'influence de la moyenne
Zn.derive <- predict(Zn.lm, clean.donnees.Zn)
clean.donnees.Zn[, Zn.res:=Zn-Zn.derive]

# Visualisation des r sidus
Zn_res.lm <- lm(Zn.res ~x+y, data=clean.donnees.Zn)
scatter3d(x=clean.donnees.Zn$x, z=clean.donnees.Zn$y, y=clean.donnees.Zn$Zn.res,
          xlab="Longitude", zlab ="Latitude", ylab='Zn residuals [mg/Kg]')
rglwidget(width=600, height=600, reuse=FALSE)


#  Variogramme des r sidus
Zn.res.gstat <- gstat(formula = Zn.res~1, data = clean.donnees.Zn, locations = ~x+y)

# Then we calculate the semi-variogrma values
Zn.res.vario <- variogram(Zn.res.gstat, cutoff = 15000, width = 1500)
head(Zn.res.vario)

clean.fit.zn <- fit.variogram(Zn.res.vario, model=vgm(c('Exp','Sph','Gau'), nugget = 85), fit.method = 6)

plot(Zn.res.vario, model = clean.fit.zn, main = "Variogram of Zinc residuals", pch=16,col='black',
     xlab="Distance", ylab="Semivariance", lty = 1, lwd = 3)

trellis.focus("panel",1,1)
llines(x=c(0,max(Zn.res.vario$dist)), y=c(var(clean.donnees.Zn$Zn.res),var(clean.donnees.Zn$Zn.res)),
       col="red", lwd=1, lty=2)
trellis.unfocus()

# Visualisaion des param tres optimis s du mod le
clean.fit.zn
```

```
#### Cr ----

# Visualisation des donn es
Cr.lm <- lm(Cr ~x+y, data=clean.donnees.Cr)
scatter3d(x=clean.donnees.Cr$x, z=clean.donnees.Cr$y, y=clean.donnees.Cr$Cr, xlab="Longitude", zlab ="Latitude",
          ylab='Cr [mg/kg]')
rglwidget(width=600, height=600, reuse=FALSE)

# Retrait de l'influence de la moyenne
Cr.derive <- predict(Cr.lm, clean.donnees.Cr)
clean.donnees.Cr[, Cr.res:=Cr-Cr.derive]

# Visualisation des r sidus
Cr_res.lm <- lm(Cr.res ~x+y, data=clean.donnees.Cr)
scatter3d(x=clean.donnees.Cr$x, z=clean.donnees.Cr$y, y=clean.donnees.Cr$Cr.res, xlab="Longitude", zlab ="Latitude",
          ylab='Cr residuals [mg/Kg]')
rglwidget(width=600, height=600, reuse=FALSE)


#  Variogramme des r sidus
Cr.res.gstat <- gstat(formula = Cr.res~1, data = clean.donnees.Cr, locations = ~x+y)

# Then we calculate the semi-variogrma values
Cr.res.vario <- variogram(Cr.res.gstat, cutoff = 20000, width = 1200)
head(Cr.res.vario)

clean.fit.cr<- fit.variogram(Cr.res.vario, model=vgm(c('Exp','Sph','Gau'), nugget = 29), fit.method = 6)

plot(Cr.res.vario, model = clean.fit.cr, main = "Variogram of Chrome residuals", pch=16,col='black', xlab="Distance",
     ylab="Semivariance", lty = 1, lwd = 3)

trellis.focus("panel",1,1)
llines(x=c(0,max(Cr.res.vario$dist)), y=c(var(clean.donnees.Cr$Cr.res),var(clean.donnees.Cr$Cr.res)),
       col="red", lwd=1, lty=2)
trellis.unfocus()

# Visualisaion des param tres optimis s du mod le
clean.fit.cr




# Pr diction spatiale sur l'ensemble du territoire    tudi    ----

## M thode d terministe : IDW ----

# Cr ation d'une fonction de pr diction pour   viter   les copiers-collers
plot.predictions <- function(prediction,varname,database, plot.title){
  ggplot() +
    geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
    geom_tile(data=prediction,
              aes(x = x, y = y, fill = varname)) +

    geom_point(data=database,
               aes(x=x, y=y, color="Measurement points"),
               shape=18,
               size=1) +
    scale_color_manual("", values="black") +
    scale_fill_gradientn(name="Zn predictions [mg/m ]", colors=c('royalblue','green3', 'yellow', 'red')) +
    theme(legend.key = element_rect(fill = "green3",
                                    color = NA)) +
    xlab("Longitude") +
    ylab("Latitude") +
    ggtitle(plot.title)+
    theme_fivethirtyeight()  +
    theme(axis.title = element_text())
}



### Ni ----

powers <- seq(0.5, 3.5, 0.25)
pmse_Ni <- data.table(power = powers, mse = rep(0,length(powers)) )# Power mean squared error
for (p in powers){
  pse <- rep(0, nrow(clean.donnees.Ni)) # Power squared errors
  for (i in 1:nrow(clean.donnees.Ni)){
    point.idw <- idw(formula = Ni~1,
                     data = clean.donnees.Ni[-i,],
                     locations = ~x+y,
                     newdata = clean.donnees.Ni[i,],
                     idp = p,
                     nmax=20,
                     debug.level = 0) # to avoid getting many output messages
    pse[i] <- (point.idw$var1.pred - clean.donnees.Ni$Ni[i])^2
  }
  pmse_Ni[power==p,"mse"] = mean(pse)
}

plot(pmse_Ni)


Ni.idw <- idw(formula = Ni~1,
              data = clean.donnees.Ni,
              locations = ~x+y,
              newdata = prov.grid,
              idp = 1.25,
              nmax=20)

setnames(Ni.idw, "var1.pred", "Ni.pred")

# We plot the predictions using IDW for Nickel
```

```
plot.predictions(Ni.idw, Ni.idw$Ni.pred, clean.donnees.Ni,
                 "Pr diction des valeurs de Nickel par distance inverse - Hainaut, Belgique")

### Zn ----

# Optimisation of the theta parameter
powers <- seq(0.5, 3.5, 0.25)
pmse_Zn <- data.table(power = powers, mse = rep(0,length(powers)) )# Power mean squared error
for (p in powers){
  pse <- rep(0, nrow(clean.donnees.Zn)) # Power squared errors
  for (i in 1:nrow(clean.donnees.Zn)){
    point.idw <- idw(formula = Zn~1,
                     data = clean.donnees.Zn[-i,],
                     locations = ~x+y,
                     newdata = clean.donnees.Zn[i,],
                     idp = p,
                     nmax=20,
                     debug.level = 0) # to avoid getting many output messages
    pse[i] <- (point.idw$var1.pred - clean.donnees.Zn$Zn[i])^2
  }
  pmse_Zn[power==p,"mse"] = mean(pse)
}

plot(pmse_Zn)


Zn.idw <- idw(formula = Zn~1,
              data = clean.donnees.Zn,
              locations = ~x+y,
              newdata = prov.grid,
              idp = 1.25,
              nmax=20)

setnames(Zn.idw, "var1.pred", "Zn.pred")

# We create a function to plot results for

plot.predictions(Zn.idw, Zn.idw$Zn.pred, clean.donnees.Zn,
                 "Pr diction des valeurs de Zinc par distance inverse - Hainaut, Belgique")

### Cr ----

powers <- seq(0.5, 3.5, 0.25)
pmse_Cr <- data.table(power = powers, mse = rep(0,length(powers)) )# Power mean squared error
for (p in powers){
  pse <- rep(0, nrow(clean.donnees.Cr)) # Power squared errors
  for (i in 1:nrow(clean.donnees.Cr)){
    point.idw <- idw(formula = Cr~1,
                     data = clean.donnees.Cr[-i,],
                     locations = ~x+y,
                     newdata = clean.donnees.Cr[i,],
                     idp = p,
                     nmax=20,
                     debug.level = 0) # to avoid getting many output messages
    pse[i] <- (point.idw$var1.pred - clean.donnees.Cr$Cr[i])^2
  }
  pmse_Cr[power==p,"mse"] = mean(pse)
}

plot(pmse_Cr)



Cr.idw <- idw(formula = Cr~1,
              data = clean.donnees.Cr,
              locations = ~x+y,
              newdata = prov.grid,
              idp = 1.25,
              nmax=20)

setnames(Cr.idw, "var1.pred", "Cr.pred")

# We create a function to plot results for

plot.predictions(Cr.idw, Cr.idw$Cr.pred, clean.donnees.Cr,
                 "Pr diction des valeurs de Chrome par distance inverse - Hainaut, Belgique ")


## Kriegage ----

# In order to realize the krigage prediction :
# we created this new file. Combination of the duplicate loc data
# to get the mean



### Ni ----


Ni.krig <- krige(formula = Ni~1,
                 data = clean.donnees.Ni,
                 locations = ~x+y,
                 newdata = prov.grid,
                 model = clean.fit.ni,
                 nmax=20)



setnames(Ni.krig, c("var1.pred", "var1.var"), c("Ni.predkrig", "Ni.varkrig"))

plot.predictions(Ni.krig, Ni.krig$Ni.predkrig, clean.donnees.Ni,
```

```
                "Prediction de la concentration en Nickel par krigeage - Hainaut, Belgique")

### Zn ----

Zn.krig <- krige(formula = Zn~1,
                 data = clean.donnees.Zn,
                 locations = ~x+y,
                 newdata = prov.grid,
                 model = clean.fit.zn,
                 nmax=20)



setnames(Zn.krig, c("var1.pred", "var1.var"), c("Zn.predkrig", "Zn.varkrig"))

plot.predictions(Zn.krig, Zn.krig$Zn.predkrig, clean.donnees.Zn,
                 "Prediction de la concentration en Zinc avec un krigeage - Hainaut, Belgique")



### Cr ----

Cr.krig <- krige(formula = Cr~1,
                 data = clean.donnees.Cr,
                 locations = ~x+y,
                 newdata = prov.grid,
                 model = clean.fit.cr,
                 nmax=20)



setnames(Cr.krig, c("var1.pred", "var1.var"), c("Cr.predkrig", "Cr.varkrig"))

plot.predictions(Cr.krig, Cr.krig$Cr.predkrig, clean.donnees.Cr,
                 "Prediction de la concentration en Chrome avec un krigeage - Hainaut, Belgique")



## Co-krigage ----

# As the higher correlation is between the Cr and Ni value, we will keep those columns
# We keep a dataset after removing all NA values for Cr and Ni

donnees.cokrig <- Donnees %>%
  drop_na(Cr) %>%
  drop_na(Ni)


# We need to create a gstat object with both data
g <- gstat(id="Ni", formula=Ni~1, data=donnees.cokrig, locations=~x+y)
g <- gstat(g, id="Cr", formula=Cr~1, data=donnees.cokrig, locations=~x+y)

v.cross <- variogram(g, cutoff=14000, width=1000)

plot(v.cross, main="Variograms and Cross-variogram of Nickel and Chrome", ylab="Semivariance", xlab="Distance")

LMC <- fit.lmc(v.cross, g, model=Ni.vg.fit, correct.diagonal=1.0000001)
LMC


g <- copy(LMC)
g <- gstat(g, id="Cr", form=Cr~1, data=clean.donnees.Cr, locations=~x+y, model=LMC$model$Cr)

Ni.cok <- predict(g, prov.grid)

plot.predictions(Ni.cok, Ni.cok$Ni.pred, donnees.cokrig,
                 "Predicted Ni Concentration using Cokriging algorithm with Cr in Hainaut province")

# Display points from both variables


ggplot() +
  geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_tile(data=Ni.cok,
            aes(x=x, y=y, fill=Ni.pred)) +
  geom_point(data=donnees.cokrig,
             aes(x=x, y=y, shape="Chrome data", size="Chrome data"),
             color="black") +
  geom_point(data=donnees.cokrig,
             aes(x=x, y=y, shape="Nickel data", size="Nickel data"),
             color="black") +
  scale_shape_manual("", values=c(1,3)) +
  scale_size_manual("", values=c(1,1)) +
  scale_fill_gradientn(name="Ni prediction [(mg/m?)]", colors=c('royalblue','green3', 'yellow', 'red'))+
  theme(legend.key=element_rect(fill="green3",
                                color=NA)) +
  xlab("Longitude") +
  ylab("Latitude") +
  ggtitle("Nickel cokriging prediction displaying both Nickel and Chrome points")+
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())



## Comparaison between Krigage and Co-krigage ----

# First we define the max and min values of both prediction variances to build
# the same scale on both figures to be able to compare them
Ni_minvar = min(c(Ni.cok$Ni.varcok, Ni.krig$Ni.varkrig))
Ni_maxvar = max(c(Ni.cok$Ni.varcok, Ni.krig$Ni.varkrig))
```

```
# Krigage Variance for Ni prediction


ggplot() +
  geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_tile(data=Ni.krig,
            aes(x = x, y = y, fill = Ni.varkrig)) +
  geom_point(data=donnees.cokrig,
             aes(x=x, y=y, color="Measurement points"),
             shape=18,
             size=1) +
  scale_color_manual("", values="black") +
  scale_fill_gradientn(name="Ni prediction variance [(mg/m )  ]",
                       colors=c('royalblue','green3', 'yellow', 'red'), limits=c(Ni_minvar,Ni_maxvar))+
  theme(legend.key = element_rect(fill = 'green3',
                                  color = NA)) +
  xlab("Longitude") +
  ylab("Latitude") +
  ggtitle("Kriging prediction variance of Ni concentration in Hainaut province")+
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())


# Co-krigage Variance for Ni prediction

ggplot() +
  geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_tile(data=Ni.cok,
            aes(x=x, y=y, fill=cov.Ni.Cr)) +
  geom_point(data=clean.donnees.Cr,
             aes(x=x, y=y, shape="Chrome data", size="Chrome data"),
             color="black") +
  geom_point(data=donnees.cokrig,
             aes(x=x, y=y, shape="Nickel data", size="Nickel data"),
             color="black") +
  scale_shape_manual("", values=c(1,3)) +
  scale_size_manual("", values=c(1,1)) +
  scale_fill_gradientn(name="Ni prediction variance [(mg/m )  ]", colors=c('royalblue','green3', 'yellow', 'red'))+
  theme(legend.key=element_rect(fill="green3",
                                color=NA)) +
  xlab("Longitude") +
  ylab("Latitude") +
  ggtitle("Cokriging prediction variance of Nickel using Chrome")+
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())


# First we define the max and min values of both prediction variances to build
# the same scale on both figures to be able to compare them
vardif <- prov.grid
vardif$vardif <- Ni.cok$Ni.var - Ni.krig$Ni.varkrig

# Variance difference between kriging and co-kriging prediction
ggplot() +
  geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_tile(data=vardif,
            aes(x=x, y=y, fill=vardif)) +
  geom_point(data=clean.donnees.Cr,
             aes(x=x, y=y, shape="Chrome data", size="Chrome data"),
             color="black") +
  geom_point(data=clean.donnees.Ni,
             aes(x=x, y=y, shape="Nickel data", size="Nickel data"),
             color="black") +

  labs(fill="Ni prediction variance difference") +
  scale_fill_gradientn( colors=c('red', 'yellow','green3', 'royalblue'))+
  scale_shape_manual("", values=c(1,3)) +
  scale_size_manual("", values=c(1,1)) +
  theme(legend.key=element_rect(fill='green3', color=NA)) +
  xlab("Longitude") +
  ylab("Latitude") +
  ggtitle("Difference between kriging variance and cokriging variance of Nickel in Hainaut province")+
  theme_fivethirtyeight()  +
  theme(axis.title = element_text())


# Th matique : Carte de risques de hautes concentrations ----

## Ni ----


Ni.condsim <- krige(formula = Ni~1,
                    data = clean.donnees.Ni,
                    loc=~x+y,
                    newdata = prov.grid,
                    model = clean.fit.ni,
                    nsim = 1000,
                    nmax = 20)

ggplot() +
  geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_tile(data=Ni.condsim,
            aes(x = x, y = y, fill = sim2)) + # center title
  geom_point(data=clean.donnees.Ni,
             aes(x=x, y=y, color="Measurement points"),
             shape=18,
             size=1) +
  scale_color_manual("", values="black") +
  scale_fill_gradientn(name="Ni simulation [mg/kg]", colors=c('royalblue','green3', 'yellow', 'red')) +
  theme(legend.key = element_rect(fill = "green3",
                                  color = NA)) +
  xlab("Longitude") +
```

```
  ylab("Latitude") +
  ggtitle("Conditional simulation") +
  theme(axis.title = element_text())

# Find the points where the concentration limit is probably exceeded

Ni.condsim <- as.data.table(Ni.condsim)
ishighN <- Ni.condsim[,-c(1,2)] > 22
riskN <- data.table(x=Ni.condsim$x, y = Ni.condsim$y, Cam = rowSums(ishighN)/1000)

ggplot() +
  geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_point(data = riskN, aes(x=x,y=y)) +
  geom_point(data = riskN[Cam>.75,], aes(x=x,y=y),
             shape = 1,
             color = 'red',
             size = 3) +
  xlab("Longitude") +
  ylab("Latitude") +
  theme(legend.key = element_rect(fill = "green3",
                                  color = NA)) +
  ggtitle("High risk areas") +
  theme(axis.title = element_text())


ggplot() +
  geom_point(data=hainaut.grid, aes(x=x, y=y), color='grey97', shape = 3) +
  geom_tile(data = riskN, aes(x=x,y=y, fill=Cam)) +
  scale_fill_gradientn(name="P(Ni > 22 mg/kg)", colors=c('royalblue','green3', 'yellow', 'red')) +
  xlab("Longitude") +
  ylab("Latitude") +
  theme(legend.key = element_rect(fill = "green3",
                                  color = NA)) +
  ggtitle("High calcium concentration") +
  theme(axis.title = element_text())

# Histogram of location

riskmapN <- colSums(ishighN)/nrow(Ni.condsim)*100
ggplot(mapping = aes(riskmapN)) +
  geom_histogram(color = 'black', bins = 30) + xlab("[%]") +
  ggtitle("Percentage of the surface with Ni > 22mg/kg")

# END
```