Projet R : Prévision de la consommation électrique française

Leonardo Martins Bianco - Guillaume Lambert

15/03/2021

Introduction

Ce projet étudie la modélisation et la prévision de la consommation électrique française de 2012 à début 2021. Les modèles de prédiction de cette donnée s'appuie principalement sur les données historiques météoroloqiques et temporelles. Cependant, avec l'apparition du COVID-19, la consommation électrique adopte un comportement différent et ces modèles de prédiction ne parviennent plus à la prédire correctement. En effet, de nouvelles variables entrent en jeu autres que la météo et le temps. Nous allons donc tenter de créer des modèles perfomants, y compris pour la période de COVID-19, notamment en utilisant de nouvelles données représentant cette période.

Dans un premier temps, nous allons importer et mettre en forme les données que nous allons utiliser. Ensuite, nous procéderons à une analyse descriptive de ces données ainsi qu'une analyse par composantes principales ACP, afin de comprendre les données et envisager la construction de modèles. Enfin, nous construirons différents modèles basées sur ces données.

Pour utiliser et partager le code sur différents ordinateurs, nous avons créé un package **packageProjet** contenant les datasets et une fonction, tous documentés.

Importation et mise en forme des données

Le jeu de données principalement utilisé est composé de l'indicateur **eCO2mix** issu du site internet du *Réseau de Transport d'Electricité* (**RTE**), l'opérateur de la distribution d'électricité en France. Il est également composé de l'indicateur **COVID-19 Government Response Tracker** publié par l'université d'Oxford. Il représente sous la forme d'un indice les mesures sanitaires prises par les états pendant la période du COVID-19.

On nomme **trainData** les données utilisées pour l'apprentissage et **testData** celles utilisées pour la prédiction. Voici une description des données :

- **Date** : date de mesure des données au pas de temps journalier. Elle s'étend du 01/01/2012 jusqu'au 15/04/2020 pour **trainData** et elle s'étend du 16/04/2020 jusqu'au 15/01/2021 pour **testData**.
- **Load** : consommation électrique du jour en MW
- **Load.1** : consommation électrique de la veille en MW
- **Load.7** : consommation électrique du jour de la semaine dernière en MW
- **Temp** : température moyenne en degrés Celcius sur 39 stations météorologiques en France
- **Temp_s95**, **Temp_s99** : températures lissées avec un paramètre 0.95, avec un paramètre 0.99
- **Temp_s95_min**, **Temp_s99_min** : températures minimales quotidiennes lissées avec paramètre 0.95, avec paramètre 0.99
- **Temp_s95_max**, **Temp_s99_max** : températures maximales quotidiennes lissées avec paramètre 0.95, avec paramètre 0.99
- **toy** : temps de l'année de 0 à 1 chaque année
- **WeekDays** : jours de la semaine

- **DLS** : changement d'heure
- **BH**, **Summer_break**, **Winter_break** : jours fériés et vacances d'été et d'hiver
- **GovernmentResponseIndex** : indice des mesures sanitaires

On vérifie que les données sont complètes. C'est le cas, on peut donc poursuivre par l'étude descriptive de nos données.

On ajoute les variables temporelles suivantes :

- **Mois** : mois de l'année de 1 à 12
- **JourDeLaSem** : numéro du jours dans la semaine de 1 à 7
- **weekend** : indicatrice qui indique si le jour de la semaine est un Samedi ou un Dimanche
- **Temps** : numéro de l'observation

On utilisera ces variables dans nos modèles mais elles ne seront pas toutes utiles et certaines se "superposent". Par exemple, **JourDeLaSem** et **weekend** sont utiles pour modéliser la temporalité hebdomadaire mais elles décrivent une information similaire. De même pour **Mois** et **toy**.

L'objectif de notre étude étant de prendre en considération la situation exceptionnelle du COVID-19, on souhaite avoir des variables la représentant. Nous avons déjà la variable **GovernmentResponseIndex** et nous souhaitons en ajouter de nouvelles. C'est pourquoi nous avons importé des données de l'INSEE sur la situation économique de la France. En effet, cette dernière a été fortement impactée par le COVID-19 et on se doute qu'elle est corrélée à la consommation électrique.

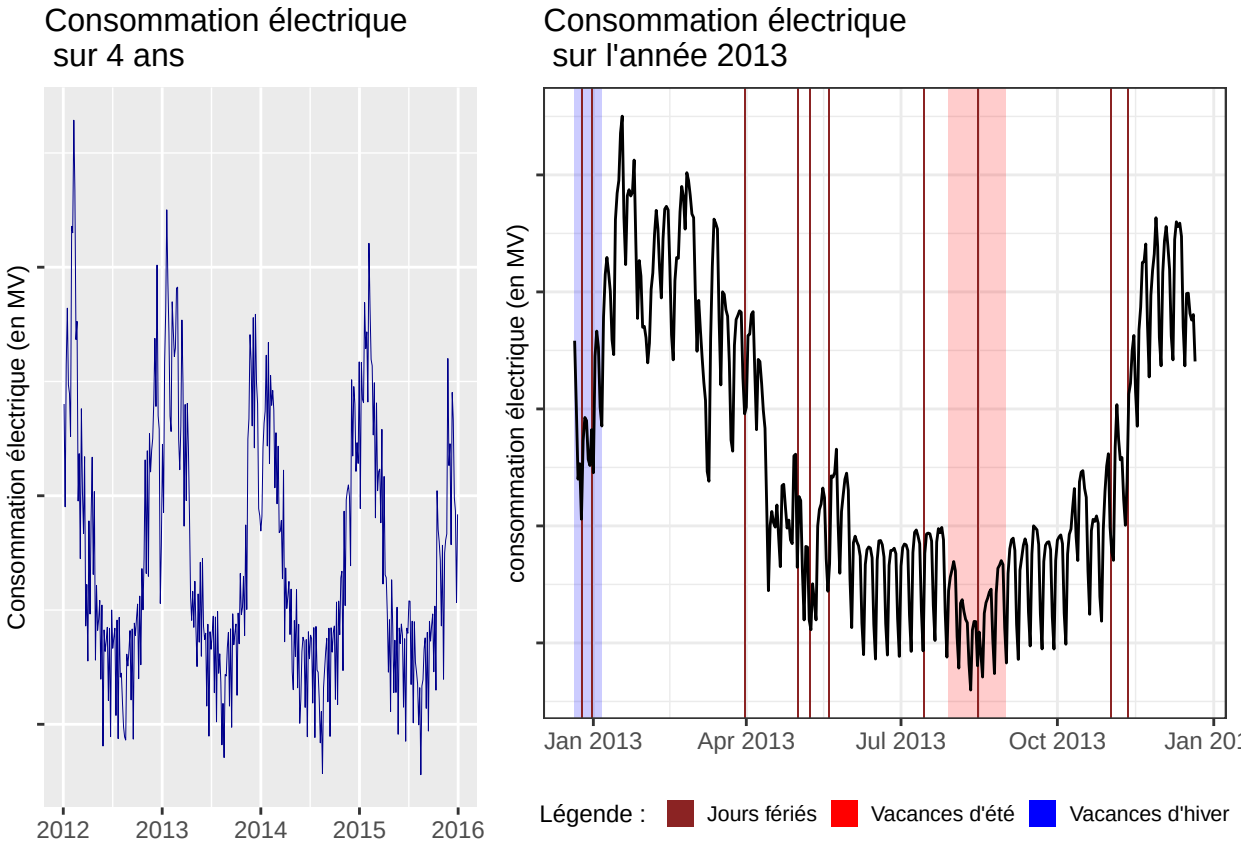
Le format des données ajoutées est trimestriel. Il est composé de différents indicateurs économiques et nous avons gardés les plus généraux pour ne pas importer trop de variables. Voici les variables que nous avons sélectionnées : **PIB**, **DepenseConsomMenages**, **DepenseConsomAPU** et **Exportations**.

Remarque : Les données de base sur Kaggle et les données économiques ont été jointes dans un fichier unique en *.CSV* puis intégrées à notre package. Ainsi, le code est utilisable avec le package uniquement sans avoir le fichier *.CSV*. D'ailleurs, la variable **weekend** est crée à partir de la fonction *weekend_indicatrice* du package.

Analyse des données

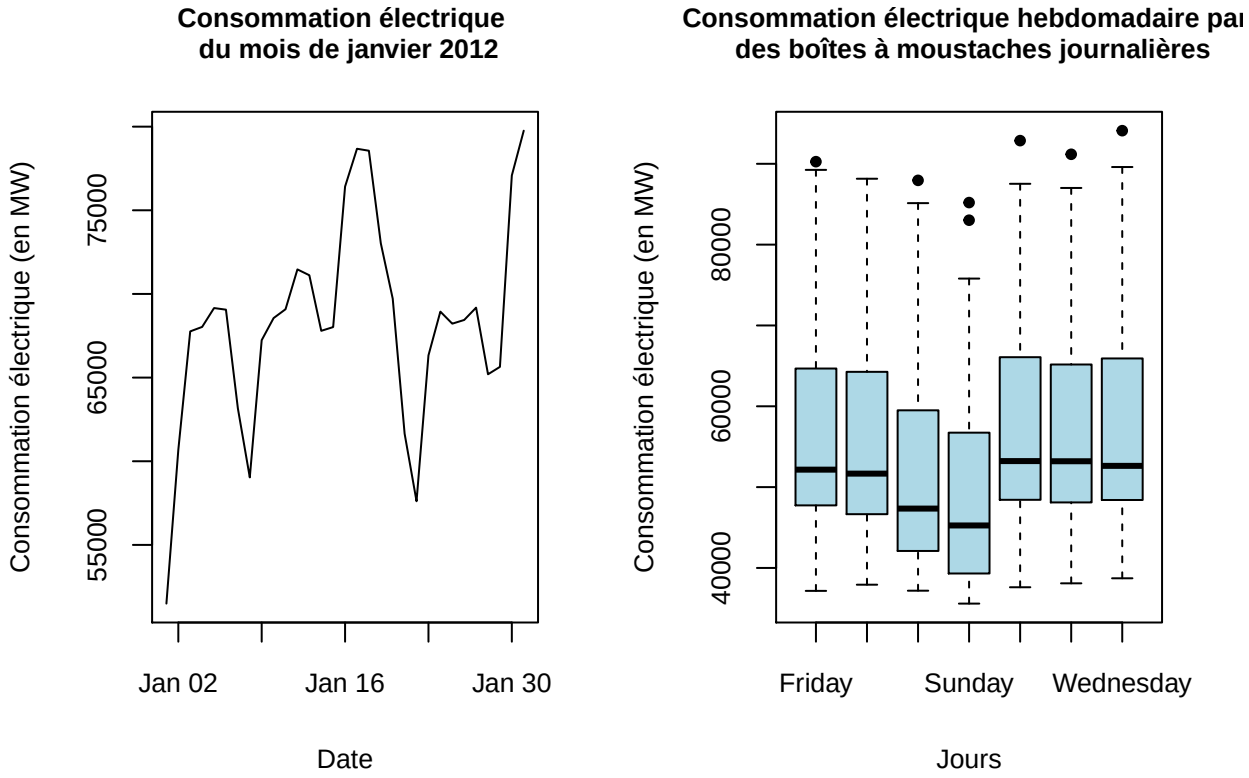
Analyse descriptive

On passe maintenant à l'étape de visualisation des données. Cela va nous permettre de les comprendre afin de créer des modèles pertinents. Tout d'abord, étudions la consommation **Load** par rapport au temps, afin de déterminer sa décomposition en tendance, saisonnalité et cycle.



Sur le graphique de gauche, on distingue un phénomène de saisonnalité annuelle très visible. On observe un pic maximal chaque année à l'hiver, ce qui correspond à l'utilisation d'électricité pour le chauffage. D'autre part, on voit un pic minimal chaque année à l'été, ce qui correspond à l'utilisation moindre de chauffage. Enfin, on ne distingue pas de tendance sur le long terme.

Sur le graphique de droite, on représente la consommation électrique sur l'année 2013 afin d'observer les facteurs temporels influant la consommation élctrique au sein d'une année. On retrouve la corrélation entre la saison et la consommation électrique comme sur le graphique de gauche, avec de plus une légère augmentation en juillet, que l'on interprète par l'utilisation de la climatisation. On remarque aussi que les vacances influencent grandement la consommation électrique avec une franche diminution locale de la consommation. On observe également l'influence des jours fériés où la consommation éléctrique diminue, surtout au mois de Mai car c'est le mois où il y en a le plus. On se dit intuitivement que ces baisses de consommation sont reliées à la baisse de l'activité et à l'arrêt du travail. Enfin, on observe une saisonnalité hebdomadaire que l'on observe plus facilemment sur les graphiques suivants :

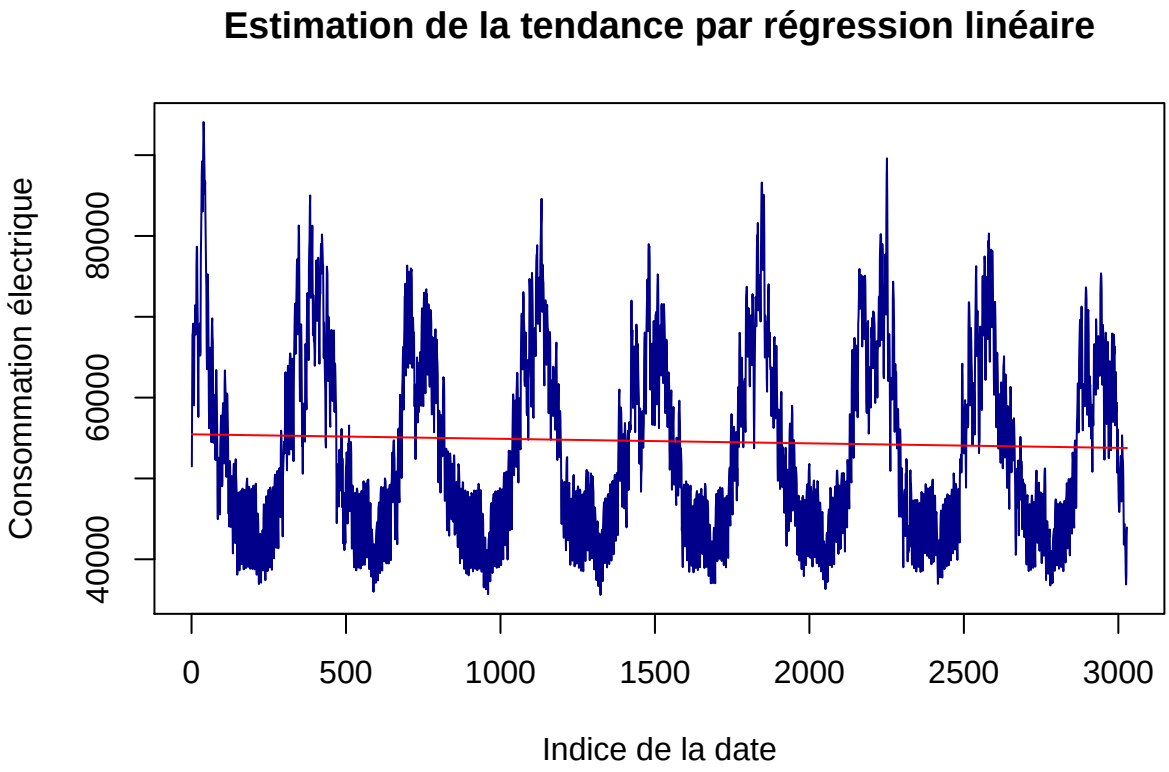


Sur le graphique de gauche, on observe bien une saisonnalité hebdomadaire avec une baisse de la consommation électrique les weekends. On interprète cela par la diminution de l'activité le weekend par rapport aux autres jours de la semaine.

Sur le graphique de droite, on retrouve la saisonnalité hebdomadaire. De plus, on voit que les jours hors weekend sont presque identiques d'un point de vue de la consommation électrique. Lors du weekend, cette dernière est plus faible le dimanche que le samedi, ce qui semble logique car l'activité est encore plus faible le dimanche que le samedi. Enfin, on retrouve une répartition en quartiles quasi similaire pour chaque jour de la semaine avec une médiane inférieure à la moyenne.

Nous venons d'étudier la consommation électrique par rapport au temps. Nous n'avons pas déterminé visuellement une tendance mais nous avons trouvé 2 saisonnalité : une annuelle et une hebdomadaire. On a également observé l'impact des vacances et des jours fériés. Cela motive l'utilisation des variables temporelles de base, ainsi que de la création de nouvelles variables temporelles (**weekend**, **JourDeLaSem**, **Mois**). Cette étude temporelle a aussi permis d'observer l'impact de la température (saisonnalité annuelle) et de l'activité (vacances, jours fériés et saisonnalité hebdomadaire).

On détermine la tendance de **Load** à l'aide d'un modèle linéaire à 2 dimensions, c'est-à-dire un modèle affine. La pente est significative dans le test de Student et vaut -0.5594. La tendance est donc très légèrement à la baisse et on le représente graphiquement :



Enfin, pour ne pas nous attarder sur l'analyse descriptive, nous ne représentons pas les graphiques de la consommation électrique en fonction d'autres variables. Cependant, nous avons quand même mis en lumière l'impact des variables non temporelles comme la température sur la consommation électrique. Nous continuons l'analyse de nos données par une analyse des composantes principales.

Analyse des composantes principales

L'analyse des composantes principales (ACP) permet de résumer et de visualiser l'information contenue dans un ensemble de données qui peut être composé de nombreuses variables, corrélées les unes avec les autres. C'est exactement notre cas. Une ACP permet d'extraire une grande partie de l'information de nos données et de la représenter selon de nouvelles variables appelées composantes principales. Ces dernières sont des combinaisons linéaires des variables de base. L'objectif d'une ACP est de déterminer les directions le long desquelles la variation des données est maximale, c'est-à-dire les directions où il y a le plus d'information. Ainsi, on réduit les dimensions des données tout en gardant le plus d'information possible.

On peut aussi expliquer une ACP comme la recherche d'une suite de p vecteurs tels que le i-ème vecteur est celui qui approche le mieux les données en étant orthogonal aux $i-1$ vecteurs restants. Ce vecteur est le "meilleur" dans le sens que la droite engendrée par ce vecteur est celle qui minimise un problème de moindres carrées.

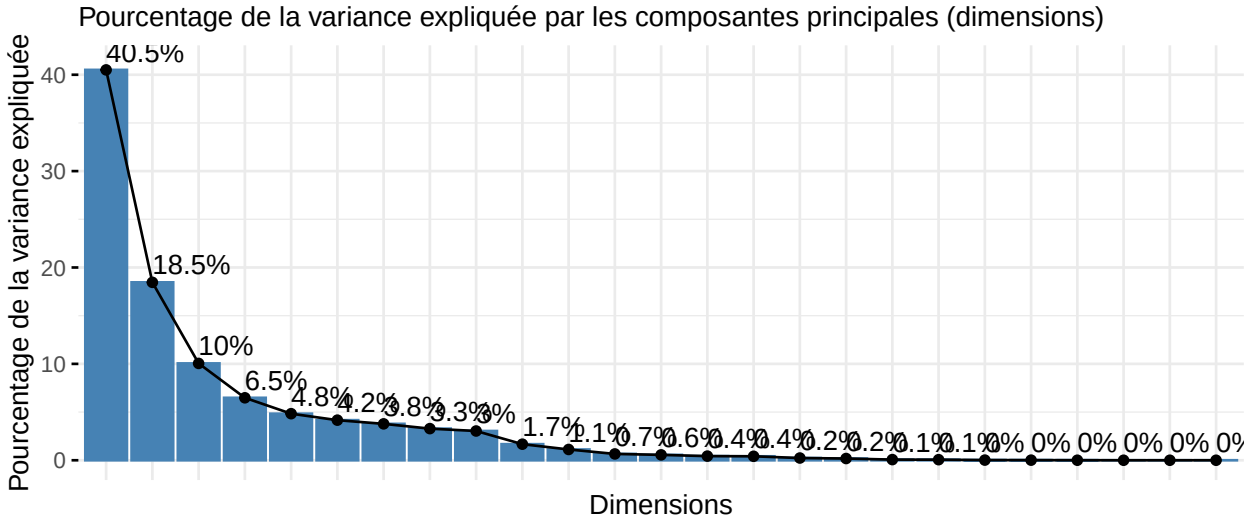
Nous allons nous concentrer sur les 3 premières composantes principales. Elles sont orthogonales et vont représenter une grande partie de la variance dans notre cas. Techniquement, la quantité de variance représentée par une composante principale est mesurée par la valeur propre associée à un vecteur p. Cette interprétation géométrique va nous servir à comprendre le graphique descriptif sur la corrélation des variables.

Avant de procéder à l'ACP, on transforme nos variables $(x_i)_i$ selon la formule suivante :

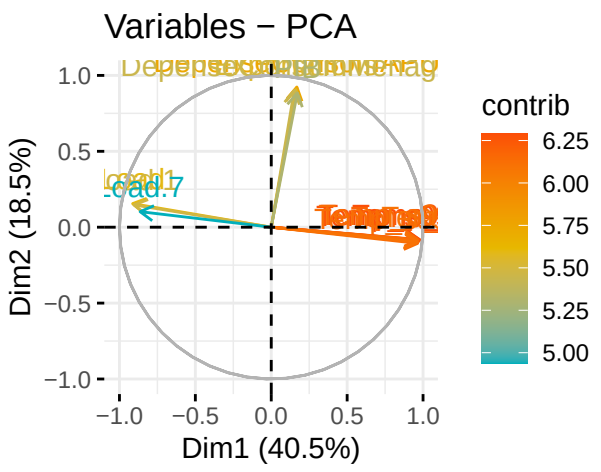
$$x_i \mapsto \frac{x_i - \text{moyenne}(x)}{\text{sd}(x)}$$

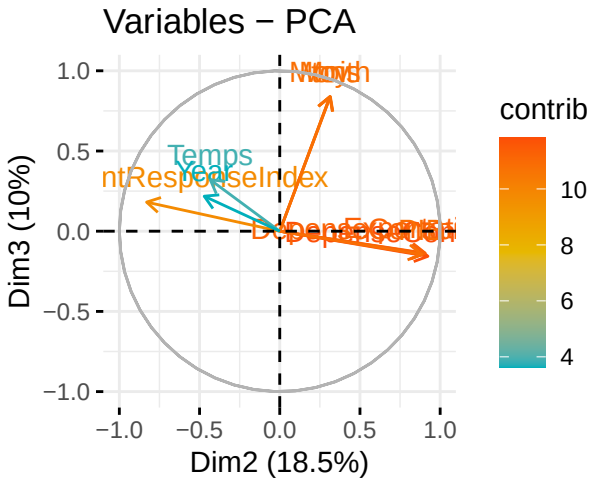
Les données deviennent donc centrées et réduites. Ainsi, elles deviennent "comprables" entre elles, alors que leurs échelles initiales sont complétement différentes. D'autre part, on retire les données non-numériques du jeu de données pour l'ACP. On justifie ici la création de la variable **JourDeLaSem** qui est une transformation numérique de la variable non-numérique **WeekDays**. Cette standardisation est appliquée par la fonction **PCA** si on utilise le paramètre *scale.unit = TRUE*.

On commence notre ACP en déterminant la variance expliquée par les différentes composantes principales calculées par la fonction **PCA**. On représente visuellement ce résultat :



On remarque que la variance expliquée des 3 premières composantes principales totalisent 69% de la variance totale. On observe également qu'à partir de la 10ème dimension, la variance expliquée par les dimensions postérieures est inférieure à 1% de la variance totale. L'ACP nous indique donc qu'il y a une forte corrélation entre les données et qu'une réduction de la dimension est possible. Dans notre cas, on ne souhaite pas faire de réduction mais on souhaite voir quelles sont les variables les plus importantes et celles qui sont corrélées. On représente donc les 2 graphiques suivants, qui représentent la contribution des variables (la longueur des vecteurs) et les relations de corrélation (l'orientation des vecteurs), pour les 3 premières composantes principales.





Etudions les 3 premières dimensions.

Première dimension : elle est principalement composée de 2 groupes de variables concentrées : les variables mesurant la température et les variables mesurant la consommation électrique (**Load**, **Load.1** et **Load.7**). Les variables des 2 groupes sont très fortement corrélées entre elles, ce qui est attendu car chaque groupe mesure une seule quantité. D'autre part, les variables des 2 groupes sont corréles négativement, ce qui est

aussi attendu car quand la température diminue, la consommation électrique augmente et vice versa.

Deuxième dimension : on peut distinguer un groupe de variables concentrées : les variables économiques **PIB**, **DepenseConsomMenages**, **DepenseConsomAPU** et **Exportations**. Elles sont fortement corrélées positivement entre elles. Le groupe est corrélé négativement avec **GovernmentResponseIndex** et dans une moindre mesure avec **Year** et **Temps**. Cela est attendu car les variables économiques mesurent la même chose et plus le temps avance, plus le COVID progresse et plus la situation économique se détériore.

Troisième dimension : elle est principalement composée des variables temporelles **toy** et **Mois** qui sont fortement corrélées entre elles. De plus, on voit que les variables **Temps** et **Year** sont légérement orientées selon cette dimension. C'est attendu car ce sont également des variables temporelles.

En résumé, les 3 composantes principales sont la consommation électrique antérieure et la température, la situation économique et sanitaire, et la temporalité. Cela confirme notre intuition sur les données et on utilise ces différents groupes de variables pour construire nos modèles. Notamment, on essayera d'utiliser les variables de chaque dimension, sans faire trop de doublon.

Construction de modèles

Modèle GAM

L'hypothèse de départ dans la construction de notre modèle GAM est que l'on peut exprimer la consommation électrique en période de COVID-19 sous la forme d'un modèle additif composé de 2 éléments. Le premier élément est la consommation électrique "normale", c'est-à-dire hors période de COVID-19. On modélisera donc ceci grâce aux variables classiques : température, variables temporelles et consommation électrique antérieure. Le deuxième élément est la pertubation particulière de la situation sanitaire sur la consommation électrique. On la modélisera donc grâce à la variable **GovernmentResponseIndex** et les variables économiques.

On adoptera la stratégie suivante :

- On découpe le dataset **trainData** en 2 : **trainDataReduite.1** correspond à la période hors-COVID-19 de 2012 à 2019 et **trainDataReduite.2** correspond à la période COVID-19.
- On construit un modèle GAM *model.GAM.1* pour modéliser la consommation électrique sur les données **trainDataReduite.1** et on la prédit sur les données **trainDataReduite.2**.
- La variable **Load** de **trainDataReduite.2** étant connue, on calculera les résidus **residus**. Ils correspondent à la perturbation particulière dûe à la crise sanitaire que notre modèle classique n'arrive pas à calculer.
- On contruit un modèle GAM *model.GAM.2* pour modéliser **residus** sur les données **trainDataReduite.2** et prédire les résidus sur les données **testData**.
- Enfin, étant donné que l'on a supposé que la consommation électrique est un modèle additif, on somme les 2 prédictions.

Modèle GAM en situation hors-COVID

```
model.GAM.1 <- gam(Load ~ s(JourDeLaSem,bs='cc', k = 7) +
                          Christmas_break + BH +
                          s(Load.1,bs='cr',k=10) + s(Load.7,bs='cr',k=10) +
                          s(Temp,bs='cr',k=50),
                          data = as.data.frame(trainDataReduite.1))
```

Voici le modèle GAM que l'on a construit pour modéliser la partie hors-COVID. Afin de le construire, nous avons utilisé les variables significatives dans l'étude descriptive et l'ACP de nos données. Nous avons aussi observer l'influence de chaque variable de manière empirique en comparant les scores du Kaggle, le GCV et le

$R^2 - ajusté$. Ce modèle utilisé sur l'ensemble des données **trainData** donne un score Kaggle de 1004.98513. On l'a calculé après la clôture de la compétition.

Analysons ce modèle. Pour les variables temporelles, nous avons choisi **JourDeLaSem**, **Christmas_break** et **BH**.

- La variable **JourDeLaSem** permet la modélisation de la saisonnalité hebdomadaire et la baisse de la consommation électrique le samedi puis le dimanche. Elle se révèle plus pertinente que l'indicatrice **weekend** car elle prend plus de valeurs (7 contre 2) et possède plus d'information pertinente que **weekend**. Afin de modéliser la relation non-linéaire entre **JourDeLaSem** et **Load**, on utilise une fonction *s()* et la base de splines utilisée est *cc* car **JourDeLaSem** représente une saisonnalité. Le paramètre k est maximal pour laisser le plus de degré de liberté possible, étant donné que 7 est petit.
- La variable **Christmas_break** joue le rôle d'une indicatrice et intervient donc sous la forme d'une fonction linéaire. Elle permet de modéliser la forte baisse de la consommation d'électricité à la fin de l'année, comme on a pu le voir sur le graphique de l'année 2013. On peut se demander pourquoi **Christmas_break** est utilisée et non **Summer_break** (l'utilisation de **Summer_Break** est néfaste pour les scores). On pense que c'est parceque la baisse de la consommation électrique est significativement plus importante pendant les vacances d'hiver que pendant les vacances d'été. On pense également que les vacances d'été sont modélisées par d'autres variables comme **Temp**.
- La variable **BH** étant elle-aussi une indicatrice, elle intervient sous la forme d'une fonction linéaire. Elle permet de modéliser la baisse de la consommation électrique les jours fériés. On a longtemps négligé cette variable alors que c'est elle qui a baissé de manière très significative notre score Kaggle de 1300 à 1000 environ. On pense que son atout majeur est qu'elle est la seule variable à modéliser la baisse de la consommation pendant les jours fériés, notamment au mois de Mai, alors qu'aucune des autres variables ne peut le faire. De plus, cette variable est une simple indicatrice donc elle ne complexifie pas beaucoup le modèle GAM.

Parmi les variables mesurant la température, on utilise uniquement **Temp** car elles sont toutes fortement corrélées, comme on l'a vu dans l'ACP. On a testé toutes les variables mesurant la température et on a pris celle qui nous donne le meilleur résultat. La relation entre **Temp** et **Load** étant non-linéaire, on utilise une fonction *s()* avec une base de splines *cr*. On limite le paramètre k à 50 pour limiter le temps de calcul et pour ne pas faire de surapprentissage. On sait que la température influence directement la consommation électrique par l'utilisation de chauffage et de climatisation. Ce que l'on peut ajouter, c'est que la température permet de modéliser la saisonnalité annuelle de la consommation électrique. En effet, on peut considérer cette saisonnalité comme étant basée sur la température. C'est pourquoi on n'utilise pas ici les variables **toy** et **Month**.

Enfin, les deux dernières variables utilisées sont **Load.1** et **Load.7**. Elles donnent de l'information sur la consommation électrique antérieure et on peut se dire intuitivement que cette information est précieuse pour prédire la consommation électrique future. La relation entre ces 2 variables et **Load** étant non-linéaire, on utilise une fonction *s()* avec une base de splines *cr*. On limite le paramètre k à 10 pour les mêmes raisons que pour la variable **Temp**.

En résumé, nous avons utilisé 2 ou 3 variables présentent dans les 3 premières composantes principales de l'ACP, hormis les variables sur la situation économique et sanitaire.

Modèle GAM en situation COVID-19

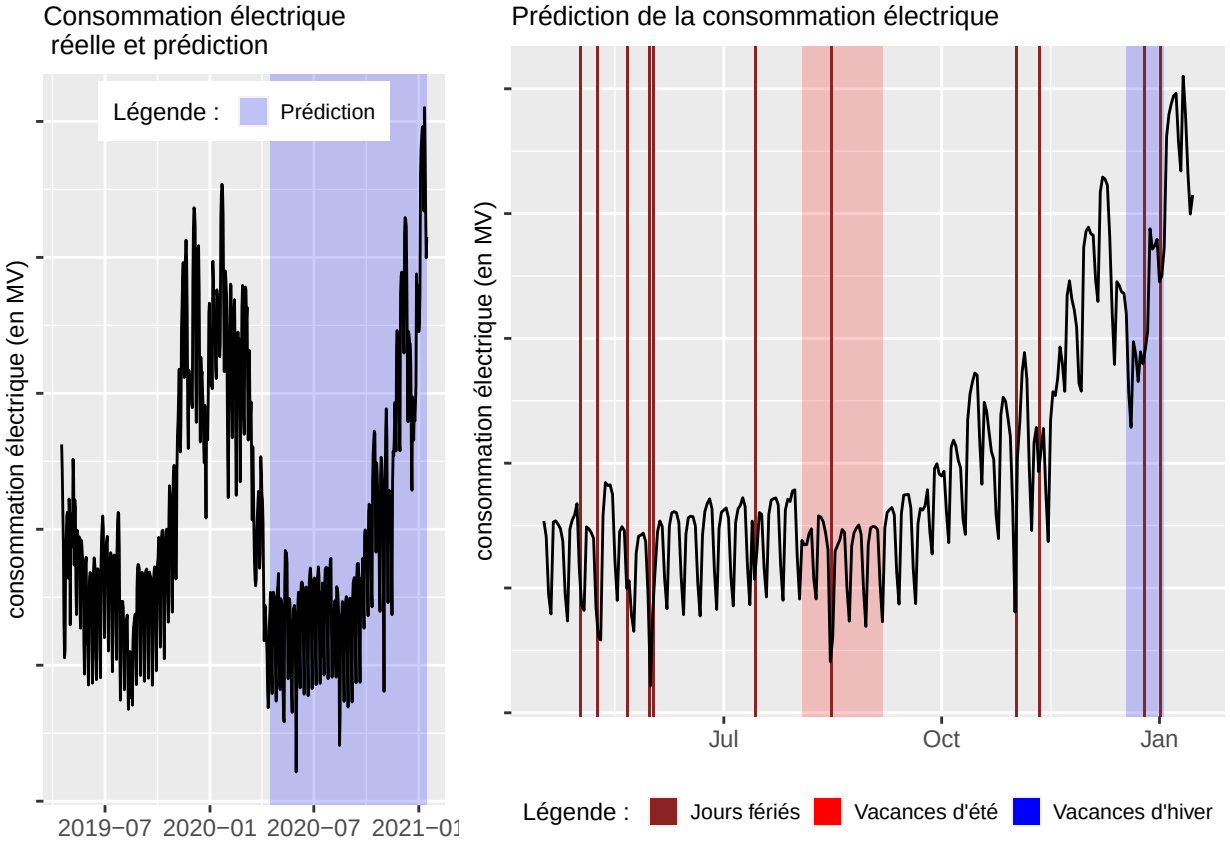
On cherche maintenant à construire un second modèle pour modéliser les résidus du premier modèle sur la période COVID-19. Pour construire ce second modèle, nous avons utilisé les variables **GovernmentResponseIndex** et celles économiques. Nous avons construit 2 types de modèle pour remplir cet objectif : un modèle GAM et une forêt aléatoire. Cependant, nous n'avons pas réussi à obtenir de résultats satisfaisants, c'est-à-dire augmentant le score du Kaggle.

Nous avons remarqué que la variable la plus significative dans le test de Student du modèle GAM est **GovernmentResponseIndex**, alors que les variables économiques sont très peu voire non significatives.

Cela confirme l'orthogonalité observée dans l'ACP. On pense que c'est à cause du format trimestriel des données économiques, qui est un format avec une échelle trop grande par rapport au format quotidien des variables **GovernmentResponseIndex** et **Load**. Sur le dataset de prédiction, les données économiques ne prennent que 4 valeurs différentes, ce qui est trop peu pour modéliser leur impact et le nombre de noeuds dans la base de splines est très faible (inférieur à 4). Enfin, on constate que les scores GCV et $R^2 - ajusté$ sont très mauvais dans le second GAM, donc il ne permet pas de modéliser correctement la pertubation dûe au COVID-19.

Afin de résoudre ce problème, on pourrait ajouter d'autres variables caractérisant la situation particulière du COVID-19, au format quotidien. D'autre part, la source de ce problème vient peut être de notre hypothèse sur la nature additive de la consommation électrique en période de COVID-19, c'est-à-dire l'hypothèse initiale de notre stratégie.

Le modèle que l'on sélectionne est donc le modèle **model.GAM.1** basé sur les données **trainData**. On effectue la prédiction et on la représente :



On voit que la prédiction semble se comporter de la même manière que les données antérieures. On observe des valeurs faibles en été et des valeurs hautes en hiver. La saisonnalité annuelle semble donc être respectée et on observe bien la saisonnalité hebdomadaire.

De plus, les périodes de vacances d'été et d'hiver marquent une baisse significative de la consommation électrique, comme observé dans les données réelles. On remarque cependant que la baisse pendant les vacances d'été est assez faible. Or nous n'avons pas utilisé la variable **Summer_break** et on peut se demander s'il serait pertinent de l'ajouter. Après différents tests, on remarque que les résultats deviennent moins bons et que l'aspect de la prévision lors des vacances d'été ne change pas. Cela est dû à l'influence d'autres variables qui ont caractérisées cette période à la place de **Summer_break** et donc elle n'apporte pas d'information supplémentaire. D'autre part, on remarque que la consommation électrique baisse également lors des jours fériés, mais là encore de façon moins voyante que dans les données observées. On peut justifier ces différences par le fait que notre modèle ne soit pas assez performant ou aussi par l'impact particulier de la situation

sanitaire.

Dans l'ensemble, on est satisfait de ce modèle GAM qui nous fournit une prédiction visuellement satisfaisante et performante (score Kaggle : 1004.98513).

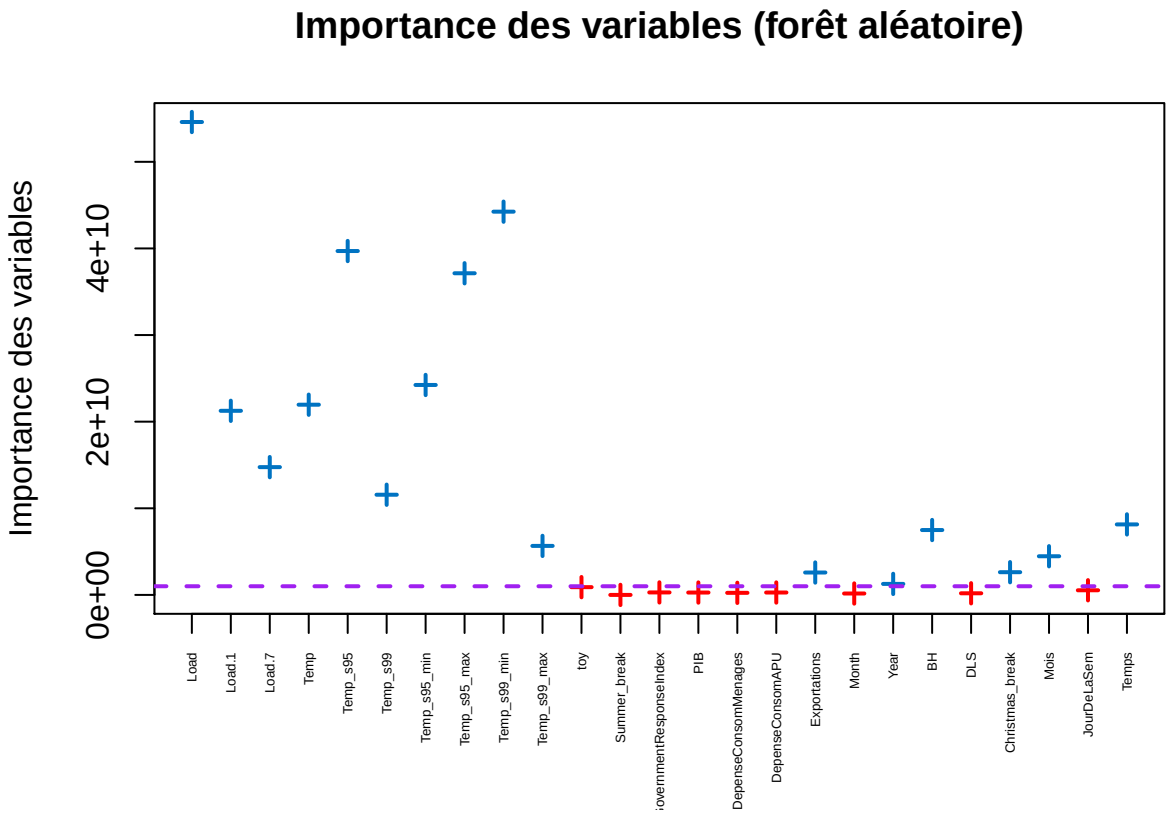
Nous n'avons donc pas utilisé la variable **GovernmentResponseIndex** et les variables économiques de l'INSEE, notamment car leur ajout détériore les différents scores. En effet, la variable **GovernmentResponseIndex** est nulle sur la grande partie du jeu de données et elle entre en jeu uniquement lors de la période COVID-19. Il ne semble donc pas pertinent de l'utiliser dans un modèle basé sur des données historiques annuelles où elle n'intervient qu'à la toute fin. Concernant les variables économiques, on pense qu'elles ne sont pas pertinentes car elles représentent des phénomènes de basse fréquence contrairement à **Load** qui incluent des phénomènes de haute fréquence. En effet, il est impossible de corréler la saisonnalité hebdomadaire de **Load** avec des données trimestrielles et il semble très compliqué de le faire pour la saisonnalité annuelle (4 valeurs pour 1 saisonnalité). Une manière d'utiliser ces données trimestriels serait de transformer la variable **Load** sous forme trimestriel également et ainsi observer la corrélation entre des données de même échelle.

On continue la création d'autres modèles sans séparer la période COVID-19. On crée d'abord un modèle de forêt aléatoire, ensuite un modèle XGBoost, et à la fin on fait "l'agrégation d'experts" de ces modèles en utilisant le package `opera`. On note que pour faire l'agrégation d'experts, on devra séparer le jeu de données d'entraînement en deux parties : une pour l'entraînement des données (80% des observations), une autre pour la validation des prévisions (20%). Dans cette séparation, la période de l'épidémie n'est pas comprise dans les données d'entraînement.

Forêt aléatoire

Notre première approche est la méthode des forêts aléatoires. On les implémente à l'aide du package `ranger`.

Les forêts aléatoires sont capables de nous indiquer les variables le plus importantes dans le modèle, donc d'abord on crée un modèle avec toutes les variables, on regarde quelles sont les plus importantes, et après on crée un deuxième modèle avec seulement les variables significatives. Voici le graphe qui indique l'importance des variables.



Dans ce graphe, on voit que l'on a des variables avec importance d'ordre 10^{10}. Ainsi, on retire de l'analyse toutes les variables qui ont un ordre de grandeur plus petit. La ligne violette représente 10^9, et on voit que les variables qui non significatives sont `toy`, `Summer-break`, `GovernmentResponseIndex`, `PIB`, `DepenseConsomMenages`, `DepenseConsomAPU`, `Exportations`, `Month`, `DLS`, `JourDeLaSem`.

On crée alors un deuxième modèle en retirant ces variables. On obtient avec ce modèle un score de 2159.88 sur Kaggle.

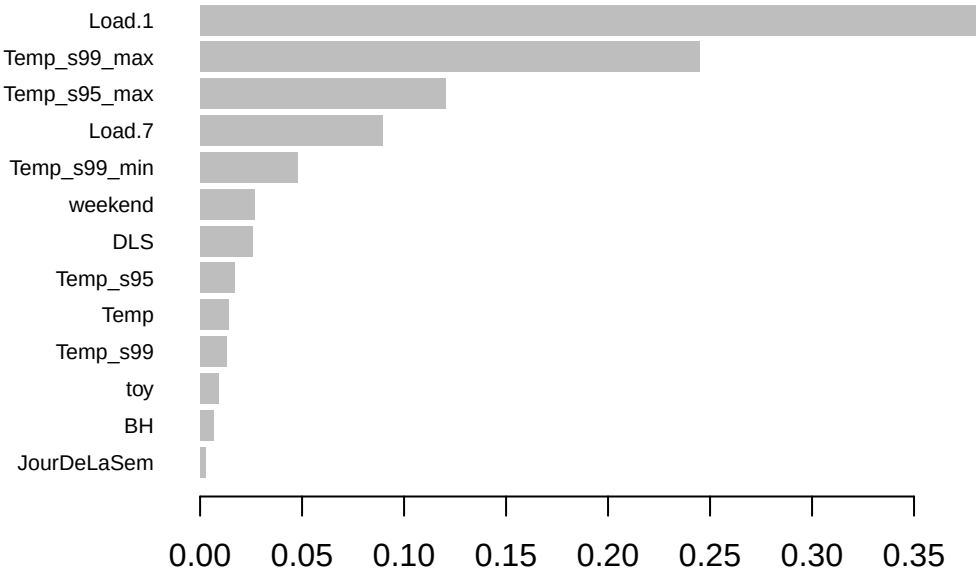
Une manière d'améliorer ces résultats est d'utiliser un XGBoost, qui est similaire à une forêt aléatoire dans le sens qu'il entraîne plusieurs arbres de décision, mais différent dans le sens que ces arbres ne sont pas indépendants. Dans le XGBoost, un arbre est entraîné sur les résidus du précédant. Ainsi, si les paramètres sont bien choisis, ce modèle est censé être plus performant que le modèle de forêt aléatoire. Un désavantage de cette approche est qu'elle est sensible au choix des paramètres, et peut facilement surapprendre les données d'entraînement. Les forêts aléatoires sont plus robustes à ces deux aspects.

XGBoost

On va créer un modèle XGBoost en utilisant le package `xgboost`.

Cette méthode est assez sensible aux paramètres choisis, ainsi on entreprend une approche rigoureuse pour son choix. On choisit une plage de valeurs possibles pour chaque paramètre du modèle, et après on crée une matrice avec toutes les combinaisons possibles des valeurs de ces paramètres. On entraîne un modèle `xgboost` pour chaque combinaison possible des paramètres, et à la fin on prend ceux qui nous donnent le meilleur modèle (en RMSE prédictive, i.e., l'erreur sur la partie de validation du jeu de données, pas l'erreur d'entraînement). On note que c'est une procédure assez coûteuse. La première utilisation de `xgboost`, dédiée à l'identification des variables importantes, a pris environ 1 heure pour s'exécuter. Une fois que les variables ont été sélectionnées, la deuxième utilisation (en retirant les variables non significatives) a pris environ 30m pour s'exécuter.

Voici le graphe de l'importance des variables :

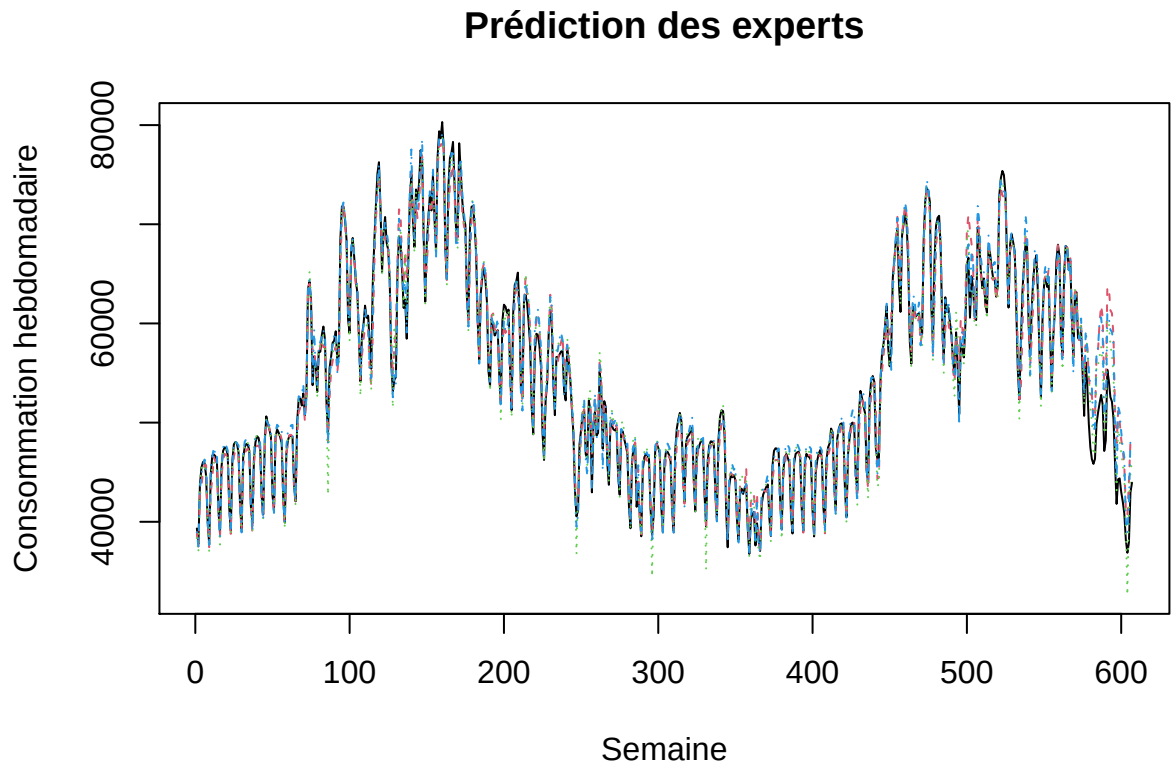


Les paramètres finaux trouvés sont `eta` = 0.05 (taille du pas de descente de gradient), `max-depth` = 5 (profondeur maximale des arbres à chaque pas), `min-child-weight` = 5 (plus ce paramètre est grand, plus l'algorithme est "conservatif"), `subsample` = 0.8 (si on réalise une descente de gradient stochastique), `colsample-bytree` = 0.8 (pourcentage des variables que l'on utilisera pour entraîner chaque arbre à chaque pas, choisies aléatoirement), et le nombre optimal d'arbres est 476. Le RMSE minimal pour ce modèle est 902.8773. Sur Kaggle, ce modèle nous donne un score de 1665.61044. On voit bien une amélioration par rapport au résultat obtenu avec la forêt aléatoire.

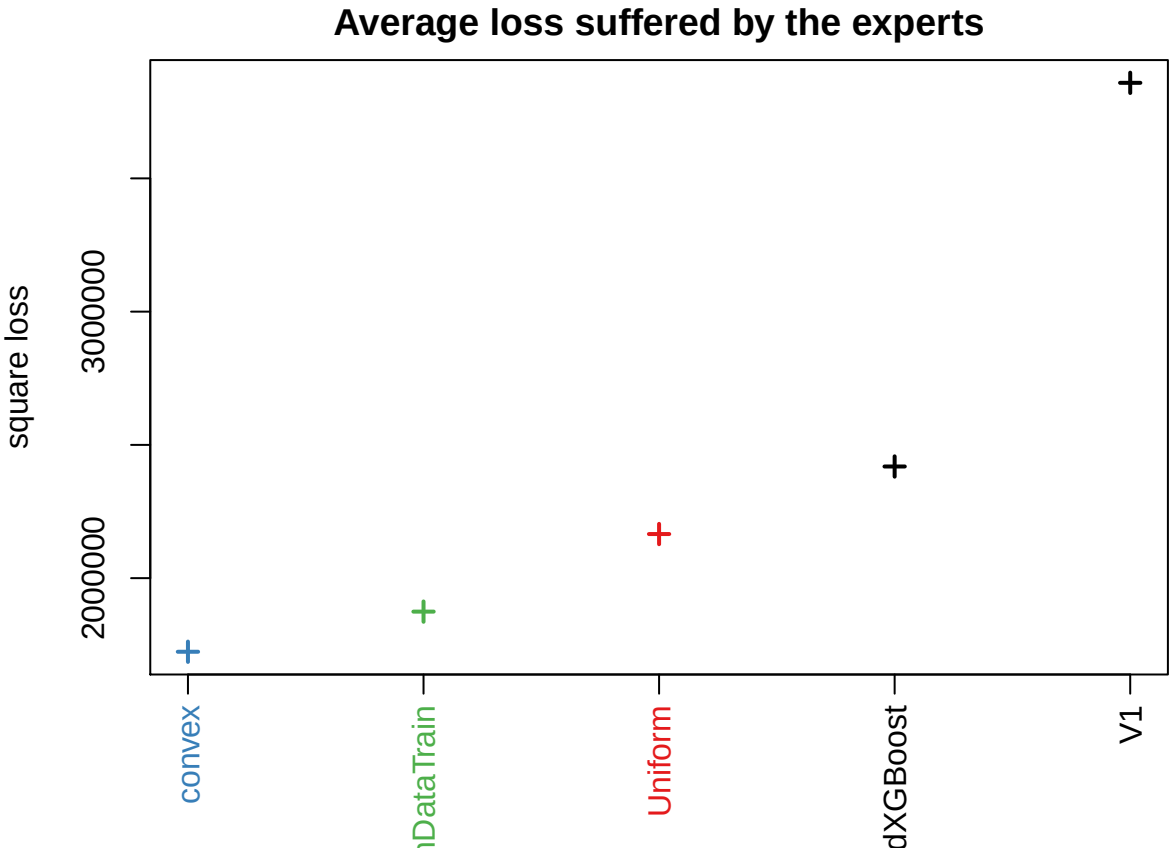
Agrégation d'experts

Une fois que l'on a crée des modèles, on veut les unifier pour faire une prédiction globale au lieu d'utiliser un modèle ou l'autre. Cette procédure s'appelle agrégation de modèles, et elle peut augmenter la puissance prédictive de notre modèle. Une approche récente est l'agrégation d'experts, où on fait une combinaison convexe des prédictions de chaque modèle, et où le coefficient de la prédiction du k-ème modèle est donné par sa performance. Au niveau logiciel, le package `opera` automatise et optimise les algorithmes qui réalisent cette procédure.

Le graphe ci-dessous montre la prédiction individuelle de chaque modèle sur dataset d'entraînement avec forêt aléatoire en bleu, XGBoost en vert, GAM en rouge et la variable **Load** en noir.



Voici un graphe comparant l'erreur de chaque modèle.



On observe que la forêt aléatoire, noté `V1` dans l'axe des abscisses, est beaucoup moins performante que

le XGBoost, qui est un peu moins performant que le GAM. Le point associé à `Uniform` dans le graphe correspond à la combinaison convexe qui donne des poids égaux aux trois modèles, i.e., c'est une moyenne. `convex` correspond à la meilleure combinaison convexe trouvée pour ces modèles. On voit qu'elle améliore le résultat de chaque modèle individuel.

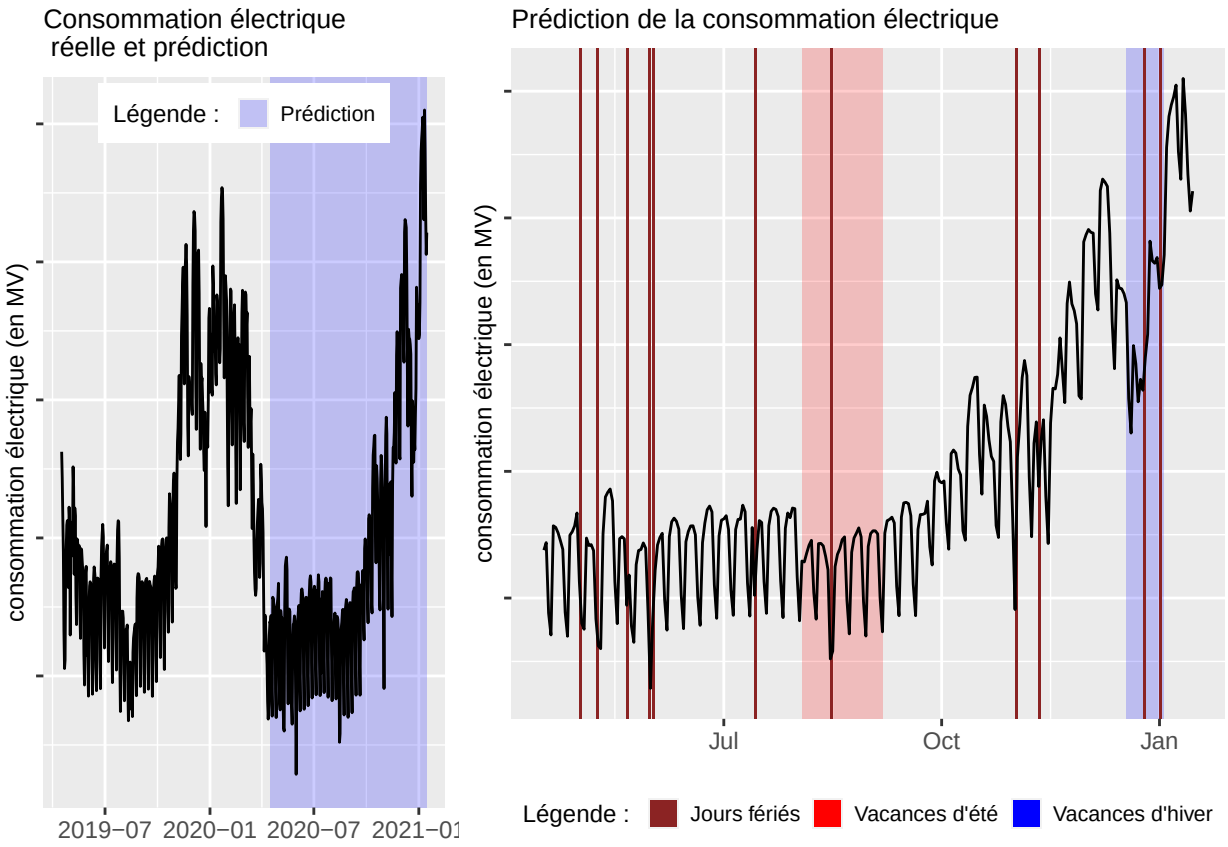
Observons les coefficients associées à chaque modèle dans la meilleure combinaison convexe. On trouve 0 pour la forêt aléatoire, c'est-à-dire, la forêt aléatoire ne doit pas être incluse dans l'agrégation des modèles, elle n'apporte aucune contribution, ce que l'on avait déjà prévu lorsqu'on a introduit XGBoost. Ensuite, on a 0.722 pour le GAM. C'est-à-dire qu'il entre dans le modèle final, et comme il est plus performant que le XGBoost, son coefficient est proportionnellement plus grand. Comme la somme est convexe, le pois restant de 0.278 est associé au modèle XGBoost. Finalement, le RMSE associé au meilleur oracle convexe est 1330.

Notre score Kaggle avec une prévision donnée par une telle agrégation est 1169.63972 alors que le score Kaggle pour le modèle GAM seul est proche de 1000. Cette agrégation n'a donc pas amélioré notre score.

Le package `opera` fournit d'autres manières de construire les agrégations. Cela change en fonction du type de données que l'on a. Le package est adapté pour travailler aussi avec des données "en ligne", c'est-à-dire, des données mises-à-jour en temps réel. L'approche utilisée ci-dessus n'est pas optimale dans notre cas et on en utilise une autre.

On peut aussi faire une agrégation en utilisant la méthode de régression Ridge. Dans ce cas, les coefficients trouvés sont -0.268 pour la forêt aléatoire, 0.737 pour le GAM, et 0.525 pour le XGBoost. La somme des coefficients est toujours 1, cependant ils ne sont pas tous positifs, donc on ne peut pas les interpréter comme le poids de chaque modèle. Le RMSE obtenu avec la régression ridge est 1280, et le score Kaggle est 968.75625. Cela montre finalement que l'agrégation peut donner de meilleurs résultats que le résultat individuel de chaque expert.

Ce modèle nous donne notre meilleur score. On le choisit donc pour notre prédiction finale que l'on représente :



Le résultat est visuellement satisfaisant et on peut faire la même interprétation graphique qu'avec le graphique similaire issu des données prédites par le modèle GAM seul.

Conclusion

Dans ce projet nous avons étudié la consommation électrique de la France, et en particulier comment elle a changé à cause de l'épidémie du Coronavirus.

Tout d'abord, on a fait le traitement et l'étude des données pour trouver quelles sont les plus importantes, les corrélations existantes entre elles, et le comportement général de la consommation électrique en fonction de chaque variable. Pour cela, on a fait une analyse descriptive et une analyse des composantes principales (ACP).

Après que l'on ait examiné les données, on a créé différents modèles. Notre première approche était un modèle GAM en deux étapes : un pour la période hors COVID-19, et un pour les résidus du premier modèle en fonction des variables économiques et l'index sur la situation sanitaire. Cependant, le deuxième GAM n'a pas pu être bien construit et on a sélectionné un modèle GAM utilisant uniquement la température, la consommation électrique antérieure et le temps.

La solution trouvée à ce problème était d'entraîner différents modèles et faire l'agrégation de ses résultats pour minimiser l'erreur qui vient de l'utilisation de la période hors COVID-19 pour modéliser la période avec COVID-19.

Pour construire l'agrégation, on a d'abord construit des forêts aléatoires, un XGBoost (avec un "grid search" pour trouver les bons paramètres), et on les unifie avec notre GAM déjà introduit. Pour l'agrégation de ces modèles, on a utilisé le package `opera`, qui nous permet de faire une agrégation par une somme convexe ou par une régression Ridge. L'agrégation avec somme convexe ne nous donne pas un meilleur résultat que le GAM seul, mais celle avec régresion Ridge améliore notre score Kaggle. Ce modèle d'agrégation parvient donc de construire un modèle plus performant que les modèles de base individuellement. C'est donc ce modèle que l'on choisit parmi tout ceux de notre projet.