Rapport de modélisation prédictive

Alexandre Loret , Lorenzo Brucato

2024-03-01

Rapport Modélisation prédictive

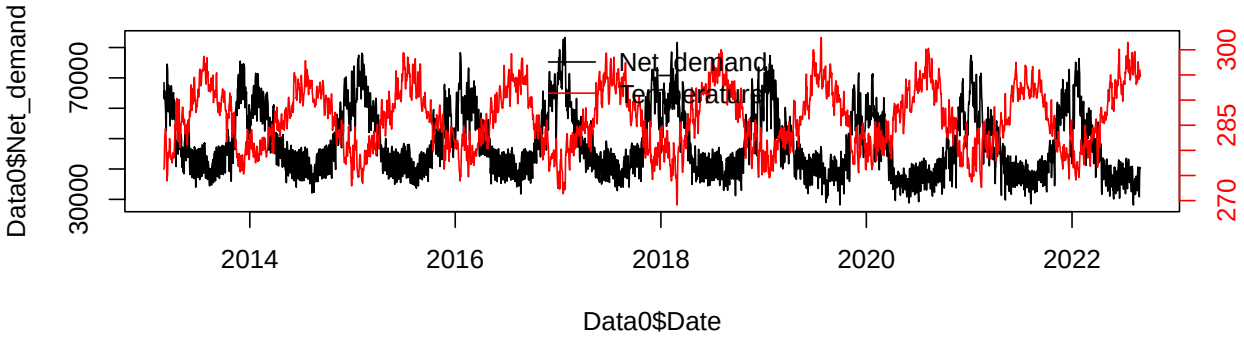
Introduction

L'objectif de ce projet de modélisation est d'expliquer au mieux la consommation énergétique en France durant la période de sobriété énergétique (octobre 2022 - octobre 2023). L'enjeu est de limiter le surplus de production dans un contexte économique et énergétique en tension où l'on a incité à baisser drastiquement notre consommation. Dans ce contexte, la variable cible de notre problème de modélisation est le quantile 95% de la demande nette d'énergie qui retire à la consommation brute les productions renouvelables estimées (solaire et éolienne).

Prise en main des données et premières visualisations

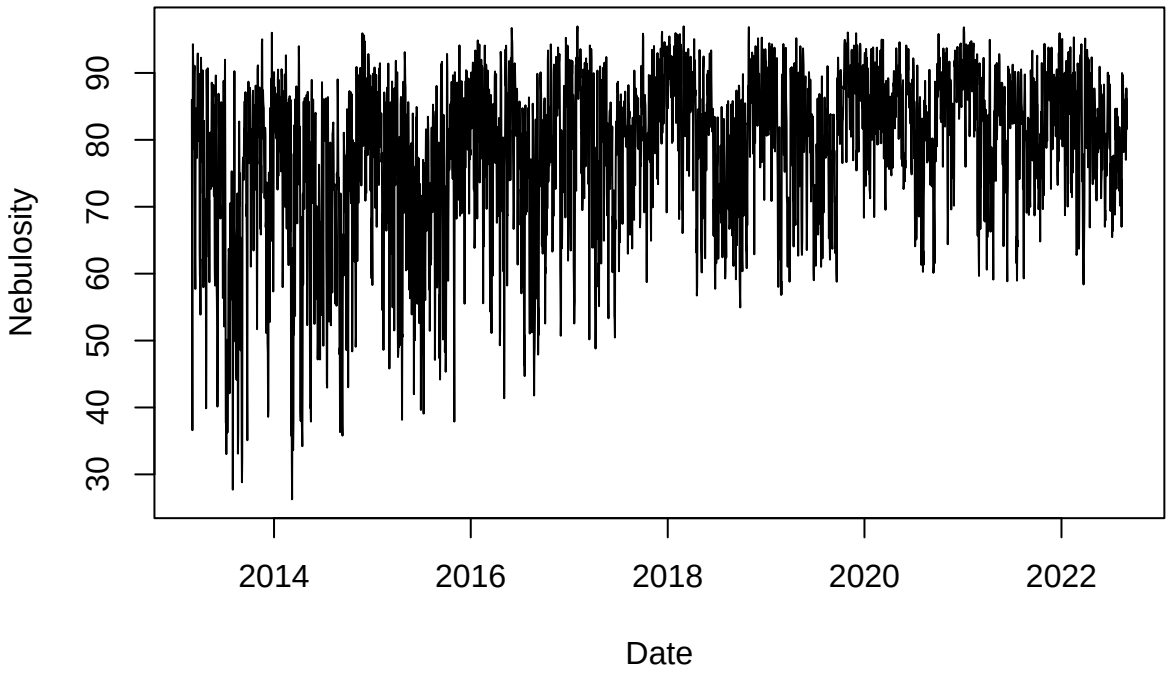
Pour la première semaine, nous nous sommes concentrés sur la visualisation et la compréhension des données. Nous avons effectué une première soumission de forêt aléatoire simple sur quelques variables explicatives pour se donner un premier score et résultats de référence, en prenant des variables de températures dont on peut visualiser la forte anti-corrélation avec la demande nette :

```
eq0 = Net_demand ~  Time + toy + Temp + Net_demand.1 + Net_demand.7 +Temp_s99 +
WeekDays + BH + Temp_s95_max + Temp_s99_max + Summer_break  + Christmas_break +
  Temp_s95_min +Temp_s99_min + DLS
```



Nous avons également restreint notre jeu d'entrainement aux données prises depuis 2018 pour deux raisons particulières : D'une part, pour que nos modèles tiennent plus en compte la période récente et ne se surajustent pas à des phénomènes passés dont on suppose qu'ils ont moins d'influence sur les comportement de la

consommation actuelle. D'autre part, on visualise aussi que certaines variables (nebulosité) ont été mesurées différement avant 2018 et risquent ainsi d'apporter du biais sur nos modèles



Par ailleurs, on a aussi ajouté de nouvelles variables dont on a pu supposer et vérifier l'impact significatif sur nos modèles (que ce soit pour l'explication de la variance ou pour la hausse de la précsion en validation) :

- La variable factorielle Covid, indiquant les périodes de confinement et dont on a pu constater l'effet au vu des variations de la consommation en 2020 par rapport aux autres années
- Une variable factorielle WeekDays3, qui permet de mettre en avant les jours de travail en semaine, qui permettent notamment d'expliquer la différence de consommation par rapport aux week-ends.
- La variable Temp_trunc, troncature sur les températures pour capturer les phenomènes extremes
- Price et Inflation (INSEE) : données malheuresement peu exploitables car non quotidiennes et prises tous les six mois. L'idée de départ était de supposer que la consommation aurait pu être impactée par l'inflation et une hausse du prix de l'éléctricité

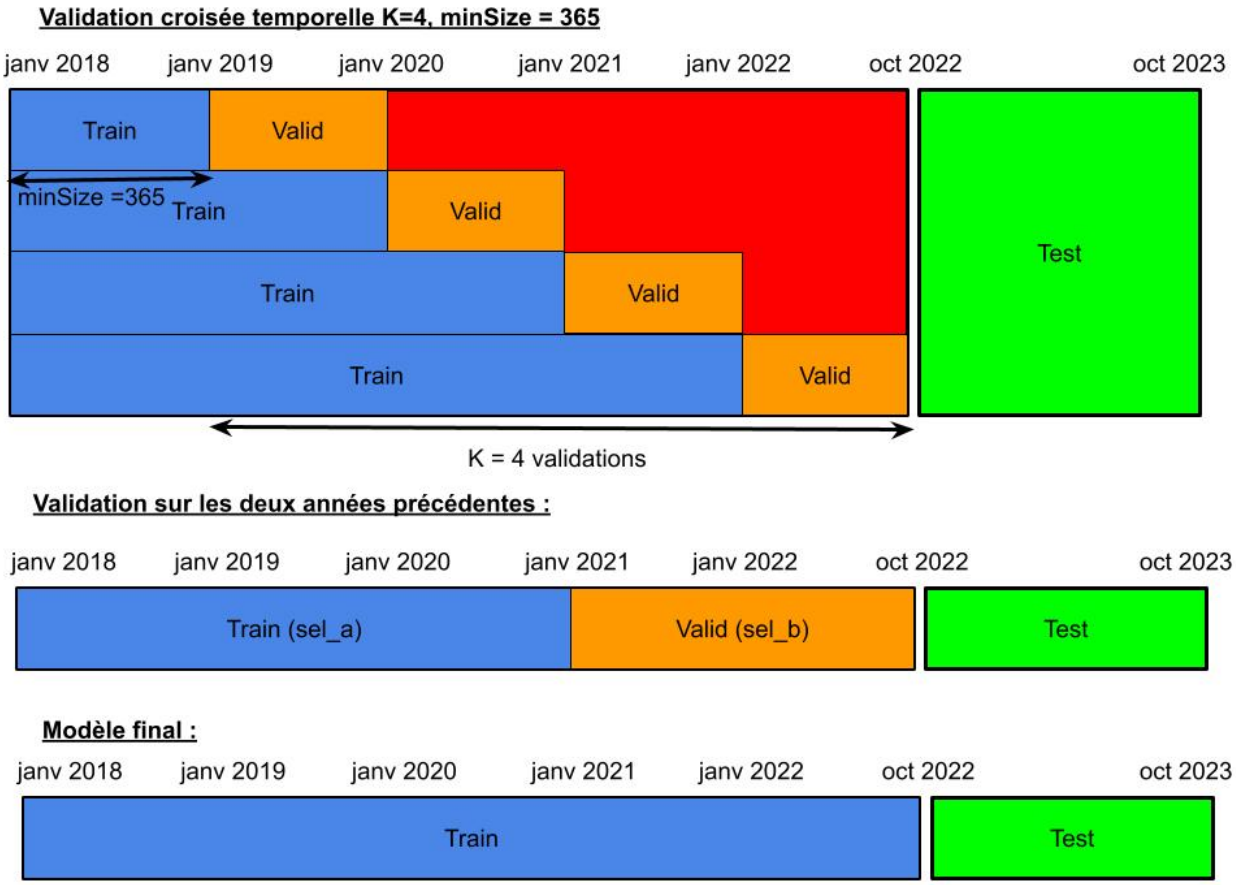
Pour valider nos modèles nous nous sommes basés conjointement sur deux métriques : la RMSE et la pinball Loss.

La RMSE nous a permis d'évaluer la performance globale de nos modèles et de vérifier que l'on ne se trouvait pas en situation de sous-apprentissage ou sur-apprentissage.

La pinballLoss était étudiée en parallèle pour mieux percevoir en particulier la capacité de notre modèle à prédire les quantiles de manière précise et robuste (car il s'agit de notre objectif final)

Pour avoir une idée plus précise de la robustesse de nos modèles et de leur capacité à généraliser, nous avons effectué pour chaque modèle :

- Une validation croisée temporelle avec un nombre de validations généralement de 4 (pour séparer selon chaque année)
- Une validation sur une période de 18 mois avant la période de test



Remarque : on constatera souvent de moins bonnes performances pour le deuxième bloc de validation (si l'on prend K=4). Cela s'explique par le fait que ce bloc de validation tombe en période de Covid dont l'effet n'est pas encore visible sur les périodes précédentes.

Modèles linéaires

Nous nous sommes d'abord intéressés au modèles de régression linéaire, en effectuant une analyse ascendante : on a ajouté au fur et à mesure les variables pour lesquelles on a pu observer à la fois une hausse de la variance expliquée / significativité globale du modèle (critère du R2, test de significativité de Student pour chacune des variables), ainsi qu'une hausse de la qualité de prédiction en validation croisée (à la fois sur le jeu d'entraînement et de validation).

Une première équation de référence prend en compte le temps et les variables de décalage sur la demande nette (Net_demand.1, Net_demand.7), la consommation globale (Load.1, Load.7), et la production solaire et éolienne (Solar_power.1, Solar_power.7, Wind_power.1, Wind_power.7). Elles permettent de mettre en avant la relation temporelle de nos données :

```
# Equation sur les variables de décalage temporel
eq1 = Net_demand ~ as.numeric(Date) + Load.1 + Load.7 + Wind_power.1 + Wind_power.7 +
  Net_demand.1 + Net_demand.7 + Solar_power.1 + Solar_power.7
```

```
##   tsize       R_2 RMSE_train RMSE_test MAPE_train MAPE_test
## 2   633 0.8773222       3804      4089       6.08      6.88
## 3   901 0.8711550       3869      4138       6.28      6.60
## 4  1169 0.8692518       3919      3521       6.36      6.34
## 5  1437 0.8679317       3840      3793       6.34      6.34
```

On a avec ce modèle de base une variance expliquée à 87% en moyenne et une RMSE élevée (supérieure à 3000) que ce soit pour le jeu d'entraînement ou de validation, indiquant que l'on reste en sous-apprentissage. De plus, la significativité est relativement faible pour les variables décalées de sept jours, notamment pour la production solaire et éolienne dont les phénomènes naturels n'ont pas de raison d'être corrélés aux effets des semaines (contrairement à la Load et la Net_demand qui sont liés aux activités humaines). On décide ainsi de conserver les variables significatives et d'y ajouter les effets des journées de travail, du covid ainsi que les effets météorologiques (température, nébulosité, vent) pouvant expliquer la production solaire et éolienne :

```
# Effets des conditions météorologiques
eq2 = Net_demand ~ as.numeric(Date) + Net_demand.1 + WeekDays3 + Covid + WeekDays3 +
  Nebulosity_weighted +  Wind_weighted + Temp
```

```
##   tsize       R_2 RMSE_train RMSE_test MAPE_train MAPE_test
## 2   633 0.9530019       2354      2799       3.71      4.87
## 3   901 0.9499387       2412      2571       3.99      4.18
## 4  1169 0.9507224       2406      2388       3.96      4.20
## 5  1437 0.9487450       2392      2480       3.98      4.47
```

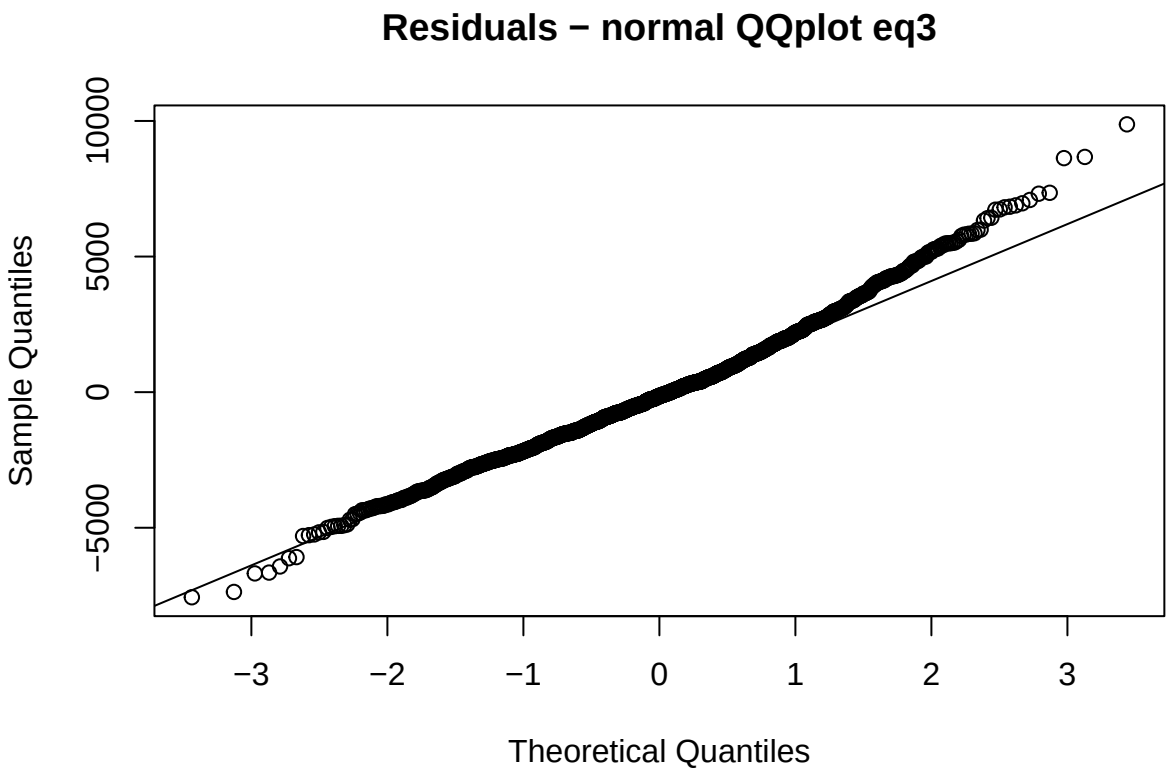
On observe sur ce modèle de régression linéaire une forte hausse de la significativité du modèle avec près de 95% de variance expliquée pour le R2, et une diminution des erreurs de prévision (RMSE, MAPE) sur le jeu d'entraînement et test de près de 30% du modèle précédent. La significativité de ce modèle par rapport au précédent est également justifiée par un test de rapport de vraisemblance (anova). Cependant, la différence entre erreur d'entraînement et test est encore faible et la RMSE élevée, ce qui laisse entendre que le modèle est encore en sous-apprentissage.

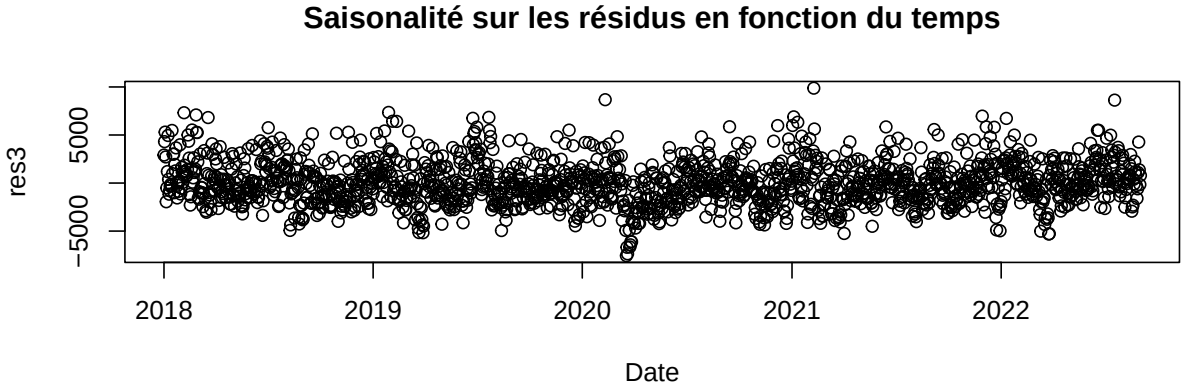
En ajoutant et tesant plusieurs variables, nous avons finalement convergé vers un premier modèle avec une significativité non négligeable pour chacune des variables :

```
# + effets des jours feriés
eq3 = Net_demand ~ as.numeric(Date) + Load.1 + Wind_power.1 +
  Nebulosity_weighted +  Wind_weighted + Temp + Temp_s95 +
  Summer_break + Christmas_break + DLS + BH_Holiday + Holiday +
  BH_before + Wind
```

```
##   tsize       R_2 RMSE_train RMSE_test MAPE_train MAPE_test
## 2   633 0.9579000       2228      2405       3.73      4.17
## 3   901 0.9566399       2245      2793       3.75      4.39
## 4  1169 0.9550276       2299      2143       3.80      3.88
## 5  1437 0.9543419       2258      2383       3.78      4.13
```

Ces modèles de regression font en particulier une hypothhèse gaussienne sur la distribution des résidus, centrés en 0 et de variance constante. En visualisant la distribution sur les résidus de nos modèles et en tracant le QQplot des résidus, on se permet ici de valider cette hypothèse :





On constate cependant encore un comportement saisonnier sur les résidus en fonction du temps. De plus l'écart entre erreur d'entrainement et validation est faible et l'erreur globale élevée. Cela laisse entendre que l'on reste en sous apprentissage et que l'on peut encore tenter d'expliquer la variabilité des résidus.

Nous avons également tenté l'ajout de variables par transformée de fourier : le score pinball du modèle a également été amélioré par l'ajout de 20 nouvelles variables (cos et sin) produites par une série de Fourier. Nous avons également essayé la regression quantile mais qui n'a pas apporté de résultat plus significatif.

Equation par série de fourier :

```
# Equation avec série de fourier
eqFourier = Net_demand ~ WeekDays3 + Temp + Temp_trunc1 + Temp_trunc2 +
```

```
  Nebulosity_weighted + Wind_weighted + BH + Cos_i + Sin_i + Net_demand.1 + Net_demand.7
```

Un dernier essai a été de décomposer la prédiction de la demande nette par l'estimation de la consommation brute et la production renouvelable séparemment.

Cependant nous ne sommes pas parvenu à trouver des modèles permettant de prédire très efficacement les productions solaires et éoliennes (moins de variables explicatives pour ces phenomènes et R_2 ne dépassant pas les 90% de variance expliquée).

En conséquence, la RMSE accumulée de ces modèles séparés donnait des résultats moins bons que certains des premiers modèles réalisés sur la demande nette. Nous avons donc abandonné cette idée.

Modèles additifs

Afin d'améliorer encore la qualité de nos prévisions, nous nous sommes ensuite intéréssés aux modèles additifs, qui nous ont permis de lisser certaines relations non linéaires de nos covariables et de gagner davantage en précision.

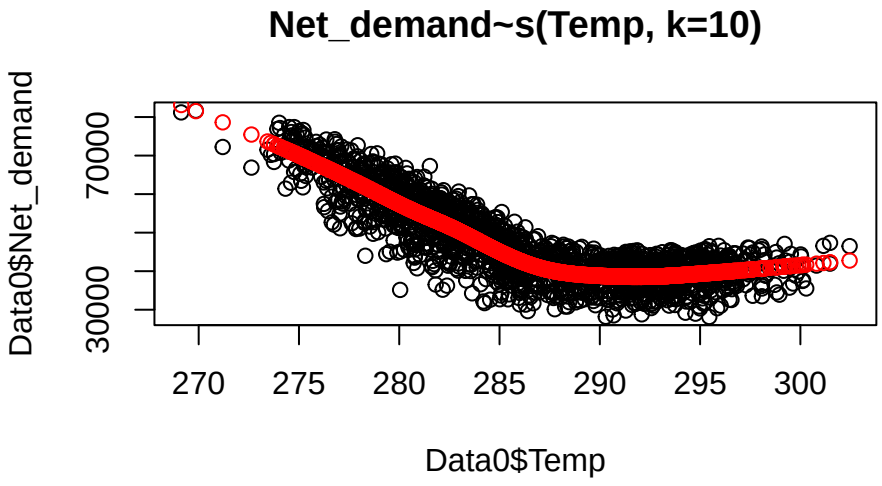
Avec le nombre important de combinaison et d'interactions de variables possibles, nous avons convergé vers plusieurs modèles gam qui ont apporté de meilleurs résultats en validation croisée.

Nous avons porposé un premier modèle gam par analyse ascendante, en ajoutant toujours progressivement des variables significatives :

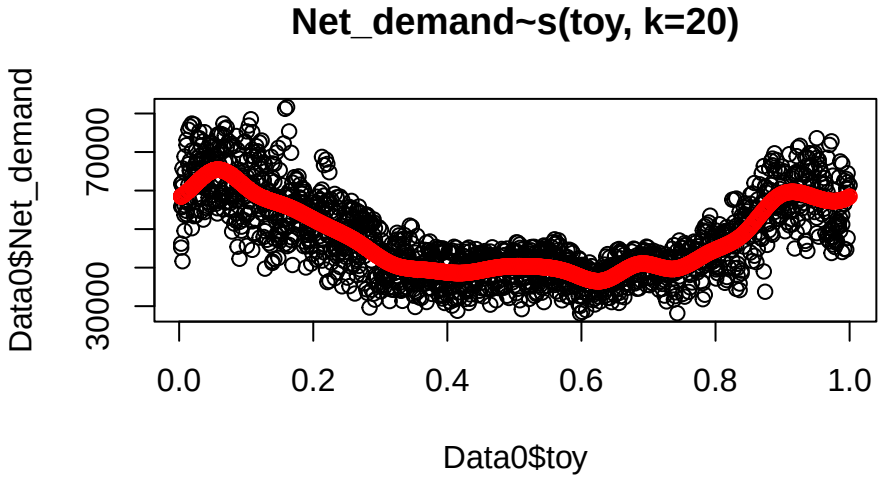
```
# Première equation gam
eqGam1 = Net_demand~s(as.numeric(Date),k=3, bs='cr') + s(toy,k=20, bs='cc') +
  s(Temp,k=10, bs='cr') + te(Net_demand.1,Net_demand.7, k=10, bs='cr') +
  s(Temp_s99,k=10, bs='cr') + WeekDays2 + BH +
  te(Nebulosity_weighted,Wind_weighted,k=10, bs='cr') + Covid
```

```
##   tsize       R_2 RMSE_train RMSE_test MAPE_train MAPE_test
## 2   633 0.9907901        946      2921       1.46      4.91
## 3   901 0.9864515       1203      2318       1.95      3.66
## 4  1169 0.9845531       1298      1516       2.06      2.78
## 5  1437 0.9847037       1254      1519       2.04      2.43
```

Dans ces modèles, nous avons un varaible cyclique **toy**. Pour les autres variables, nous avons ajusté nos découpages (paramètre k) selon les relations que nous avons visualisé. Voici quelques exemples d'ajustement visualisés sur la variable de température et toy :



La variable cyclique toy en particulier nous permet de capturer des tendances de variabilité différentes sur une année complète. En augmentant le paramètre k à 20, on peut ainsi capturer plusieurs comportements différents sur l'année. Nous avons vérifié en parallèle que ce choix de paramètre n'amenait pas de surapprentissage en validation croisée :



Une deuxième version du gam avec BH_before en plus :

```
# eqGam2
eqGam2 = Net_demand~s(as.numeric(Date),k=3, bs='cr') + s(toy, k=20, bs='cc') +
  s(Temp,k=10, bs='cr') + te(Net_demand.1,Net_demand.7, k=10, bs='cr') +
  s(Temp_s99,k=10, bs='cr') + WeekDays2 + BH +
  te(Nebulosity_weighted,Wind_weighted,k=10, bs='cr') + Covid + BH_before
```

```
##   tsize       R_2 RMSE_train RMSE_test MAPE_train MAPE_test
## 2   633 0.9909780        935      2893       1.46      4.88
## 3   901 0.9866724       1191      2295       1.94      3.61
## 4  1169 0.9848736       1285      1532       2.05      2.79
```

```
## 5  1437 0.9849319       1244      1498       2.03      2.40
```

Cependant, la significativité par rapport au modèle précédent reste faible ici.

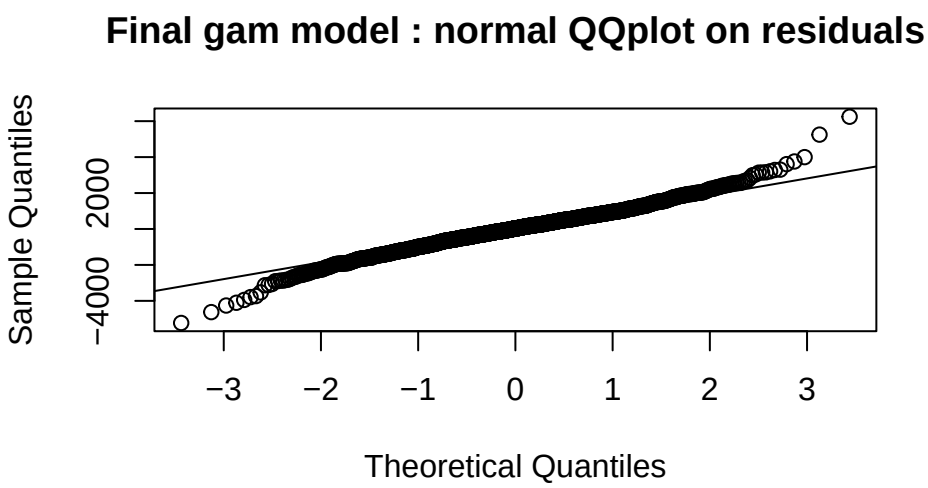
Nous avons aussi cherché des modèles gam en repartant de l'équation eq3 de nos modèles linéaires, en lissant les covariables et en supposant que le meilleur modèle pour la regression linéaire serait une meilleure référence pour nos modèles additifs. Sur ce modèle en particulier, on a jouté l'effet des vacances. En particulier **Holiday__zone__b** est un peu plus significative ici, suggérant que la zone centrale des périodes de vacances est plus pertinent (possiblement car elle est la période où le plus de personnes sont en vacances)

```
# Analyse ascendante en partant de eq3
eqGam3 = Net_demand ~ s(as.numeric(Date), k=3, bs='cr') + te(Load.1, Load.7, bs='cr') +
  Solar_power.1 + Wind_power.1 + Net_demand.1 + Net_demand.7 + WeekDays3 + Covid +
  Nebulosity_weighted + Wind_weighted + te(Temp, Temp_s95, k=3, bs='cr') +
  Summer_break + Christmas_break + DLS + BH_Holiday + s(toy, k=20, bs='cc') +
  + Holiday + Holiday_zone_b +  BH + BH_before + BH_after
```

```
##   tsize       R_2 RMSE_train RMSE_test MAPE_train MAPE_test
## 2   633 0.9932118        873      2246       1.43      3.67
## 3   901 0.9899946       1054      1915       1.73      3.19
## 4  1169 0.9892520       1104      1234       1.80      2.36
## 5  1437 0.9890168       1090      1089       1.79      1.87
```

Sur ce modèle encore, on est parvenu à baisser de manière bien plus significative nos scores de train et tests. En validation croisée, la RMSE test et train est ici entre 1000 et 1200 environ sur les dernières périodes. Il s'agit du modèle gam simple qui a donné la plus grande performance sur kaggle sur le private set (129 sur le private set, visible après coup seulement).

De la même manière que pour la régression linéaire, on a également supposé une distribution gaussienne de nos résidus au vu du QQplot ci-dessous :



A ce stade nous ne sommes pas vraiment parvenu à trouver

de modèles significativement meilleurs. Nous nous sommes donc ensuite intéréssés aux résidus des modèles gam pour voir si l'on pouvait encore tenter d'expliquer une part de variabilité.

Random forest sur les résidus

Ce dernier modèle gam a été retenu comme référence à nouveau pour effectuer des random forest gaussiennes sur les résidus. On a au départ proposé deux nouveaux modèles :

En réappliquant ce qui fonctionnait avec le modèle précédent (notamment des regroupements de variables sous un te()), Deux nouveaux modèles ont été proposés avec une forêt aléatoire sur les résidus :

```
# Modele A :
eqGamRf1 = Net_demand~s(as.numeric(Date),k=3, bs='cr') +
  te(Load.1, Load.7, bs='cr') + te(Net_demand.1, Net_demand.7, bs='cr') +
  Solar_power.1 + Wind_power.1 + WeekDays3 + Covid+ Nebulosity_weighted +
  Wind_weighted + te(Temp, Temp_s95, k=3, bs='cr') + Summer_break +
  DLS + BH_Holiday + BH_before + s(toy,k=30, bs='cc')

# + rf sur les résidus (complet)
```

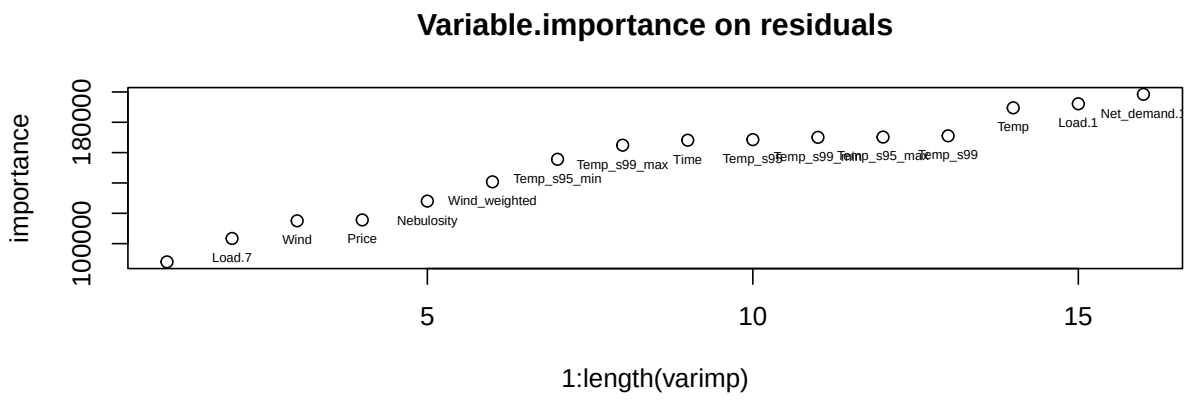
```
# Modele B :
eqGamRf2 = Net_demand~s(as.numeric(Date),k=3, bs='cr') + te(Load.1, Load.7, bs='cr') +
  te(Net_demand.1, Net_demand.7, bs='cr') + te(Solar_power.1, Wind_power.1, bs='cr') +
  WeekDays3 + Covid + Nebulosity_weighted + Wind_weighted +
  te(Temp, Temp_s95, k=3, bs='cr') + BH_Holiday + BH_before + s(toy,k=30, bs='cc') +
  Temp_trunc2

# + rf sur les résidus (complet)
```

Même si les résultats en test ont été légèrement améliorés avec ces modèles, on a constaté que l'écart en RMSE et MAPE se creuse davantage entre la performance en entrainement et test sur la validation (~ 500 en train, 1000 en test), ce qui laisse entendre que les méthodes de random forest sur les résidus commencent à faire du surapprentissage. Cependant, elles ont permis de baisser les scores de pinball loss.

Recherche sur l'importance des variables pour les résidus/integration sur les modèles gam

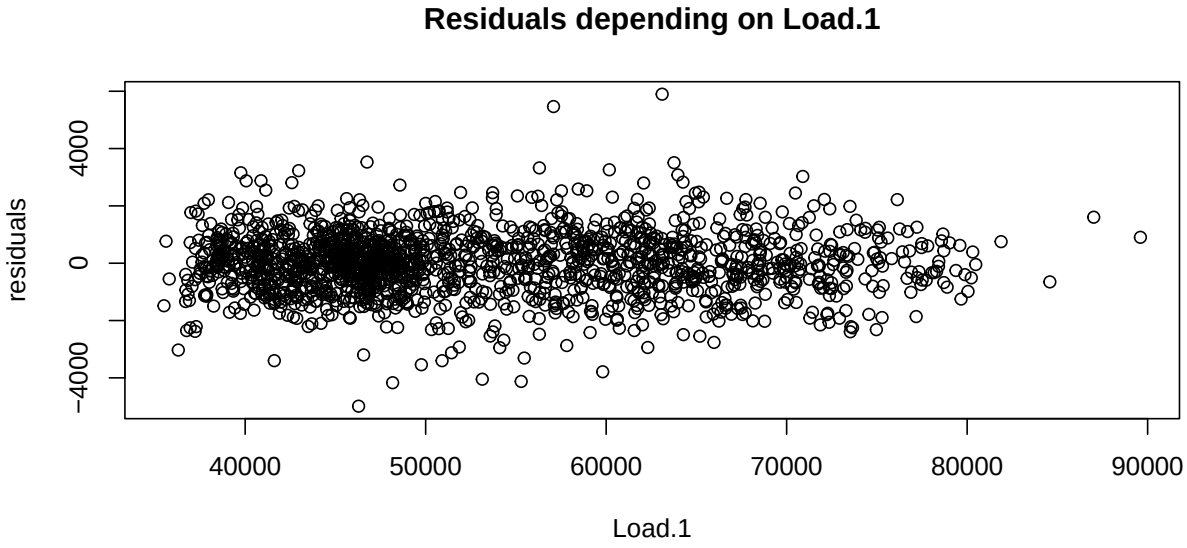
Nous avons ensuite cherché à limiter le forêt aléatoire aux variables les plus importantes : sur la figure ci-dessous, nous avons affiché l'importance des variables sur les résidus provenant du gam **eqGamRf1** :



On voit ici que les valeurs expliquant au mieux les résidus sont les variables Net_demand.1, Load.1 et les variables de températures. En se restreignant aux variables les plus importantes, nous avons pu obtenir un

modèle gam avec random forest sur l'equation **eqGamRf1** nous donnant finalement le modèle retenu avec un score public de 122 en pinball loss et de 125 en score privé.

En se basant sur ces résultats nous avons ensuite cherché à trouver de nouvelles relations entre les résidus restants et les variables les plus significatives. On a visualisé ici la dépendance entre les résidus et les variables les plus significatives pour la forêt aléatoire :



En rajoutant des interactions que l'on a trouvé significatives entre ces variables (tests de significativité du summary), nous avons tenté d'améliorer encore notre modèle gam (**eqGamRf1**). Nous avons en plus essayé de raffiner les interactions entre variables en utilisant des ti() (pour mieux distinguer les dépendances réellement significatives).

Mais les différences en résultats n'ont pas été significatifs en validation et sur le score publique (que ce soit en RMSE ~1000 ou en pinball loss ~130). Quelques uns des modèles essayés avaient été en réalité plus performants sur le private set (avec le meilleur score privé de 119 en pinball loss). Mais la différence n'est de quelques points seulement, ce qui ne permet pas réellement de les considérer comme meilleurs.

Une dernière idée aurait pu être de réaliser une aggrégation d'expert pour pouvoir distinguer les meilleurs modèles et avoir une prévision finale plus robuste.

Boosting, dernières tentatives et conclusion

La dernière semaine a permis d'expérimenter le boosting. Nous avons repris les équations **eqGamRf1** et **eqGamRf2**, mais en remplacant la prédiction des résidus par des méthodes de Boosting. Nous avons également essayé d'améliorer la pinball loss en passant sur des modèles quantiles (qgam).

Aucune de ces méthodes n'a vraiment convaincu ou apporté de meilleurs résultats par rapport à ce qui a été réalisé précédemment (scores similaires ou même bien moins bons en validation croisée et sur kaggle). Nous sommes donc resté sur les modèles que nous avions réalisés avec des gam et random forest. Le modèle final retenu est **eqGamRf1.**