Pollution de l'air à Paris

Yanis AMIROU, Gwendal JOUAN, Titouan MORVAN

February 2020

Contents

1 Introduction

Dans le cadre du projet, nous sommes intéressés au problème de la pollution de l'air à Paris. La qualité de l'air est un enjeu de santé publique majeur auquel sont confrontées la plupart des grandes villes. En effet, l'exposition prolongée aux polluants au delà des seuils recommandés entraîne sur le long terme des complications respiratoires et cardiovasculaires.

Il existe deux groupes de polluants : les polluants primaires qui sont émis directement par les sources de pollution - trafic routier, industrie et chauffage - et les polluants secondaires qui proviennent de réactions chimiques entre gaz. Le premier groupe comporte par exemple les oxydes de carbone (CO2), oxydes d'azotes (NOx) et les particules fines (PM10,PM25) tandis que le second comporte l'ozone et le dioxyde d'azote (NO2). La qualité de l'air est le résultat des émissions de polluants d'une part et des conditions de dispersion et transformation de ces polluants d'autre part. C'est donc un phénomène complexe difficile à prédir aussi bien par des modèles physiques que par des modèles statistiques (ou par la combinaison des deux).

Nous avons collecté des données de mesures de polluants, des données météorologiques et des données de trafic routier qui couvrent la période 2014-2018. Dans une premier temps, nous traitons le problème de prévision temporelle, c'est à dire prédire la pollution future à partir de données d'entraînement et des variables explicatives. Dans un second temps nous traitons le problème de prévision spatiale. On cherche à interpoler les mesures des différents stations pour obtenir des cartes de pollution.

2 Présentation des données

2.1 Mesures de pollution

Nous utilisons les données d'Airparif, une association qui s'occupe depuis 1979 de la surveillance de la qualité de l'air en Ile-de-France. Elle dispose d'un réseau d'environ 70 stations réparties dans Paris intra-muros et en Ile-de-France qui mesurent les concentrations de différents polluants : NO2, 03, PM10 et PM25 . Il convient de noter que seule une partie de ces stations sont permanentes et que toutes ne mesurent pas l'ensemble des polluants (cf 1). Elles sont classées en trois catégories suivant leur localisation : urbaine, péri-urbaine et trafic. Les mesures de la plupart des stations sont disponibles en temps réel ainsi que les archives au pas horaire [4].

station	Code postal	Typologie	N02	PM25	PM10	03
Rue Bonaparte	75006	Trafic	✓	✗	✗	✗
Place de l'Opéra	75002	Trafic	✓	✗	✗	✗
Paris Centre	75004	Urbaine	✓	✗	✗	✓
Paris stade Lenglen	75015	Urbaine	✓	✗	✓	✗
Boulevard Périph Est	75012	Trafic	✓	✓	✓	✗

Table 1: Exemples de stations de mesures de qualité de l'air

2.2 Météo

Nous utilisons des données au pas horaire issues de la station Montsouris dans le 14ème arrondissement et obtenues grâce à l'aide de MétéoBlue [2]. Plus exactement, ce sont des données issues de simulations recalculant à posteriori la météo. Les variables météorologiques sont :

- température (°C)
- direction du vent (°)
- vitesse du vent (km/h)
- radiations solaires
- humidité relative (%)

- pression au niveau de la mer (Pa)

- couverture nuageuse (basse, intérmédiaire, haute)

- durée d'ensoleillement (min)

Nous voulions initialement obtenir des données météos à plus haute résolution spatiale (par exemple un quadrillage de Paris). Certaines variables comme la direction/vitesse du vent peuvent en effet présenter des variations locales relativement importantes qui affectent la dispersion des polluants. Nous n'avons malheureusemment pas pu accéder à de telles données dont l'accès est payant.

2.3 Trafic routier

Le trafic routier est l'une des principales sources d'émission de NO2 et c'est donc une variable explicative essentielle. Nous utilisons les données mises à disposition par la Ville de Paris dans le cadre de leur démarche open data [1]. Un réseau de capteurs magnétiques mesure en temps réel le trafic parisien et fournit différents indicateurs :

- **q** : le débit, c'est à dire le nombre de voitures par heure.

- **k** : le taux d'occupation, c'est à dire le temps de présence de véhicules sur la boucle en pourcentage d'un intervalle de temps fixe.

- **etat_trafic** : indicateur calculé à partir de k et q qui caractérise la situation du trafic : 0 Inconnu, 1 Fluide, 2 Pré-saturé, 3 Saturé, 4 Bloqué.

- **etat_barre** : État ouvert ou non (barré, inconnu ou invalide) à la circulation de l'arc

C'est uniquement la connaissance simultanée de k et q qui permet de caractériser le trafic. En effet, le débit dans un bouchon ou de nuit sur une route quasi vide sera faible alors que les deux situations sont très différentes. Néanmoins, il existe une relation forte entre le débit q et le taux d'occupation k qui se traduit par un "diagramme fondamental" dans le jargon de l'ingénierie du trafic routier. Cette relation est dépendante du tronçon considéré mais comporte de manière générale 2 régimes. Dans le premier le débit augmente avec le taux d'occupation (trafic fluide). Au dela d'un certain palier de taux d'occupation, les véhicules commencent à se géner entre eux, on entre alors dans le second régime et le débit diminue avec chaque incrément du taux d'occuption (voir par exemple [6] pour plus de détails).

Nous avons tracé dans la figure 1 les valeurs de q en fonction de celles de k au niveau du quai des célestins mesurées entre Janvier 2015 et Janvier 2020. Ici, la relation $q - k$ peut être correctement approchée par une simple fonction affine par morceaux et on voit clairement le changement de régime qui se fait aux alentours d'un taux d'occupation de 16%

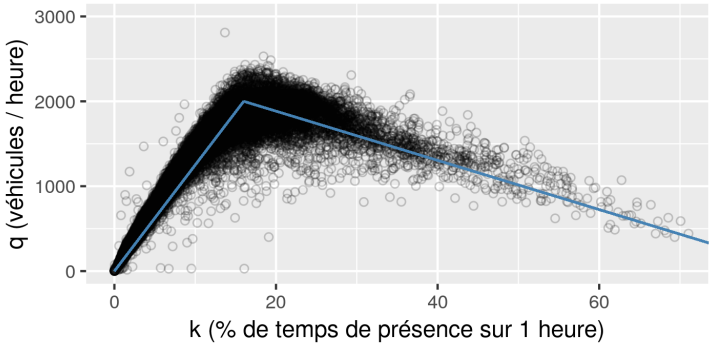


Figure 1: Relation q-k pour le quai des célestins

Le réseau couvre bien l'ensemble de la ville (plus de 5000 bornes) et pour chaque station de pollution parisienne, un capteur est présent dans la même rue ou dans une rue voisine. Les informations géographiques [5] sont disponibles au format Shapefile et l'on peut facilement en tirer des cartes à l'aide

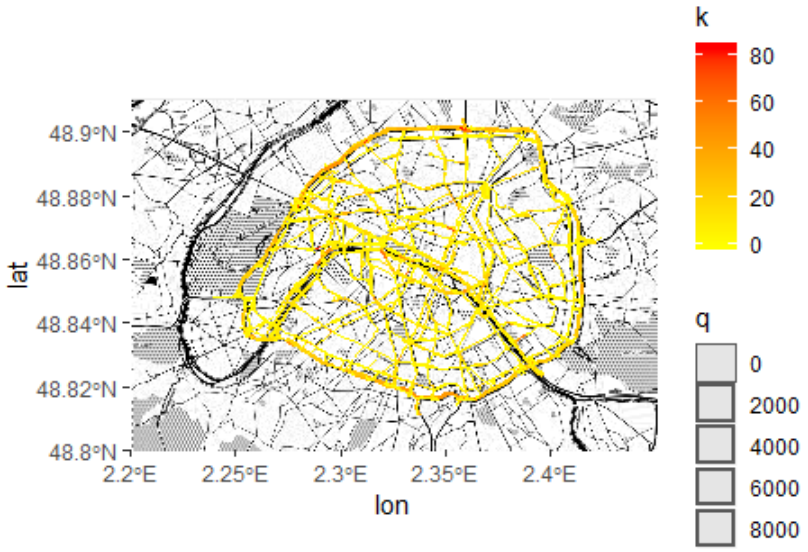


Figure 2: Trafic routier à 20h le 01-02-2017

du package sf de R . Dans la carte ci-dessous la couleur indique la fluidité du trafic, jaune fluide et rouge congestionné, et l'épaisseur des axes est proportionnelle au débit.

Remarque : Les données historiques sont disponibles depuis 2014 et la base de donnée est par conséquent très volumineuse, plus de 30Go, et longue à manipuler. Bien que ne nous n'avions besoin des données historiques que pour une vingtaine de rues (correspondant à nos capteurs), nous avons dû tout télécharger ...

2.4 Traitement des données

Nous avons utilisé le package ImputeTS pour visualiser les données manquantes dans nos séries. On peut découper la série en batches et visualiser le pourcentage de NA dans chaque batch. On peut aussi obtenir la répartition des périodes sans données, c'est à dire nombre de périodes de longueurs l manquantes.

Les valeurs manquantes isolées ne posent pas tant problème car il est facile de les compléter fidèlement par interpolation. Le problème est en revanche nettement plus ardu lorsqu'il manque un jour, une semaine entière, voire plus et ce n'est malheureusement pas si rare quand on travaille avec des capteurs physiques qui peuvent tomber en panne.

2.4.1 Météo

Les données proviennent d'un site à visée commerciale et sont donc très propres. Nous n'avons pas eu de traitement particulier à effectuer.

2.4.2 Trafic

C'est tout bon ou tout mauvais. Certaines rues sont inexploitables probablement à cause de travaux ou de pannes de capteurs de longue durée (réparer un capteur magnétique sous la chaussée ne doit pas être simple...). Lorsque que les données sont disponibles sur toute la période, elles sont plutôt propres avec un taux de données manquants dépassant rarement les 2-3% et sur des périodes assez courtes.

Nous avons ici décidé de compléter les données manquantes à l'aide d'un simple modèle linéaire incluant comme variables explicatives la date l'heure de la journée, le jour de la semaine ainsi que la température et les précipitations provenant des données météo. Inclure la date ici doit permetre de capturer une tendance globale. Idéalement on pourrait également inclure des informations du calendrier tels que les jours feriers ou les vacances mais cela n'a pas été fait ici faute de temps. La variation des grandeurs de trafic ("q"

et "k") étant fortement périodique, l'heure de la journée est prise en compte via une transformation par les 5 premiers termes de la série de fourier de période 24 heures. Les variables basées sur l'heure h sont donc $cos(k2\pi h/24)$ et $sin(k2\pi h/24)$ avec $k = 1, ..., 5$. Au final, toutes les variables inclues dans le modèles linéaires sont significatives pour la prédictions de q et k et on obtient une adéquation relativement bonne entre les données et le modèle (voir figure 2.4.2 pour le cas de q sur une semaine)

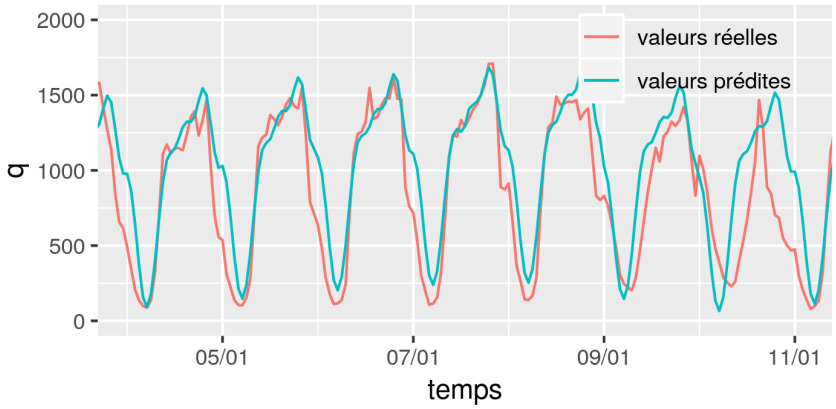


Figure 3: Valeurs de q sur une semaine

Le modèle ainsi calibré est ensuite utilisé pour compléter les valeurs de q et k manquantes.

2.4.3 Pollution

Ce sont les données de pollution qui nous ont posé le plus problème. Suivant les stations le taux de données manquantes est de 3 à 10%, ce qui est relativement acceptable. En revanche, des stations présentent des trous d'une semaine et dans de rares cas un mois entier. Remplacer par la moyenne ou par la données présente la plus proche est donc impensable dans de tels cas. Les techniques d'interpolation linéaires ne fournissent pas des résultats très réalistes non plus car elles ne respectent pas les cycles journaliers.

Nous avons tout d'abord séléctionné la station la plus propre pour effectuer nos prédictions puis nous avons complété les données manquantes non pas par la moyenne sur la base mais par la moyenne des observations à la même heure afin de respecter le cycle journalier.

2.5 Statistiques descriptives

On présente ici les statistiques descriptives de la concentration en N02 à la station Quai des Célestins dans le 4ème arrondissement.

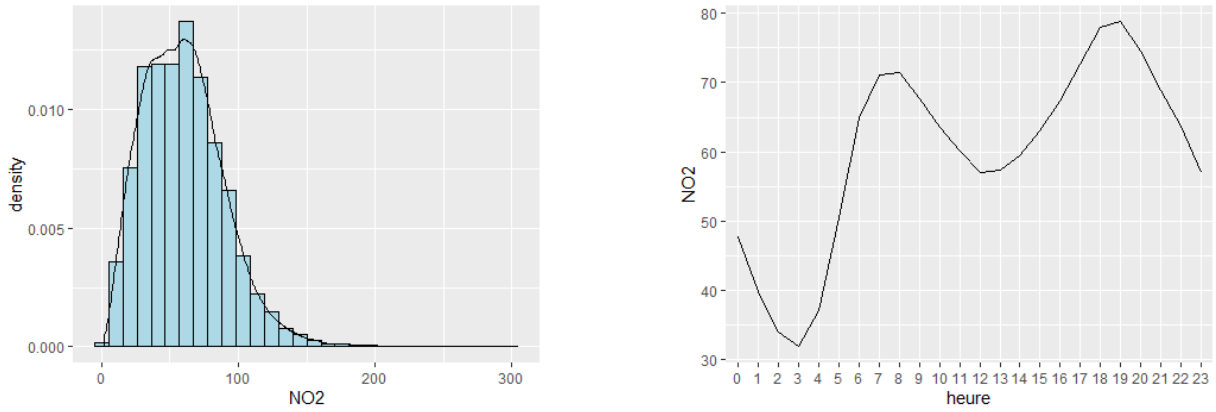


Figure 5: Moyennes journalières.

L'histogramme est assez plat autour de la moyenne égale à 60 $\mu g/m^3$ et présente une décroissance exponentielle au delà d'un seuil.

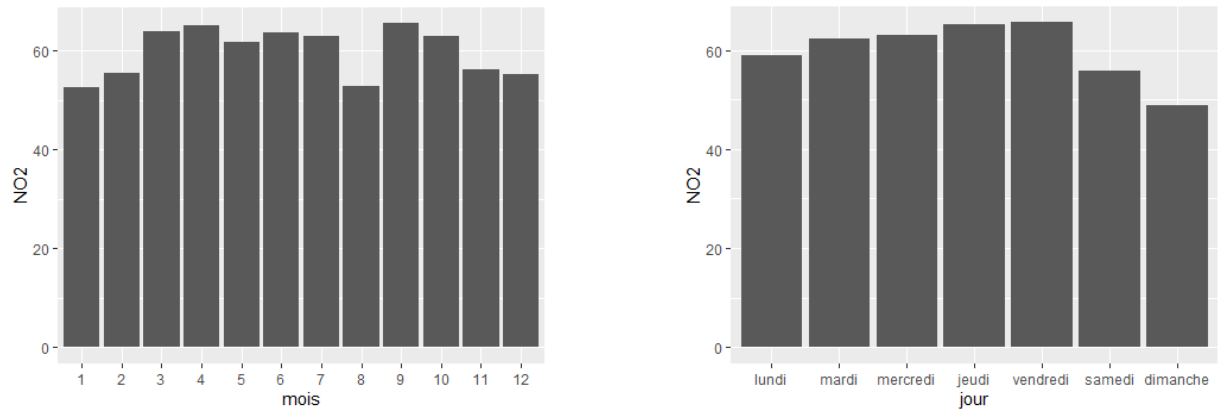
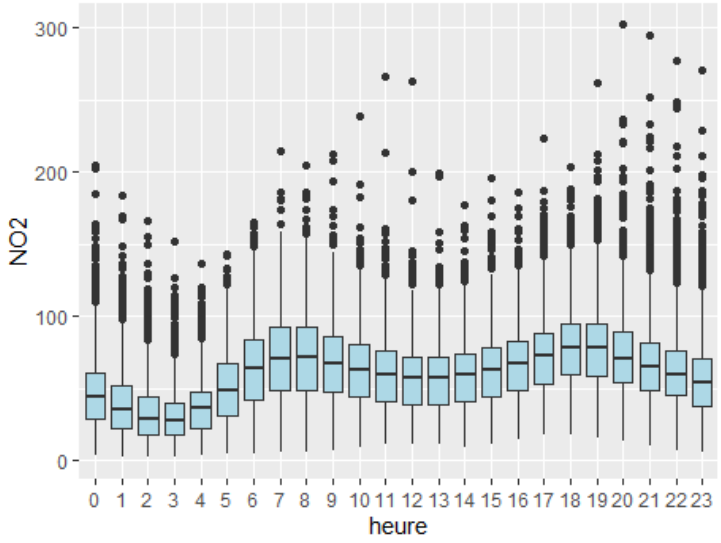


Figure 7: Moyennes journalières.

On observe que le niveau de polluants suit assez bien l'activité humaine. Le profil journalier présente deux pics : l'un autour de 7-8h correspondant au départ au travail et l'autre autour de 18-19h correspondant au retour. Le profil hebdomadaire est quasi constant en semaine avec une diminution le weekend. On observe également l'effet de la faible activité du mois d'août. Il est par ailleurs assez instructif de consulter les mesures lors des journées "Paris sans voitures" : on observe une diminution assez conséquente de la pollution ! (pour une analyse plus détailée voir [3])



On observe sur le boxplot par heure une très forte dispersion avec des pics de pollution assez nombreux jusqu'à 6 fois la valeur moyenne. On se doute bien que prédire ces pics avec précision ne va pas être simple.

En ce qui concerne l'évolution globale des mesures de NO_2 on observe une tendance à la baisse significative (p-value ¡ 0.01) de $-1,47 \mu g/m^3$ par an en se basant sur un modèle linéaire simple entre le niveau de pollution mesuré et le temps (bien que cette tendance ne soit pas forcément évidente à discerner "à l'oeil nu" à partir du graphique cf figure 2.5).

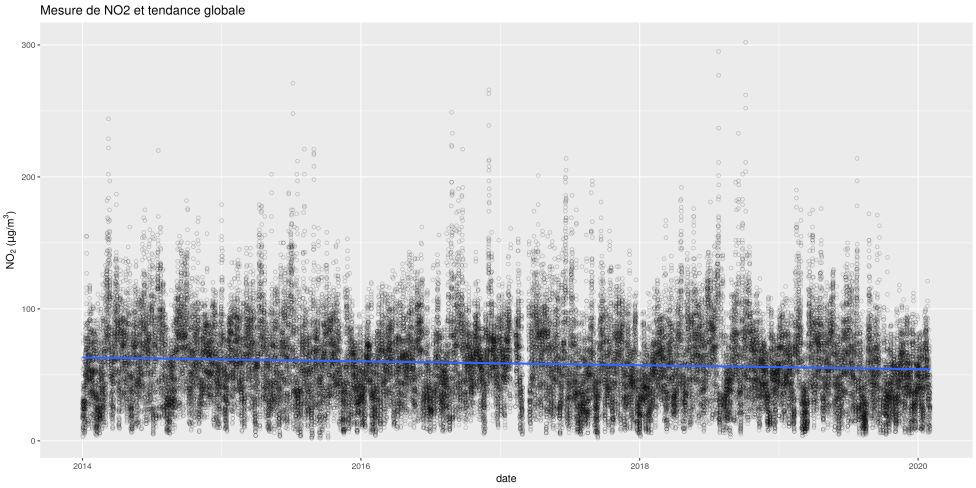


Figure 8: Tendances globales de la concentration de NO_2 mesurée

3 Prédictions temporelles : dépassement de seuil

L'intérêt principal d'un modèle de prédiction temporelles (à court terme) est de pouvoir alerter d'un dépassement probable d'un seuil de pollution, par exemple la veille pour le lendemain. Dans ce cas le modèle doit être construit avec des variables connues à l'instant $t-1$ pour la prédiction à l'instant t. Un tel modèle peut par exemple inclure les prévisions météos, des données du calendrier (jour de la semaine, mois de l'année etc.) mais ne peut évidément pas inclure les données mesurés à l'instant t pour lequel on veut faire la prédiction (par exemple les différentes mesures du trafic routier).

Dans cette section on esssaie de construire un modèle prédisant la veille pour le lendemain un dépassement de seuil du taux horaire de NO_2. Le seuil maximal recommandé par l'OMS est de $200\mu g/m^3$ (en moyenne horaire). Les dépassement de seuils pour cette limite étant relativement rare, nous avons décidé d'abaisser ce seuil $180\mu g/m^3$ afin que le problème de prédiction soit plus abordable. Sur la période allant de Janvier 2014 à Janvier 2020 il y eu 44 cas de dépassement de ce taux horaire (on compte ici au maximum un seul dépassement par journée : si pendant la journée ce seuil a été dépassé sur plusieurs heures on ne compte dans tous les cas qu'une seule occurence). On donne dans les figures ci-dessous le nombre de cas de dépassement (sur la période Janvier 2014 à Janvier 2020) en fonction du mois de l'année et de la température maximale pendant la journée

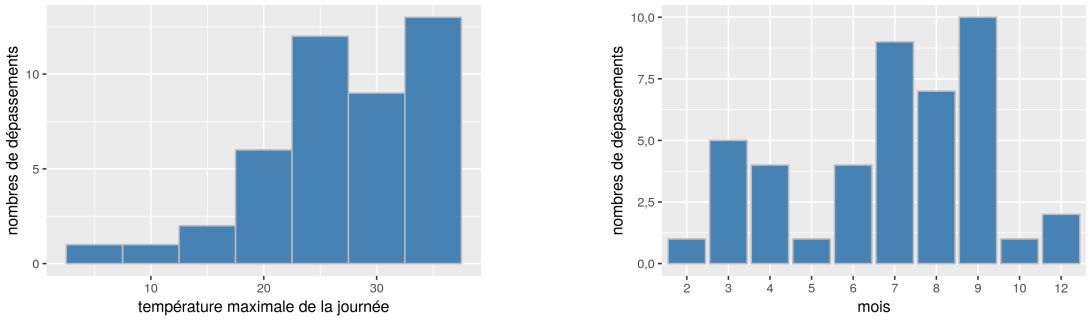


Figure 10: Cas de dépassement de seuils sur la période Janvier 2014 à Janvier 2020

Pour prédire s'il y a un risque de dépassement au jour J, les données que l'on considérera ici sont les données météo du jour J lui même, et plus précisément les valeurs maximum, moyenne et minimale sur la journée des données horaires de la base de donnée météo (voir section 2.2). On peut en effet considérer que ces données sont connues avec une bonne précision la veille pour le lendeman à l'aide des prédictions météorologique. Idéalement on baserait le modèle sur les prédictions qui ont effectivément était produites la veille mais pour des raisons commodité on s'en tient ici aux valeurs mesurées le jour J. Le temps de

résidence du NO_2 étant d'environ 24h on inclut également les taux maximums minimums et moyens du NO_2 la veille. On intègre également les données du calendrier : jour de la semaine et semaine de l'année.

3.1 Prédictions de la valeur maximale

Une première approche pour prédire les cas de dépassement de seuil peut être de tenter de prédire non pas si oui ou non il y a dépassement (problème de classification) mais plutôt le taux maximum de NO_2 dans la journée via une régression (linéaire ou autre). On a donc d'abord tenté de construire un simple modèle linéaire liant le taux NO_2 maximum de la journée aux données disponibles. Etant donné le nombre relativement important de données (35 variables explicatives par jours). nous avons opté pour le modèle LASSO pour permettre une certaine sélection de variable. Le modèle est calibré sur un jeu de données d'entrainement de 1186 observations (sur 2186 observations totales). Le facteur de pénalisation est ici choisi de façon à minimiser l'erreur MSE de validation croisé (via la fonction `cv.glmnet()`.

En fait on se rend vite compte que le modèle ainsi obtenu sera incapable de prédire les cas de dépassements. En effet on observe sur la figure 11, qui représente les valeurs prédites et les valeurs mesurées en fonction de la date, que le modèle n'arrive pas à "capter" les pics, et celà même sur les données d'entrainement. La raison étant probablement que ceux ci sont trop rares.

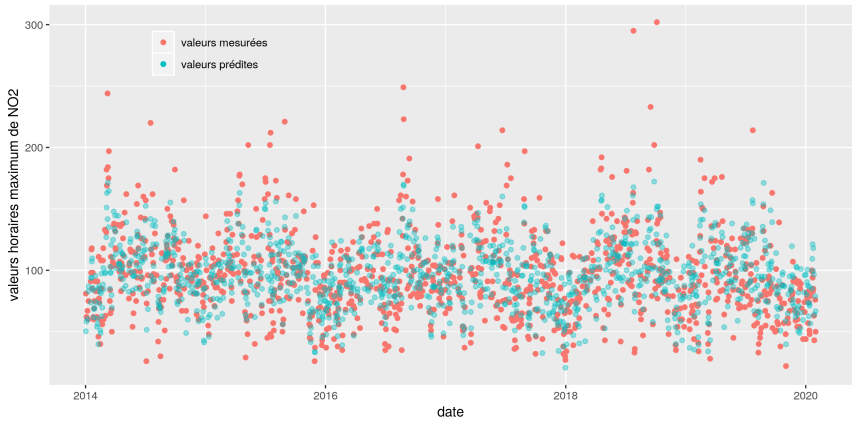


Figure 11: Mesures contre modèle pour le LASSO (données d'entrainements)

Nous avons également tenté de prédire le taux maximum de NO_2 avec une approche de type forêt aléatoire. Ce type de modèle s'en sort ici beaucoup mieux que le LASSO puisqu'on arrive effectivement à prédire certains cas de dépassement de seuil sur les données test (voir figure 13). Les résultats ne sont cependant pas réellement satisfaisant. En considérant ici que le modèle a fait une prédiction correcte s'il prédit un taux de NO_2 supérieur à la limite pour les jours ou il y eu effectivement dépassement de seuil (on ne s'intéresse pas au fait que la valeur réelle soit proche ou non de la valeur réelle, seulement s'il y'a eu effectivement dépassement ou non) on obtient, sur un échantillons de 1000 observations (16 cas de dépassements), 3 prédictions correctes, 13 faux négatifs et 2 faux positifs.

3.2 Classification : dépassement de seuil ou non

La deuxième approche consiste à effectivement aborder ce problème comme un problème de classification (dépassement ou non du seuil), par exemple en utilisant une régression logistique avec toujours une pénalisation L1. Cette approche est cependant elle aussi destinée à échouée si l'on se base sur les échantillons "bruts" des données d'entrainement et sur la fonction de perte logistique. En effet les cas de dépassement de seuil étant rares, la perte est minimisée pour un modèle qui n'en prédit jamais. Pour palier à ce problème on peut considérer d'autres types de fonctions de pertes ou "dupliquer" les cas de dépassement de seuils dans l'échantillon d'entrainement. Nous avons tenter cette deuxième approche en prenant les poids selon la formule : 1− (nbre d'élèments dans la classe) /(nbre total d'élèments). Après "rééquilbrage" des échantillons, on effectue donc une régression logistique avec pénalisation L1 (et facteur de pénalisation déterminé par validation croisée sur les données d'entrainement). Ce modèle est cependant décevant puisque pour un nombre total de 1000 observations sur l'échantillons test, et en ne retenant que

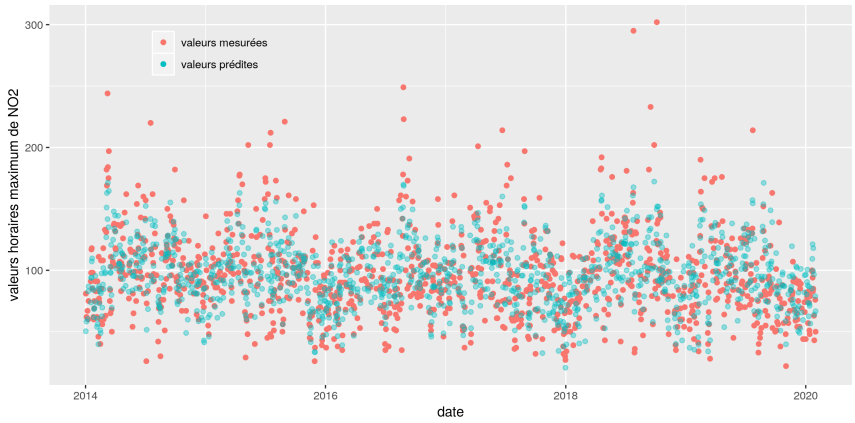


Figure 12: Mesures contre prédictions pour la forêt aléatoire (données test)

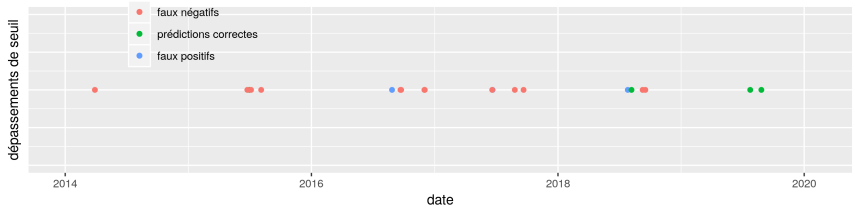


Figure 13: Détections, faux positifs et faux négatifs pour la forêt aléatoire (données test)

les observations dont la probabilité de dépassement est supérieure à 0,5 on obtient 10 prédictions correctes, 6 faux négatifs et 80 faux positifs.

Nous avons également utilisé un modèle de forêt aléatoire pour la classification (sans rééchantillonage, l'algorithme fonctionnant bizarrement mieux sans que avec). On obtient dans ce cas 3 prédictions correctes, 6 faux positifs et 13 faux négatifs sur l'échantillon test. Un taux de faux postifs qui est donc bien plus faible mais au prix d'un nombre de prédictions correctes plus faible également. Nous avons aussi testé le boosting sur des arbres de classification (package `xgBoost`) qui nous donne 4 prédictions correctes 9 faux positifs et 12 faux négatifs.

3.3 Conclusion

Nous avons traité dans cette section le cas de la prédiction de dépassement de seuils d'un polluant à 24h. La première difficulté étant que s'agissant de prédictions, nous avons exclut les données de trafics automobiles dont on sait qu'elles sont fortement corrélés avec les taux mesurés de NO_2. En ne retenant que les données météos et calendaire, il n'est pas sûr que nous avions les données nécessaire pour attaquer le problème. La deuxième difficulté vient du fait que les cas de dépassement de seuils sont relativement rares. Un tel problème doit donc être traité dans un cadre particulier. Nous avons malheureusement pas pu étudier la littérature traitant de la détection d'"évenements rares". Il existe a priori beaucoup d'approche pour lever les difficultés que l'on a rencontrées ici et il aurait probablement fallu les creuser un peu plus...

4 Estimation du taux de NO2

Ici nous nous attaquons à un problème plus modeste, qui est la prévision du taux de NO_2 à l'instant t en fonction de toutes les variables connues à ce même instant t (mesures météo, trafic etc.). L'intérêt de tels modèles peut être par exemple la complétion de données manquantes sur une station, ou bien l'estimation du taux de NO_2 à un endroit ou il n'existe pas de mesure (à condition que les modèles calibrés soit généralisable). Ce dernier cas peut également être traité dans le cadre de l'interpolation spatiale comme nous le verrons dans la section suivante). Les modèles statistiques peuvent aussi dans certains cas (modèles linéaires par exemple) informer sur l'importance que peuvent jouer les différentes variables (trafic, météo etc) sur la pollution quand les phénomènes physiques sont mals compris ou trop complexes.

4.1 Modèles linéaires, LASSO et GAM

4.1.1 Modèle linéaire

La principale difficulté ici pour construire un modèle linéaire qui généralise correctement est la sélection de variables. On se doute que parmis les variables recueillies dans nos données certaines sont fortement corrélées entres elles. On a vu par exemple précédement que les grandeurs liées aux trafic sont très corrélées à l'heure de la journée (voir la section 2.4.2), il faudra donc intégrer l'une ou l'autre de ces grandeurs. Dans ce cas particulier on peut estimer que l'heure de la journée n'est pertinente que dans la mesure où c'est une variable "proxy" pour l'activité humaine en général et pour le trafic routier en particulier. Si cette activité peut-être mesurée plus directement (comme avec les mesures de trafic) alors on peut espérer une meilleur généralisation du modèle en intégrant ces variables à la place de l'heure.

On donne ci dessous (figure 4.1.1 les corrélations entre concentration de NO_2 mesurés et les 3 variables de trafic : **k**, **q** et **etat_trafic**. Ces graphiques semblent bien faire apparaitre des tendances mais restent difficiles à interpréter directement. En particulier la variable **etat_trafic** qui est un indicateur à 5 niveaux du trafic (inconnu, fluide, pré saturé, saturé et bloqué) n'est que moyennement informatif pris seul. La médiane des concentrations mesurés est sensiblement plus faible pour un trafic fluide mais les valeurs extrèmes excèdent celles des états saturé et bloqué.

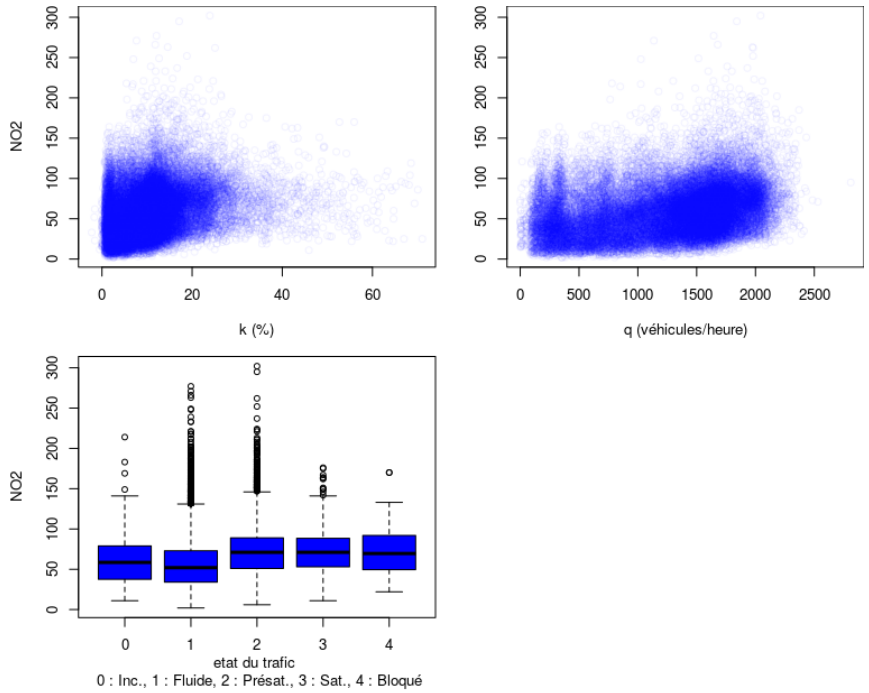


Figure 14: corrélations entres variables de trafic et concentration de NO_2

En ce qui concerne les grandeurs météos, leurs effets sur le taux de NO_2 mesuré et les éventuelles corrélations entre elles sont moins évidentes. On donne ci dessous (figure 15) le taux de NO_2 en fonctions

des principales grandeurs météo mesurées. A "vu d'oeil", seules la vitesse du vent et peut-être le niveau de précipitation ont une corrélation claire avec la concentration de NO_2. La température semble jouer à partir d'un seuil d'environ $25C°$.

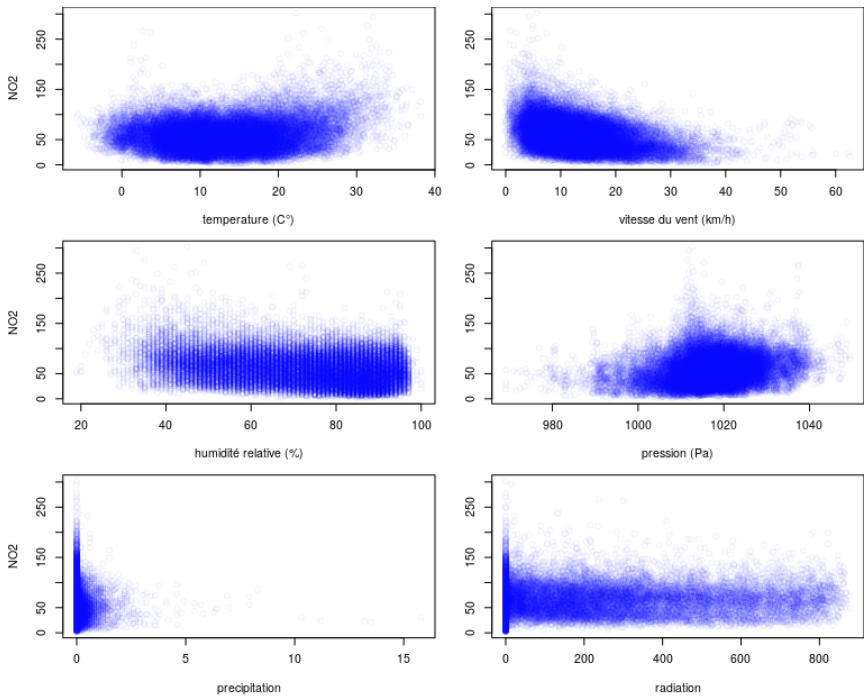


Figure 15: corrélations entres variables météo et concentration de NO_2

De cette analyse qualitative on retient comme variables explicatives pour notre modèle linéaire les grandeurs suivantes :

1. le temps : c'est à dire la date (codé en POSIXct). Cela doit permetre de prendre en compte la tendance globale mise en évidence dans la section 2.5.
2. `k` : le taux d'occupation, avec deux niveaux d'expression : l'un pour les valeurs en dessous de 16% l'autre pour les valeurs au dessus de ce seuil.
3. `k_tronc` qui sont les valeurs k au dessus du seuil de 16%. On espère ainsi capter la différence de régime (cf section 2.3).
4. `Temp_tronc` : la température, avec un seuil à $25C°$ (les valeurs plus faibles sont ignorées).
5. `Temp_tronc` la vitesse du vent.
6. `precipitation` : les précipitations.

Toutes les autres variables sont ignorées. On donne dans la figure 16 l'adéquation entre le modèle et les données d'entrainement sur une semaine en Janvier 2015. Le modèle reproduit qualitativement les tendances mais reste très grossier. En particulier il ne semble pas capturer correctement les pics.
Sur les figures suivantes (17 et 18) on donne cette fois ci la prédiction contre les valeurs mesurées (sur les données tests) sur 2 semaines différentes. On voit ici que le modèle peut reproduire qualitativement la tendance des taux observés à certains moment mais être complètement en dehors sur d'autres moment. On note aussi la tendance à sous estimer les pics qu'on observait déjà sur la comparaison avec les données d'entrainement.

On obtient un MAPE de 40.75 et un RMSE de 27. Les coefficients du modèles linéaire et leur p-values sont donnés dans le tableau 2.

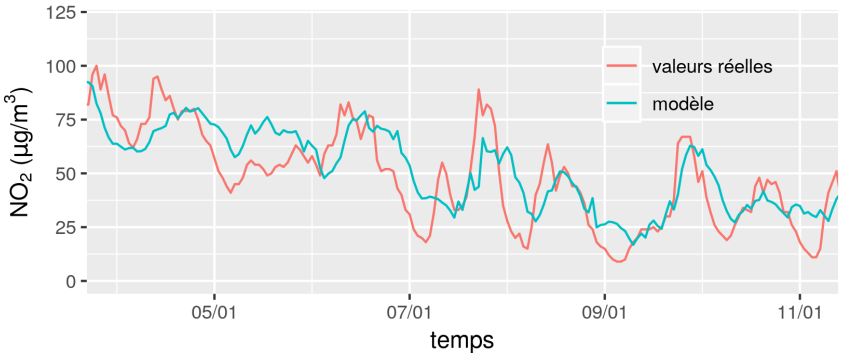


Figure 16: Comparaison données d'entrainement - modèle

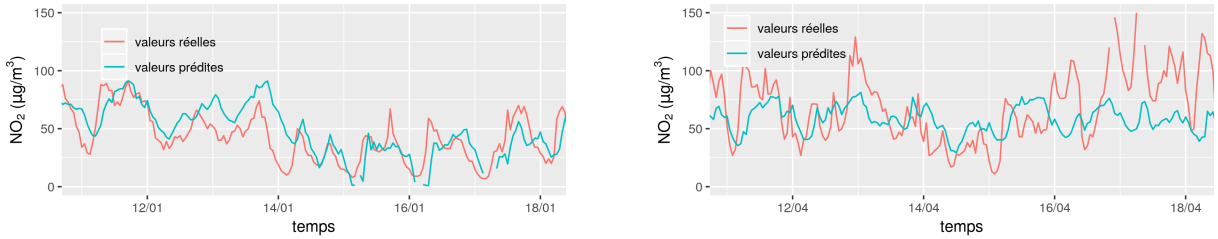


Figure 18: Mesures contre prévisions

	Estimate	Std. Error	t value	Pr(> |t|)	
(Intercept)	5,748e+02	3,437e+01	16,723	< 2e-16	***
temps	-3,503e-07	2,395e-08	-14,630	< 2e-16	***
k	2,237e+00	6,553e-02	34,144	< 2e-16	***
k_tronc	-2,101e+00	1,305e-01	-16,104	< 2e-16	***
temperature	-3,157e-01	4,355e-02	-7,250	4,47e-13	***
Temp_tronc	6,153e+00	2,995e-01	20,542	< 2e-16	***
windspeed	-1,830e+00	3,407e-02	-53,701	< 2e-16	***
precipitation	7,796e-01	9,543e-01	0,817	0,414	

Table 2: Résumé du modèle linéaire

4.1.2 Régression avec LASSO

L'alternative à la sélection de variable "à la main" telle qu'elle a été faite dans le paragraphe précédent, est la sélection de modèle via une régression de type LASSO bien qu'une telle méthode ne se justifie que moyennement ici étant donné le petit nombre de variables.

On entraine donc un modèle LASSO en intégrant l'ensemble des données météo et de trafic disponibles (y compris les variables "seuillées" `k_tronc` et `Temp_tronc`). Le choix du paramètre de pénalisation λ est réalisé à l'aide de la fonction `cv.glmnet()` du package `glmnet` qui le détermine en minimisant l'erreur mse de validation croisée (10 sous échantillons). Au final, cette approche ne permet pas de sélectionner de variables puisque les coefficient estimé sont tous non nuls (voir tableau 3).

La figure 19 donne l'adéquation du modèle aux données d'entrainement. Les figures 20 et 21 donne les prévision contre les valeurs mesurées sur 2 semaines de l'échantillon test. On obtient un MAPE de 53,2 et un RMSE de 26. Le modèle LASSO ne fait donc pas mieux que le modèle linéaire sans pénalisation.

4.1.3 GAM

On fait toujours l'hypothèse que les effets s'ajoutent mais cette fois cette relation n'est pas forcément linéaire. Nous expérimentons à présent l'algorithme GAM. Nous utilisons une base de splines cubique cyclique avec 7 degrés de libertés sur les variables temporels (jours, semaines, mois) et la base de splines

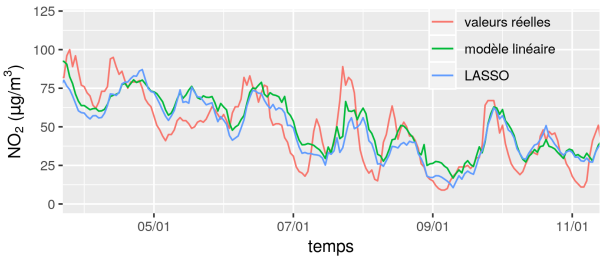


Figure 19: Comparaison données d'entrainement - modèle

(Intercept)	-1,577195e+02
k	-1,710272e+00
q	2,706268e-02
relativehumidity	-2,372254e-01
sl_pressure	2,386715e-01
precipitation	5,325706e+00
shortwave_radiation	-4,423804e-03
windspeed	-1,745061e+00
Temp_tronc	4,471661e+00
k_tronc	2,724846e+00

Table 3: Valeurs des coefficients pour le LASSO

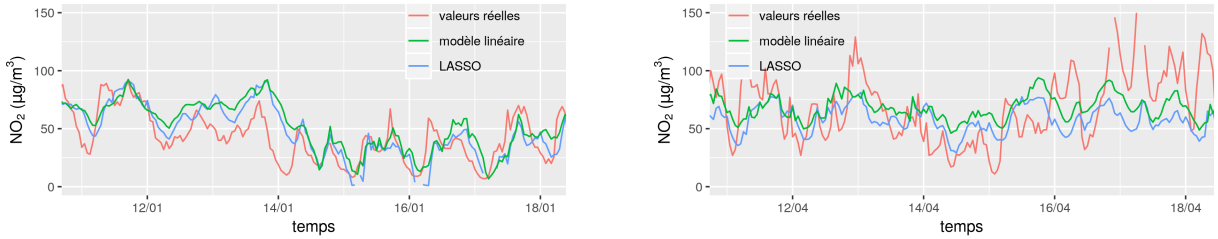


Figure 21: Mesures contre prévisions sur deux semaines différentes

régréssions cubiques avec 4 degrés de liberté sur les autres variables. Le choix des bases de splines est guidé par l'intuition donnée par les graphes de l'émission de NO2 en fonction de chaque variable.

Nous avons entrainé le modèle sur les données du premier Janvier 2014 à la semaine s-1.

```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  60.1032     0.1109     542   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                edf Ref.df       F  p-value
s(hour)       5.000  5.000 1910.45  < 2e-16 ***
s(k)          2.935  2.995   26.46  < 2e-16 ***
s(q)          2.985  2.999   49.40  < 2e-16 ***
s(windspeed)  2.880  2.988 2699.96  < 2e-16 ***
s(winddirection) 2.999  3.000 1762.48  < 2e-16 ***
s(mois)       4.978  5.000   12.47 1.03e-12 ***
s(week)       4.748  5.000   14.00 1.72e-14 ***
s(jour)       4.976  5.000  313.92  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.512   Deviance explained = 51.3%
GCV = 431.47  Scale est. = 431.07    n = 35061
```

Figure 22: Résumé du modèle GAM entrainé sur les données 2014-2017

Les scores obtenus sont:
Pour l'année 2018, RMSE = 22 et MAPE = 37.46
Pour la première semaine de 2018, RMSE= 17 et MAPE=49.7
Pour la semaine du 17 au 24 Décembre 2018, RMSE= 13 et MAPE = 34.73

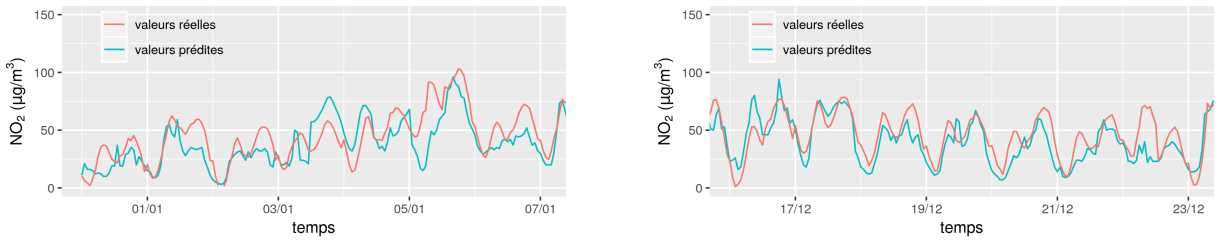


Figure 24: Mesures contre prévisions sur deux semaines différentes

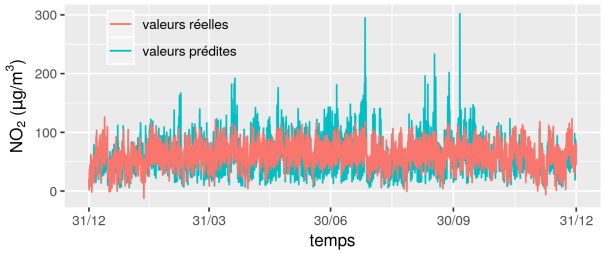


Figure 25: Mesure contre prévision du GAM sur l'année 2018

4.2 Réseaux de neurones

Dans cette section, nous testons un réseaux de neurones classique, le perceptron multi-couche. L'architecture que nous avons utilisé est un model séquentiel constitué de 6 couches cachées avec respectivement 20,40,60,80,100,120 neurones. Nous avons utilisé des fonctions d'activations 'ReLu' afin d'augmenter progressivement la non-linéarité du modèle. De plus, nous avons inclus une régularisation L1 (LASSO) de 0.001 sur chaque couche afin de réduire la compléxité du modèle et éviter le sur-apprentissage. La régularisation L1 sur chaque couche cachée permet également de séléctioner les variables les plus pertinentes en entrée et les neurones pertinent sur chaque couche. C'est l'architecture qui semble avoir les meilleurs prédictions parmis plusieurs autres que nous avons testés. Nous avons entrainé le modèle sur les données du premier Janvier 2014 à la semaine s-1. Les scores obtenues sont:
Pour l'année 2018, RMSE = 19.26 et MAPE = 34.95
Pour la première semaine de 2018, RMSE= 19.73 et MAPE= 57.15
Pour la semaine du 17 au 24 Décembre 2018, RMSE= 12.26 et MAPE = 30.07
L'un des points faibles des réseaux de neurones est leur interprétabilité. Ces modèles sont d'ailleur appelé boite noire. Ceci est dû au fait qu'il est difficile de comprendre le role joué par chaque couche du réseaux et l'importance de chaque variable d'entrée. Notons que les scores obtenus changent sensiblement en fonction de l'architecture du réseaux de neurones et il n'y a pas de méthode générale pour trouver la meilleure architecture. Ces résultats sont donc sans doute à améliorer en travaillant sur l'architecture du réseaux de neurones. Il est interessant de tester une architecture du type réseaux de neurones récurents. En particulier, les réseaux du type LSTM permettent de prendre en compte les taux de pollution passés dans la prédiction.

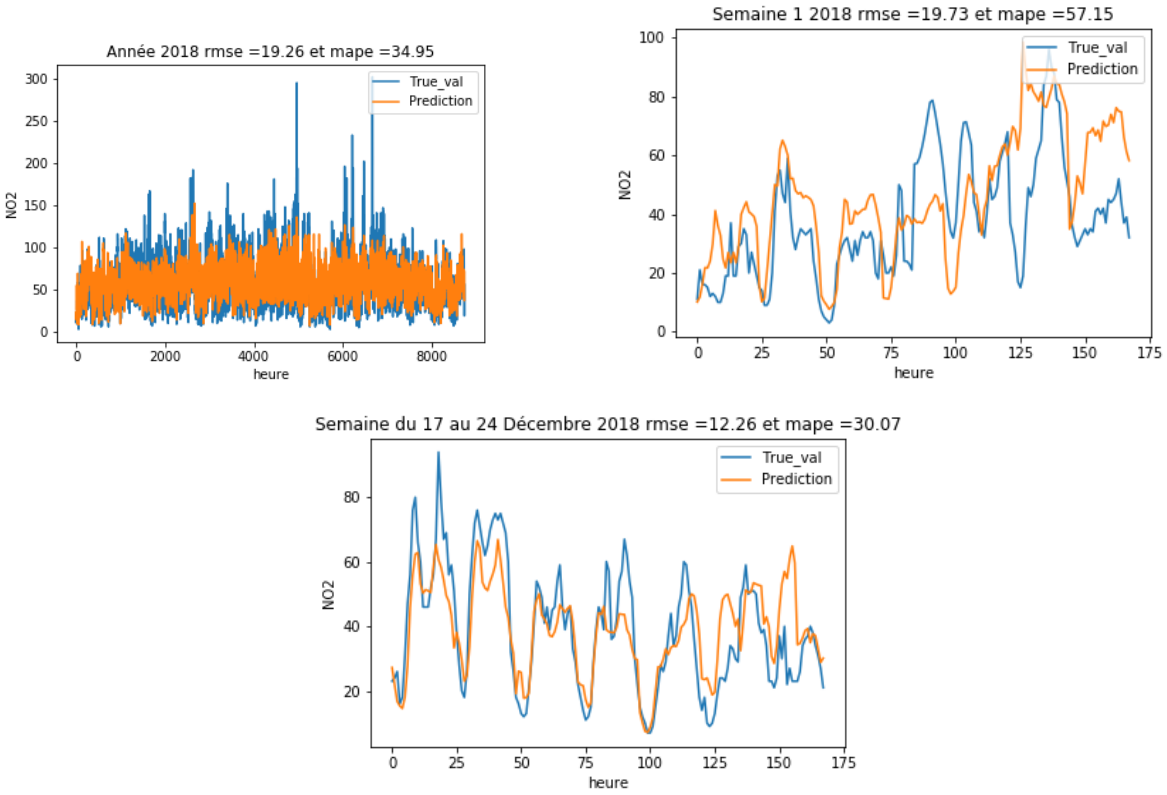


Figure 28: Mesure contre prévision du réseaux de neurones

4.3 Modèles d'ensemble

On teste dans cette section les modèles d'ensemble : arbres de régression, baggin et forêts aléatoires. On utilise les variables explicatives : heure, humidité relative, radiations solaires, température, direction/vitesse du vent, débit du trafic et taux d'occupation (l'influence de la pression et des précipitions étaient négligeables).

4.3.1 Arbre de régression

On commence par les arbres de régressions qui, bien que peu efficaces, ont le mérite d'être facilement interprétables. On utilise la fonction par défaut du package **tree** et on obtient un RMSE de 25 et un MAPE de 44.7. Les prévisions sont très mauvaises et rendent très grossièrement le cycle journalier. On remarque de manière étonnante que les variables de trafic ne sont pas retenues dans l'arbre par défaut (heure, vitesse et direction du vent). Par contre si l'on retire l'heure des variables explicatives, les données de trafic redeviennent prépondérantes.

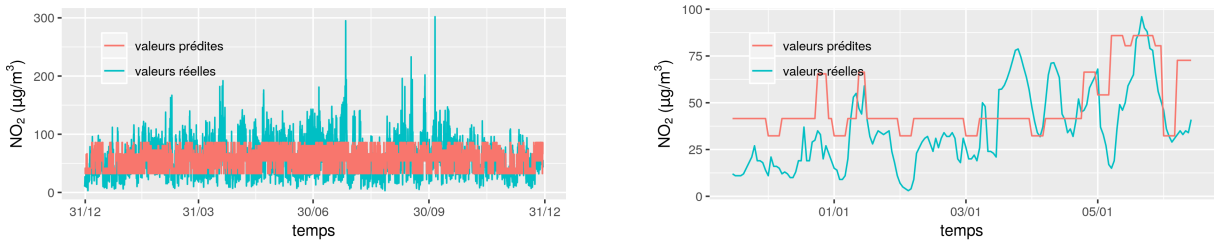


Figure 30: Mesures contre prévisions de l'arbre de régression la première semaine

4.3.2 Bagging

On utilise le bagging qui consiste à entraîner des bases learners (ici des arbres de régression) sur des sous-échantillons obtenus par bootsrap. Après étude des paramètres, on retient **nbag** = 50 arbres entraînés avec des échantillon de taille **size** = 300. On obtient un RMSE de 22 et un MAPE de 40.9. C'est une légère amélioration par rapport aux arbres de régressions mais il reste encore du travail. En effet, on observe que la forme du cycle journalier est plutôt bien rendue mais que la modèle suit difficilement lors des pics ou des chutes (en fait dès que la concentration dévie du cycle journalier moyen).

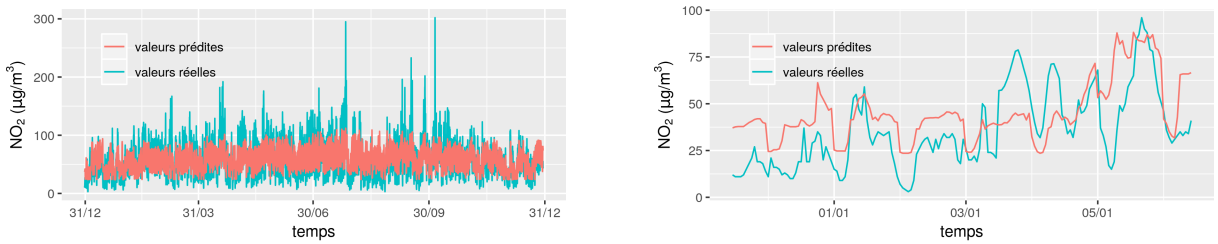


Figure 32: Mesures contre prévisions par bagging la première semaine

4.3.3 Forêts aléatoires

On passe aux forêts aléatoires qui sont reconnues commme une méhode efficace dans la littérature sur la prédiction de la pollution de l'air. On utilise le package R **RandomForest**. On retient les paramètres suivants : le nombre d'arbres **ntree** = 50, le nombre de variable à tester à chaque coupe **mtry** = 4 et la taille de l'échantillon pour entraîner chaque arbre sampsize **sampsize** = 10000.

On obtient un RMSE de 21 et un MAPE de 34.2. Hormis les réseaux de neurones, c'est la méthode la plus efficace jusqu'à présent.

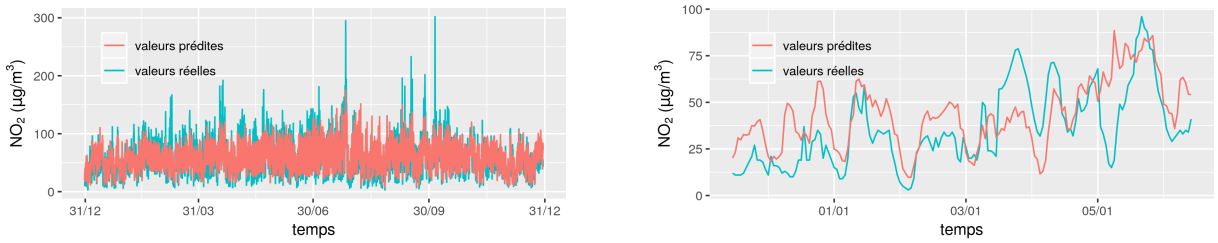


Figure 34: Mesures contre prévisions par RF la première semaine

L'allure globale sur l'année ressemble nettement plus aux mesures que pour les autres techniques. En revanche, la première semaine n'est pas bien prédite avec des jours sous-estimés (5 janvier) et des jours surestimés (8 janvier). La première semaine de janvier est en effet doublement difficile à prévoir car c'est une semaine de vacances qui débute par le 1er janvier, journée qui n'est pas des plus productives ! Heureusement si l'on choisit d'autres semaines, on vérifie que les résultats sont meilleurs.

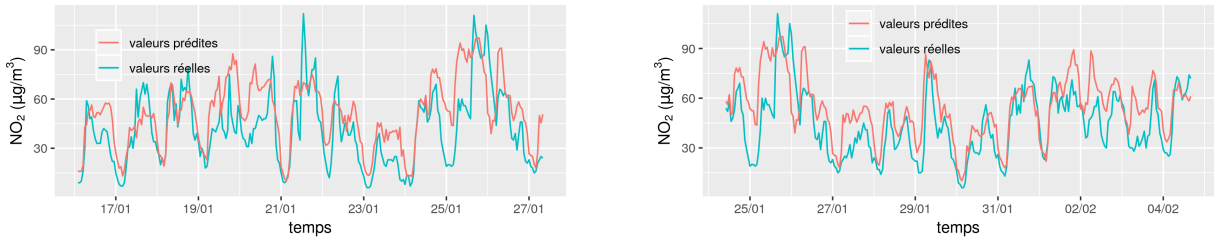


Figure 36: Mesures contre prévisions par RF

variable	%IncMSE	IncNodePurity
k	120.41	967501
q	150.5	1184458
temperature	134.5	1575773
wind direction	285.5	2938777
wind speed	299.8	3086032
shortwave radiation	108.1	657084
relitave humidity	119.5	1008877
hour	404.9	2979438

Table 4: Importance des variables

De manière générale, les modèles peuvent très bien prédire une semaine et être mauvais pour une autre. On réprésente le MAPE par semaine de l'année pour illustrer ce phénomène. On remarque que le modèle est très peu performant tout début janvier et au mois d'août qui sont des périodes de faible activité.

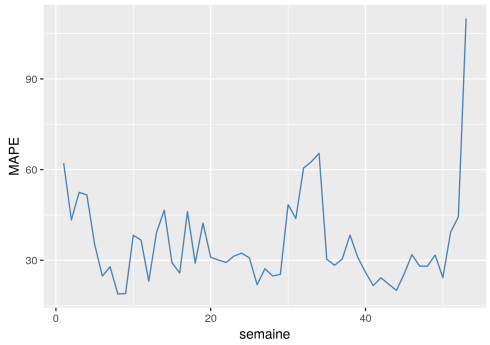


Figure 37: MAPE par semaine

Concernant l'importance des variables, l'heure de la journée semble la plus importante suivie du vent et du trafic routier.

Remarque : On peut retirer la saisonnalité (par exemple en effectuant un GAM à splines cycliques, $k = 24$, sur l'heure) puis modéliser la série désaisonnalisée avec une RF. L'erreur obtenue est sensiblement la même (avec ou sans la variable heure), ce qui montre que la RF a capté le cycle journalier. Par ailleurs, le GAM donne le profil journalier moyen et à titre de comparaison on peut calculer l'erreur commise en le prenant comme prédicteur (indépendant des variables explicatives hormis l'heure donc). On obtient un RMSE de 27 et un MAPE de 51.5 largement supérieurs aux autres méthodes et c'est assez rassurant.

4.3.4 Boosting

On explore les méthodes dites par boosting qui consistent à entraîner séquentiellement des prédicteurs faibles chacun corrigeant l'erreur du précédent.

On commence par un modèle de type "Generalized boosting regression" avec comme base learner un arbre de régression. On doit spécifier le nombre d'arbres **Ntree** utilisés. On retient **Ntree**=500 car il n'y a pas d'amélioration notable au delà. On obtient un RMSE de 21 et un MAPE de 36.9 qui est légèrement moins bien que les forêts aléatoires. On note la même difficulté à prédire janvier et les mois d'été.

On teste une autre méthode de boosting Xgboost qui est connue pour son efficacité mais qui nécessite de calibrer de nombreux hyer-paramètres.

On choisit les paramètres optimaux de Xgboost par gridsearch : le nombre d'itérations **nrounds**=300, la profondeur maximale **maxdepth**=10, le ratio du nombre de variables pour construire chaque arbre **colsamplebytree**=0.9 et on garde la valeur par défaut du taux d'apprentissage **eta**=0.1. On obtient un RMSE de 21 et un MAPE de 32.9 soit le meilleur jusqu'à présent.

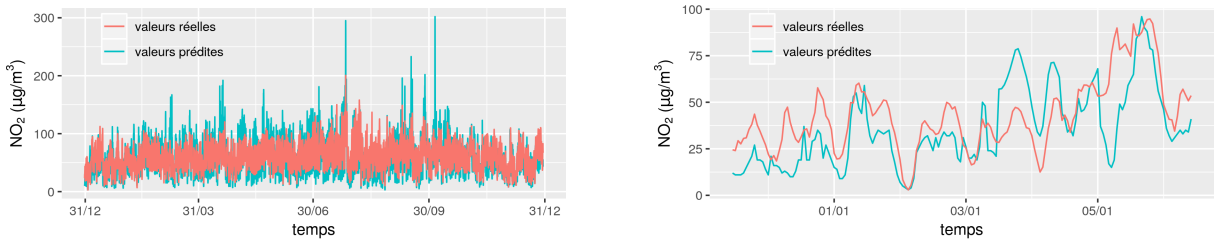


Figure 38: Mesures contre prévisions par GBM

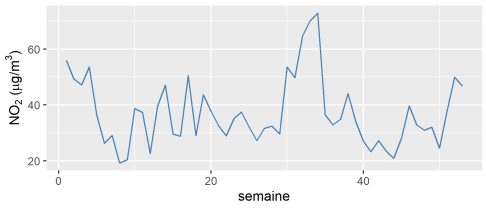


Figure 39: MAPE par semaine

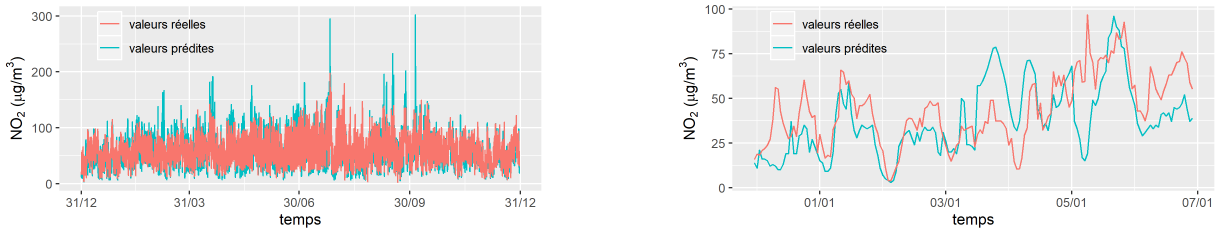


Figure 40: Mesures contre prévisions par Xgboost

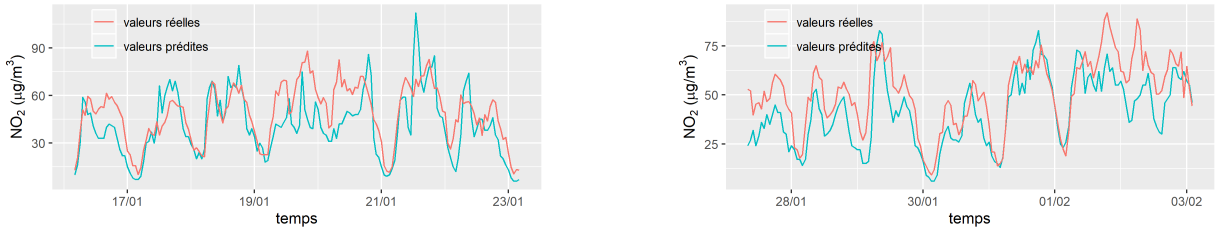


Figure 41: Mesures contre prévisions par Xgboost

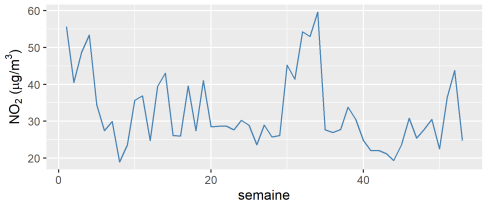


Figure 42: MAPE par semaine

4.4 Conclusion

On note tout d'abord que malgré le cadre plus simple que nous avons choisit dans cette partie (c'est à dire la prédiction, ou plutôt l'estimation du taux de NO_2 à l'instant t connaissant les données de trafic et météo de l'instant t) les modèles sont tous assez peu précis, avec une erreur MAPE de l'ordre de 35 à 50 %.

Plusieurs raisons peuvent expliquer cela. Premièrement, le problème étudié ici est peut-être intrinsèquement difficile, avec des mécanismes physico-chimiques complexes et des intéractions entre différentes activités humaines (trafic, chaufage etc.). Une deuxième raison, liée à la première, est que les variables explicatives utilisées n'étaient peut-être pas les bonnes. On peut imaginer que la prédiction aurait pu être améliorée en n'intégrant par exemple pas seulement les mesures brutes de mesure du trafic (débit q et taux d'occupation k) mais une fonction des 2 qui fonctionnerait comme un prémodèle de la pollution liées au trafic routier (voir par exemple [7] pour des exemples de modèlisation de l'impact du trafic routier sur la pollution de l'air). Nous avons pas non plus dans le cadre de cette section intégré à nos modèles des données temporelles (par exemple le taux de NO_2 de la veille).

On note également qu'aucun des modèles n'a réussi à prédire les pics de pollutions ce qui est cohérent avec les difficultés rencontrées dans la section 3. Ceci peut venir du fait que ces pics sont causés par évenèments qui ne sont pas codés dans nos données ou bien tout simplement du fait que les cas de pics sont trop rares pour avoir un impact sur la minimisation de la perte utilisée.

Pour ce qui est des comparaisons, le modèle qui s'en sort le mieux pour la de mesure d'erreur de prédiction est le réseau de neurone, suivit de la forêt aléatoire du boosting d'arbres et des GAM. Les modèles linéaires classique et LASSO sont ici assez mauvais. L'approche LASSO en particulier ne se justifie pas.

Modèle	MAPE	RMSE
modèle linéaire	40.75	27
LASSO	53.2	26
GAM	34.73	22
arbre de régression	44.7	25
bagging	40.9	22
forêt aléatoire	34.2	21
GBM	36.9	21
Xgboost	32.9	21
Réseau de neurones	34.95	19.26

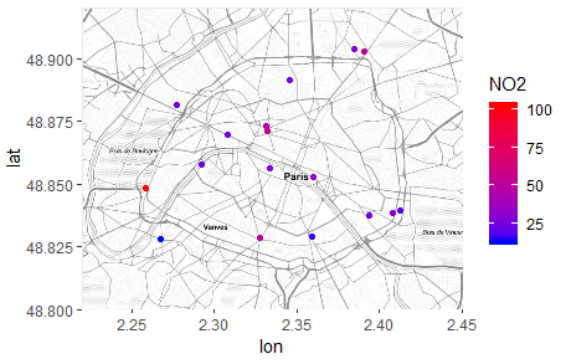
Table 5: Comparaisons des différentes approches

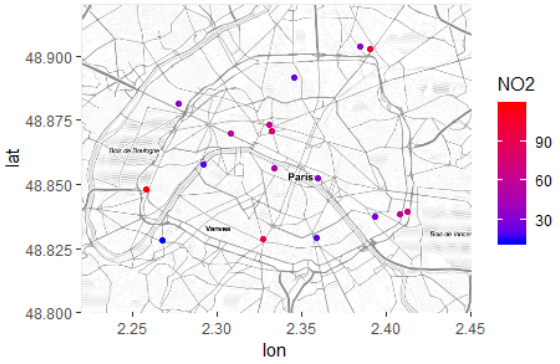
5 Prédictions spatiales

Nous disposons d'un ensemble de stations dans Paris et l'on souhaite obtenir une carte du niveau de pollution dans tout Paris. La contrentation en polluants dépend fortement des conditions météorologiques (dispersives ou non), de la topographie du lieu (exposition au vent, altitude) et des conditions de trafic, si bien que d'une rue à l'autre on peut observer des variations conséquentes. C'est assez flagrant sur les cartes ci-dessous où il est difficile de définir des zones de faible/forte concentration. Deux stations très proches peuvent mesurer des concentrations assez différentes (en par particulier Opéra/Hausman et Bd Soult/Périph Est) et entre deux instants les profils sont assez éloignés. Par ailleurs, on ne dispose de très peu de stations dans le Nord-Est parisien.

A l'exception de la première technique, les modèles d'interpolation seront basés sur des moyennes à poids des mesures. Nous quantitifierons la qualité de l'interpolation par cross-validation leave-one-out, c'est à dire que nous établirons une carte à partir des mesures de toutes les stations sauf une et nous calculerons l'erreur entre la vraie valeur et celle prédite par la carte en cette station.

Nous utiliserons le package R **gstat** qui implémente de nombreuses méthodes d'interpolation spatiales et comporte une fonction pour estimer l'erreur par cross-validation.





5.1 Interpolation cubique

Avant d'introduire les méthodes à moyennes, présentons d'abord une approche par interplation par splines bicubiques (l'équivalent d'un spline cubique en 1D). On utilise la fonction interp du package **akima** qui réalise une interpolation à partir de données qui ne sont pas nécessairement sur une grille régulière.

La carte représente la concentration en NO2 le 01/11/2017 à 9h obtenue par ce procédé. On observe une pollution importante près des stations mesurant la pollution sur le périphérique ainsi que dans l'hypercentre de Paris (en heure de pointe). On remarque aussi que l'Est Parisien est très uniforme avec une forte concentration car on interpole avec très peu de stations dont deux prrès du boulevard périphirique ou d'une route nationale (RN2) .

Dans le package Akima, il n'y a pas de fonction pré-programmée pour calculer l'erreur par validation croisée et pas de méthode simple pour superposer la carte de Paris. On présente donc juste cette méthode à titre indicatif et pour comparer visuellement aux autres méthodes du package **gstat**.

5.2 Interpolation par distance inverse

On utilise ici une méthode d'interpolation à poids IDW (inverse distance weighting). La valeur en un point est la moyenne des mesures aux **nmax** stations les plus proches pondérées par l'inverse de la distance au carré (**nmax**=∞ par défaut). Le choix du carré est classique mais on peut choisir une autre puissance via **idp**.

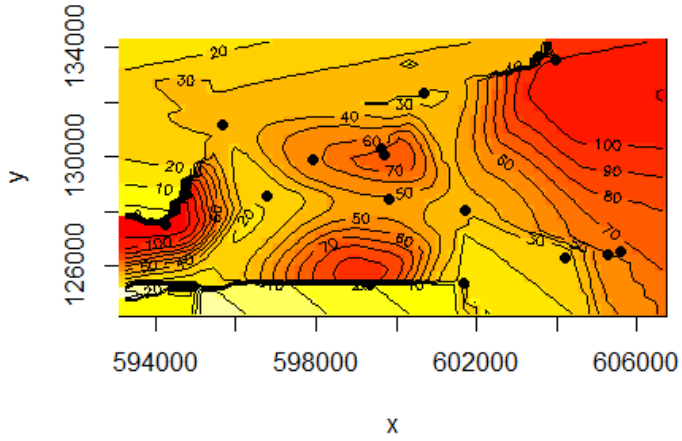


Figure 43: splines

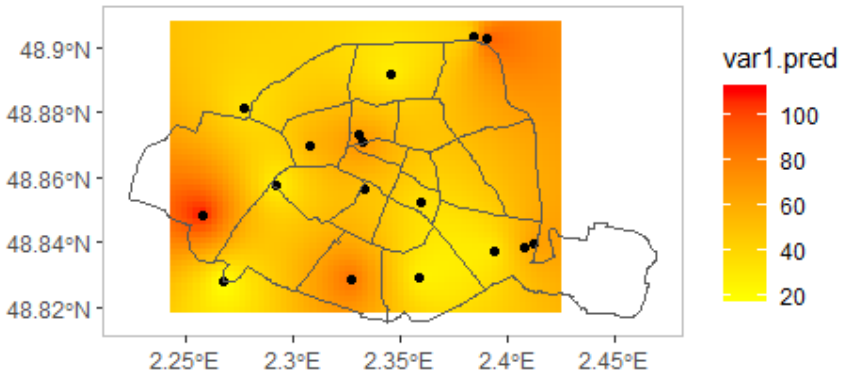


Figure 44: IDW

On observe que contrairemet à l'interpolation bicubique qui prédit une forte concentration dans l'est Parisien, le modèle IDW est plus conservateur et lisse plus.

MAE	RMSE	RMSE sd
31.24	38.68	1.289

Table 6: Erreur IDW

5.3 Interpolation par plus proches voisins

On utilise ici l'interpolation par plus proches voisins avec **nmax=5**. Elle correspond en fait à IDW avec **idp** $= 0$.

La carte est bien sûre très discontinue : il faudrait beaucoup plus de capteurs pour que cette méthode marche en pratique.

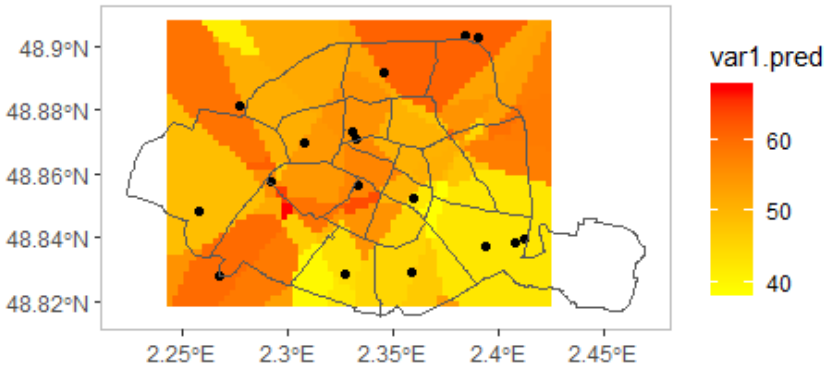


Figure 45: 5 plus proches voisins

MAE	RMSE	RMSE sd
26.19	30.89	1.031

Table 7: Erreur IDW

5.4 Krigeage direct

Les techniques précédentes n'exploitent que très peu la corrélation entre les stations - ou elle ne l'exploite que via la distance. On se propose donc de procéder par krigeage qui est une approche classique en statistiques spatiales et qui a été développée par des ingénieurs miniers pour étudier l'abondance de certaines minéraux. Cette méthode repose sur l'estimation du semi-variogramme :

$$\gamma(h) = \frac{1}{2}(Var(Z(s+h) - Z(s)) \tag{1}$$

qui permet ensuite de calculer les poids du "meilleur" estimateur affine (au sens de sans biais et de variance minimale) :

$$\hat{Z}(s_0) = \alpha + \sum_{i=1}^{n} \lambda_i Z(s_i) \tag{2}$$

Le variogramme est très bruité et il est difficile de fitter un modèle (en testant des des variogrammes par direction ce n'est pas beaucoup mieux). Le problème vient probablement du faible nombre de stations et du fait que certaines paires de stations très proches peuvent mesurer des concentrations assez différentes suivant leur positionnement périphérique/trafic/urbain (Pantin/Aubervilliers, Bd Soult/Périph Est, Opéra/Hausman).

On calcule malgré tout une carte par krigeage à partir de ce variogramme fitté et on obtient un résultat assez similaire à IDW.

Dès que l'on s'éloigne un peu des stations la variance de krigeage devient vite très grande.

MAE	RMSE	RMSE sd
25.6	31.61	1.055

Table 8: Erreur krigeage

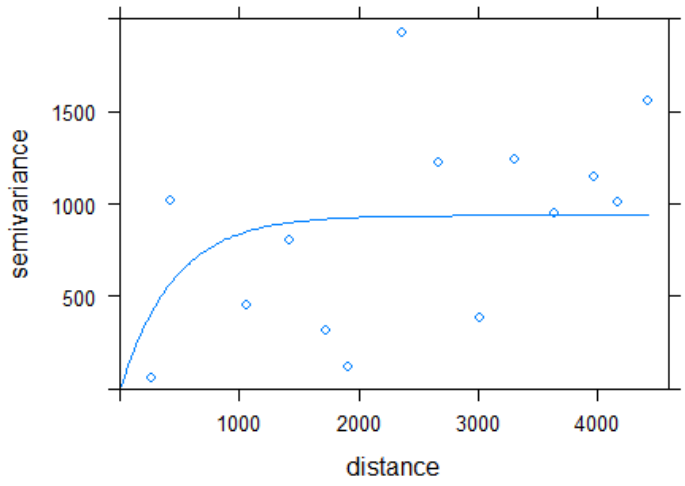


Figure 46: semivariogramme

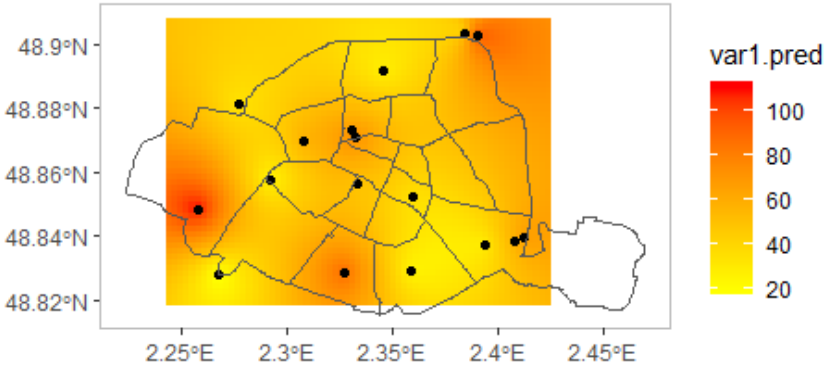


Figure 47: Carte par krigeage

5.5 Conclusion

Les résultats présentés ici ne sont que moyennement convaincants principalement car le nombre de stations est trop faible mais il existe aussi deux autres raisons.

Tout d'abord, la prévision en pratique ne se fait pas directement sur les mesures mais sur la différence entre les mesures et un modèle physique qui prédit à plus haute résolution spatiale.

Deuxièment, on interpole des stations de natures différents : trafic/urbaine/périurbaine. Une méthode pour contourner ce problème est de distinguer pollution de fond et pollution routière. La carte de pollution de fond n'utilise pas les données de stations de type trafic et produit une carte gobale. La carte de pollution routière prédit la pollution uniquement sur les axes principaux en utilisant les stations de trafic et en supposant que la pollution tend à se conserver le long des axes. La carte finale est obtenue par superposition des deux.

A notre échelle de temps et de compétence, ces deux approches sont difficiles à mettre en oeuvre mais nous avons cependant envisagé un compromis entre les deux. On pourrait commencer par réaliser une régression de la concentration de NO2 sur le trafic puis kriger l'erreur. On superposerait alors la carte de NO2 obtenue par régression sur les capteurs routier à celle obtenue par krigeage de l'erreur. Nous n'avons malheureusement pas eu le temps de mettre en place cette méthode.

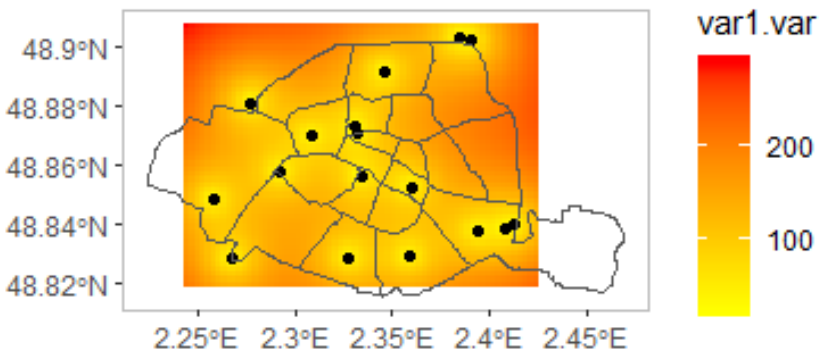


Figure 48: Variance de krigeage

6 Conclusion générale

Nous sommes rendus compte que la pollution de l'air est un problème nettement plus complexe qu'il n'y parait. Nous pensions au début du projet qu'avec les variables explicatives adéquates, à savoir la météo pour la dispersion des polluants et le trafic routier pour son émission, nous serions en mesure de prédire la concentration avec une bonne précision. Cependant, le lien entre ces variables explicatives et la concentration en polluant est hautement non-linéaire si bien que dans la littérature il n'existe pas de méthode canonique. Une méthode peut très bien marcher pour une station et moyennement pour une autre. Des variables peuvent être importantes pour une station et moins pour une autre. Au quai des Célestins l'heure de la journée est essentielle, le vent semble jouer un rôle majeur (proximité de la Seine et forte exposition) et le trafic routier joue également dans une moindre mesure.

Nous avons tout d'abord essayé de prédire des dépassement de seuil à partir des données à j-1,d'abord en calculant le taux maximum d'une journée via une régression puis en voyant la tâche comme un problème de classification. Puis nous avons traité le problème plus simple de "prédire" une année en supposant connues les variables explicatives. Nous avons utilisé des modèles de régressions linéaire, LASSO et GAM; des modèles d'ensemble Random Forest et de Boosting; un modèle par réseau de neurones perceptron multi-couches. Ces modèles ont tous les mêmes difficulté à prédire les pics (ce qui est cohérent avec les difficultées rencontrées avec les dépassements de seuil). Enfin, nous avons exploité l'aspect spatial de la base de données en produisant des cartes de pollution par interpolation des mesures aux station.

References

[1] Comptages routiers permanents. Ville de Paris. https://opendata.paris.fr/explore/dataset/comptages-routiers-permanents/information/.

[2] Meteo montsouris. MeteoBlue. https://www.meteoblue.com/fr/meteo/semaine/parc-montsouris_france_11789323.

[3] Paris sans voitures. https://www.airparif.asso.fr/_pdf/publications/communique_presse_journee_sans_voiture_150927.pdf.

[4] Pollution. Airparif. https://www.airparif.asso.fr/telechargement/telechargement-station.

[5] Referentiel geographique. Ville de Paris. https://opendata.paris.fr/explore/dataset/referentiel-comptages-routiers/information/.

[6] Aurélien Duret, Stéphane Chanut, Frédéric Murard, Sylvain Belloche, Sébastien Plantier, and Fabrice Reclus. *Théorie du trafic et régulation dynamique.*

[7] Masoud Fallah Shorshani. *Modélisation de l'impact du trafic routier sur la pollution de l'air et des eaux de ruissellement.* PhD thesis, Paris Est, 2014.