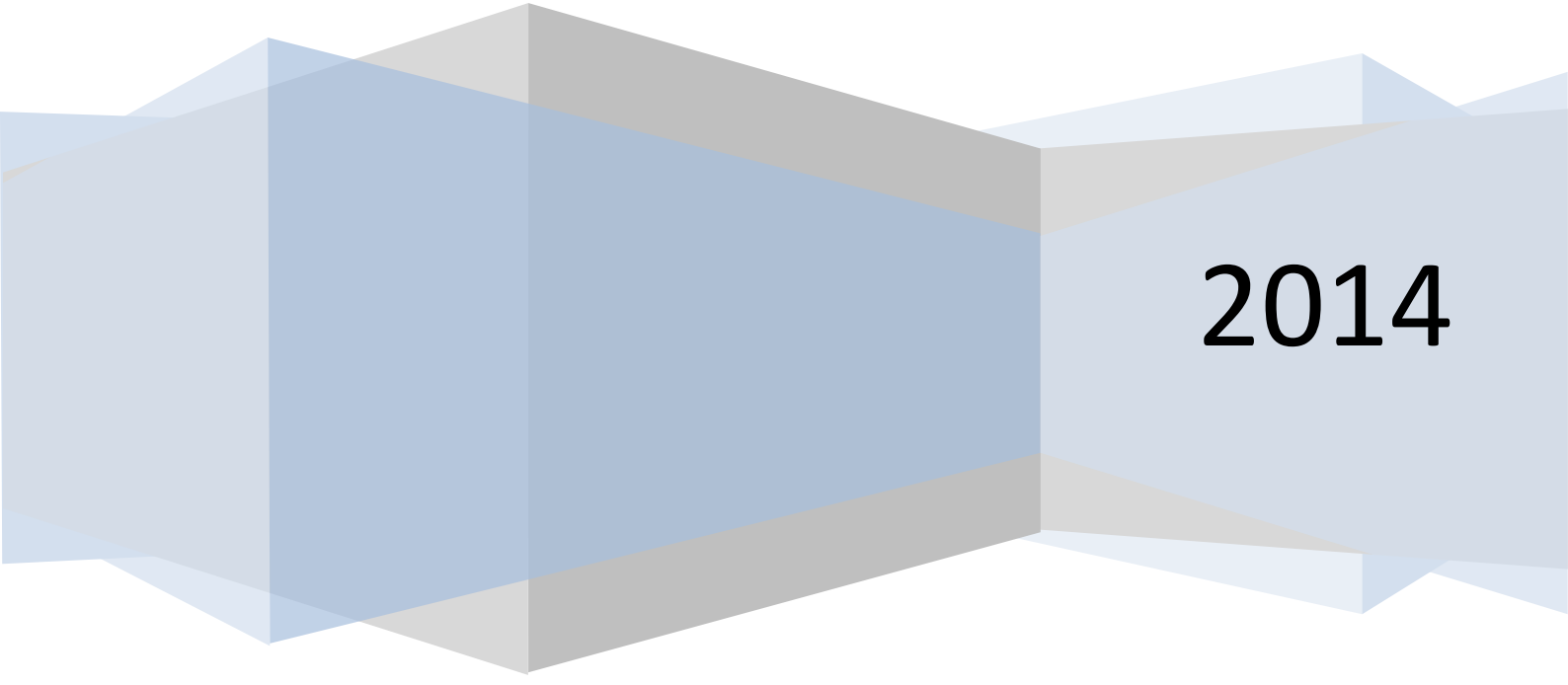
Université Paris Sud Ecole Supérieur d'Electricité

RAPPORT DE PROJET

Prévision de la consommation électrique

Luu Duy Tung – Alaeddine Chibani



2014

Table des matières

I. Introduction

1. Contexte du projet

Ce projet analyse la consommation électrique mesurée sur une zone du réseau américain. D'autre part, on observe la température mesurée par 11 stations météorologiques répartissant autour du territoire des Etas Unis. Les données ne sont pas localisées. En effet, nous nous ne connaissons pas un lien direct entre les stations de météo et les zones de consommation. En pratique, plusieurs stations de météo peuvent influencer la zone étudiée.

Les observations du projet sont effectuées pendant 4 ans (de 01 :00 :00 01/01/2004 à 00 :00 :00 01/01/2008) avec un intervalle de temps qui vaut 1 heure entre 2 observations consécutives.

2. Démarche du projet

L'objectif principal du projet est de prévoir la consommation électrique, en se basant sur trois types de prédictions : la prédiction à très court terme (prévision à une heure), la prédiction à court terme (prévision à une journée) et la prédiction à moyen terme (prévision à une semaine). Ainsi l'évaluation de la qualité de prévision figure parmi les objectifs du projet.

La réalisation du projet passe par 3 étapes fondamentales :

- **Statistique exploratoire :** Réalisation de différentes analyses exploratoires pour mieux appréhender le problème et essayer de trouver quelques idées pour construire des modèles.
- **Estimation :** Il s'agit de l'étape la plus importante du déroulement du projet, au sein de laquelle on construira des modèles pour estimer des données, puis on évaluera la qualité d'estimation, et ensuite on interprétera graphiquement ces modèles. Dans ce contexte, 2 types de modèles seront utilisés : La forêt aléatoire et le modèle additif généralisé (GAM).
- **Prédiction :** Il s'agit du but principal du projet. On appliquera les modèles déjà construits à la prévision de la consommation électrique, on évaluera la qualité de la prédiction et finalement on interprétera graphiquement les résultats.

3. Présentation des données

On dispose de $n = 30048$ observations. Chaque observation inclut une variable de réponse qui représente la consommation électrique ainsi autres covariables décrites comme suit :

- Year : variable quantitative qui représente l'année de l'observation.
- Monthf : variable qualitative qui représente le mois de l'observation.

- Dayf : variable qualitative qui représente le jour de l'observation.
- Hourf : variable qualitative qui représente l'heure de l'observation.
- Time : variable quantitative qui représente l'indice de l'observation.
- Toy : variable quantitative de 0 (début d'année) à 1 (fin d'année) qui représente l'indice de l'observation dans l'année .
- Dow : variable qualitative qui représente le jour de la semaine.
- Daytype : variable qualitative qui représente aussi le jour de la semaine avec intégration des jours fériés.
- Station1…Station11 : variable quantitative qui représente la température mesurée par les stations météorologiques (**en °F**).

Par ailleurs, on a une autre série de 3672 observations faites ultérieurement qui serviront pour évaluer la qualité de prédiction.

II. Statistique exploratoire

1. Tendance et saisonnalité

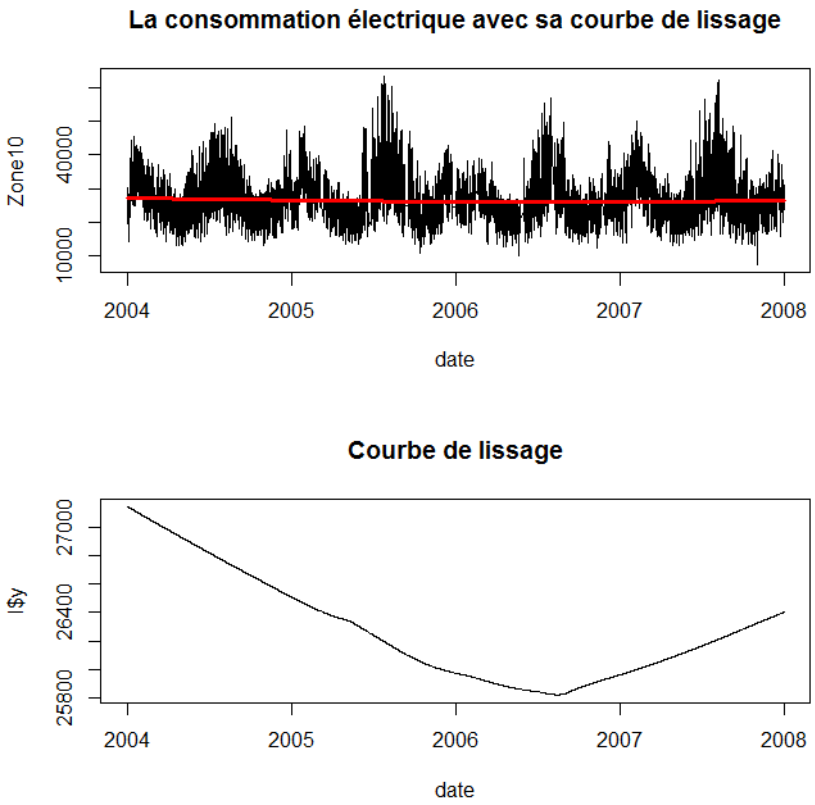


Figure 2.2.1 : Tendance et saisonnalité

- Après avoir représenté la consommation électrique dans la zone 10 pendant une durée de 4 ans qui s'étale entre 2004 et 2008. On remarque une courbe quasi-sinusoïdale. La consommation atteint une valeur maximale chaque été dans chaque année. Cela peut être

expliqué par l'utilisation de la climatisation dans les ménages et dans les entreprises. Ce qui nous permettra de supposer que la zone 10 connait des températures élevées chaque été. Ainsi, cela peut être expliqué par l'augmentation des activités de divertissement dans cette saison.

- Après avoir représenté la courbe de lissage durant l'intervalle qui s'étale de 2004 à 2007 on remarque que la courbe diminue progressivement jusqu'à atteindre un minimum au milieu de l'année 2006 puis commence à croitre jusqu'à la fin de l'intervalle. Cela peut être interprété une très faible tendance.

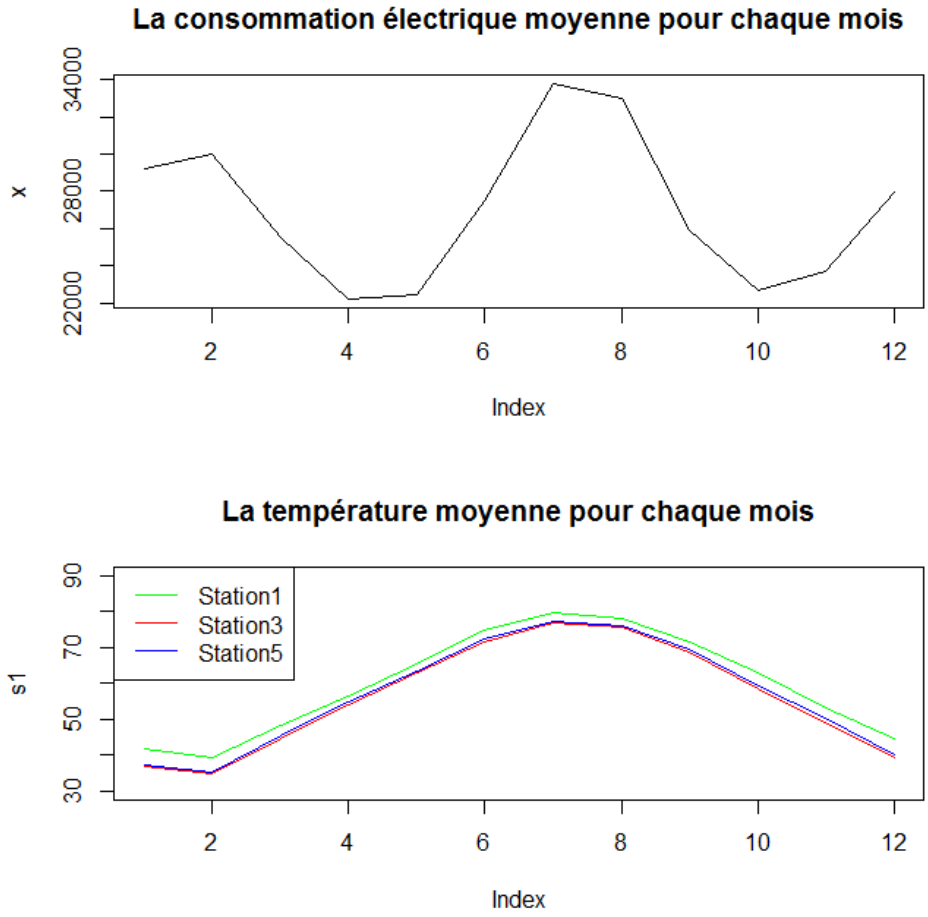


Figure 2.2.2 : Saison annuelle

- Après avoir analysé l'importance des covariables pour la prédiction de la consommation électrique, on constate que les influences des températures mesurées par les stations 1, 3 et 5 sont les plus fortes (l'analyse est reportée à la partie III.2.3, Etape 1). Alors, on comparera la consommation électrique avec les températures mesurées par ces 3 stations.
- Après avoir représenté la température durant toute l'année on constate que le pique de la consommation correspond bien à la pique de température. La consommation électrique moyenne est forte quand la température est très forte (par l'utilisation de

climatiseur) ou très faible (par l'utilisation de chauffage). Par ailleurs, on a une consommation minimale vers le mois 5 et 10 ce qui peut être expliqué par l'absence de l'utilisation du climatiseur ou chauffage.

- Après avoir comparé les températures mesurées par les stations : 1, 3 et 5. On constate que les courbes sont quasi-identiques ce qui peut être interprété par le fait que ces trois stations sont situées dans la même zone de météo.

2. Corrélation

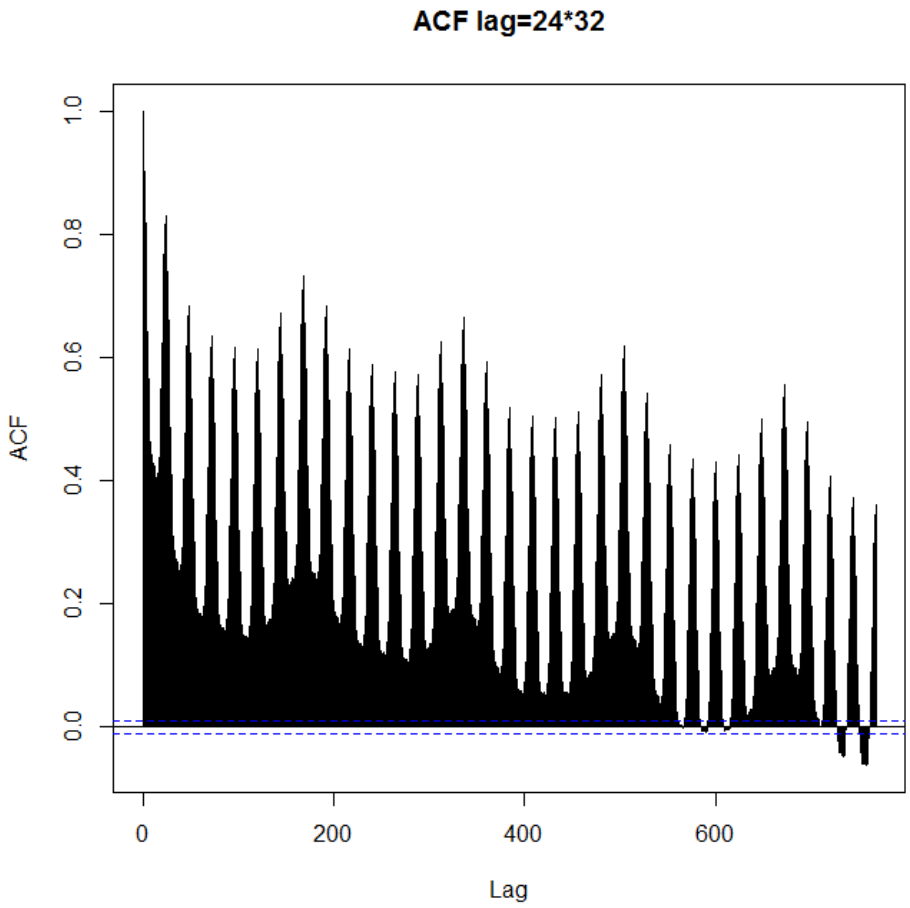


Figure 2.2.3 : Fonction d'autocorrélation empirique

- Dans cette partie de l'analyse exploratoire, on utilisera la fonction d'autocorrélation empirique. On remarque une très forte corrélation journalière. On constate aussi, une corrélation hebdomadaire. Par contre il n'y a pas de corrélation mensuelle.

3. Saisonnalité hebdomadaire

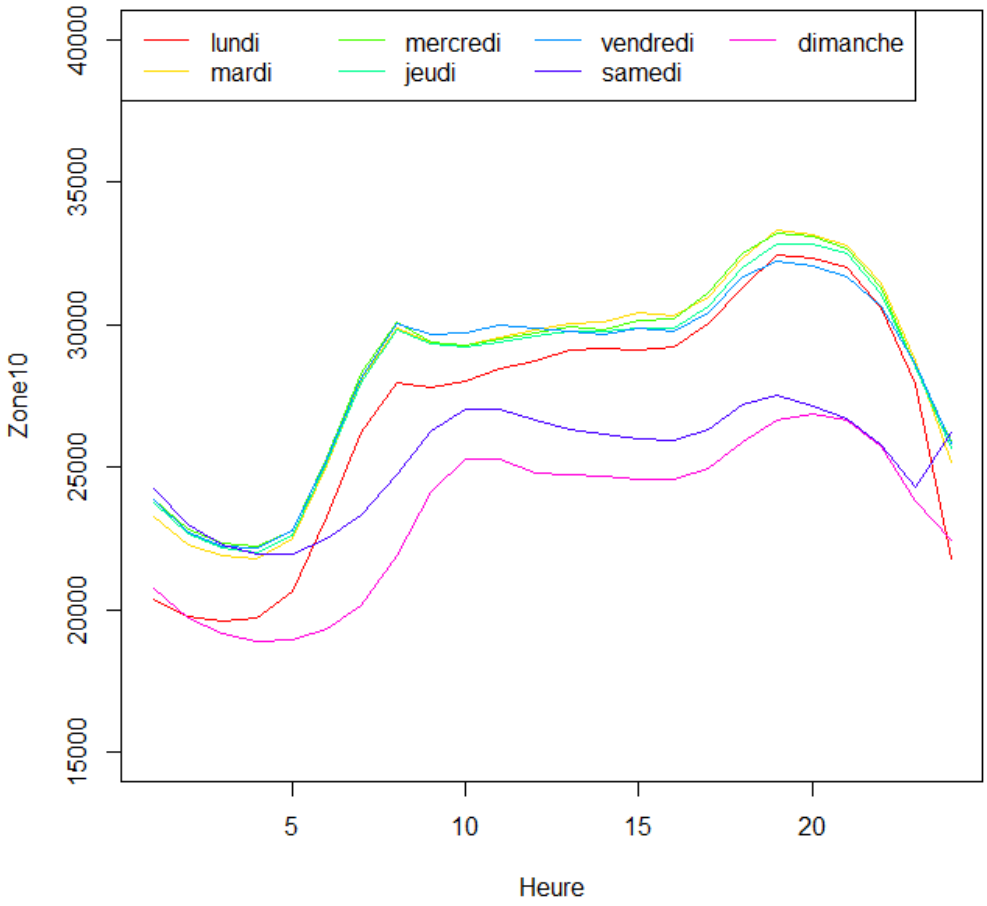


Figure 2.2.4 : La consommation électrique moyenne en fonction de l'heure pour chaque jour de la semaine

- Après avoir représenté la consommation électrique moyenne pendant l'intervalle d'une journée pour de différents types de jour, on constate que les allures de la consommation pour les jours de Mardi à Vendredi sont quasi-identiques. On remarque que la consommation Lundi et un peu moins forte que les autres jours de la semaine, ce qui peut être expliqué par le fait qu'il existe des entreprises qui ne travaillent pas ce jour.
- D'autre part, on constate que pendant le weekend on a une faible consommation par rapport aux autres jours de la semaine. Ce qui peut être expliqué par l'arrêt de quelques industriels pendant les weekends. Pourtant, on remarque une augmentation de consommation électrique Samedi soir, et cela est dû à l'augmentation des moyens de divertissement dans cette période de la semaine.

III. Estimation

1. Forêt aléatoire

1.1 Notion

Arbre de décision

Arbre de décision est une méthode de machine learning introduite par Leo Breiman en 1984. Le principe de cette méthode est de l'utilisation des covariables pour subdiviser des données en groupes homogènes de façon récursive et dyadique.

Soit $X_1 \dots X_p$ les covariables et Y variable de réponse. Pour construire un arbre de décision, on choisit une variable explicative X_i et un seuil de coupe, puis on sépare les données en deux sous ensembles. A chaque sous ensemble, on recommence le même processus jusqu'à un critère d'arrêt. Un nœud terminal (une feuille) est associé à une valeur prédite de Y qui est la moyenne des valeurs y_i (dans le cas de régression) appartient à cette feuille. On choisit la covariable et le seuil de coupe tel que les deux sous ensembles sont les plus homogènes possibles pour la variable de réponse (c-a-d minimiser la somme de variance de ces deux sous ensembles).

Pour prédire Y en fonction des covariables, il suffit de suivre des règles de décisions des covariables de racine jusqu'à la feuille et prendre la valeur prédite associée à cette feuille.

Forêt aléatoire

La forêt aléatoire est construite à partir plusieurs arbres de décision. Pour chaque arbre $a \in Forêt$, on tire une échantillon aléatoire de données (avec remise) pour construire l'arbre. A chaque nœud, on tire uniformément q variables parmi p variables totales pour former la décision associée à cette nœud.

Pour prédire Y en fonction des covariables ; il suffit de prédire Y par chaque arbre et moyenner les valeurs prédites.

Critère variance-explained

Ce critère sert à évaluer la qualité d'une forêt F quelconque, il est calculé par :

$$PVE(F) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}.$$

Où y_i des vraies valeurs de la variable de réponse, $\hat{y}_i$ valeurs prédites par la forêt F et $\bar{y}$ la moyenne des y_i. Comme $\bar{y}$ est aussi l'estimateur du modèle identité, alors ce critère est similaire au critère R2, alors ce critère a un autre nom : *pseudo Rsquare*.

Comme dans la forêt aléatoire, on peut éviter le surapprentissage en construisant un grand nombre d'arbre. Donc, on ne doit pas utiliser des critères avec la pénalisation.

1.2 Construction d'une forêt aléatoire

D'abord, on construit une forêt aléatoire de 250 arbres de la consommation électrique en fonction de toutes les covariables. Chaque forêt est construite avec $\text{PartieEntier}(0.632 * n) = 18990$ individus avec $PartieEntier\left(\frac{p}{3}\right) = 6$ variables tirées pour la construction de chaque nœud.

model <- randomForest(Zone10 ~ Station1 + Station2 + Station3 + Station4 + Station5 + Station6+Station7+Station8+Station9+Station10+Station11+Year+Monthf+Dayf+Hourf+Toy+dow+daytype+Time, data=donnees, na.action=na.omit,ntree=250)

On obtient une forêt avec variance explained égale à 0.9442. Puis, on analyse l'autocorrélation empirique de la série des résidus.

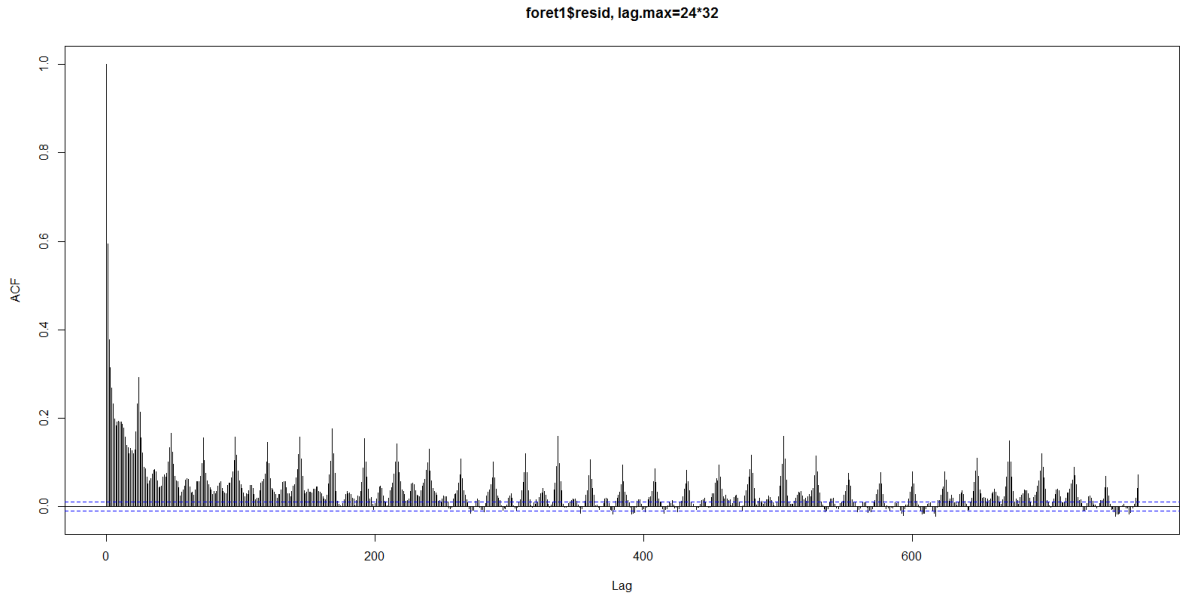


Figure 1.2.1 : acf de de la forêt aléatoire (consommation ~ toutes les covariables)

D'après la figure 1.2.1, l'acf présente une forte corrélation entre les observations voisines et des piques journalières. Alors la consommation électrique au moment t dépend celle des moments précédents $t-1$, $t-2$, $t-3$ …. $t-24$, $t-48$ … Donc, on propose une nouvelle forêt :

model <- randomForest(Zone10 ~ Station1 + Station2 + Station3 + Station4 + Station5 + Station6+Station7+Station8+Station9+Station10+Station11+Year+Monthf+Dayf+Hourf+Toy+dow+daytype+Time+lag1+lag2+lag3+lag4+lag5+lag24+lag48+lag72+lag96+lag120+lag144+lag168, data=donneesr, na.action=na.omit,ntree=250)

Dans ce modèle, lagi correspond à Y_{t-i} .

On obtient une forêt avec la variance explained égale à 0.9801, puis on réutilise acf.

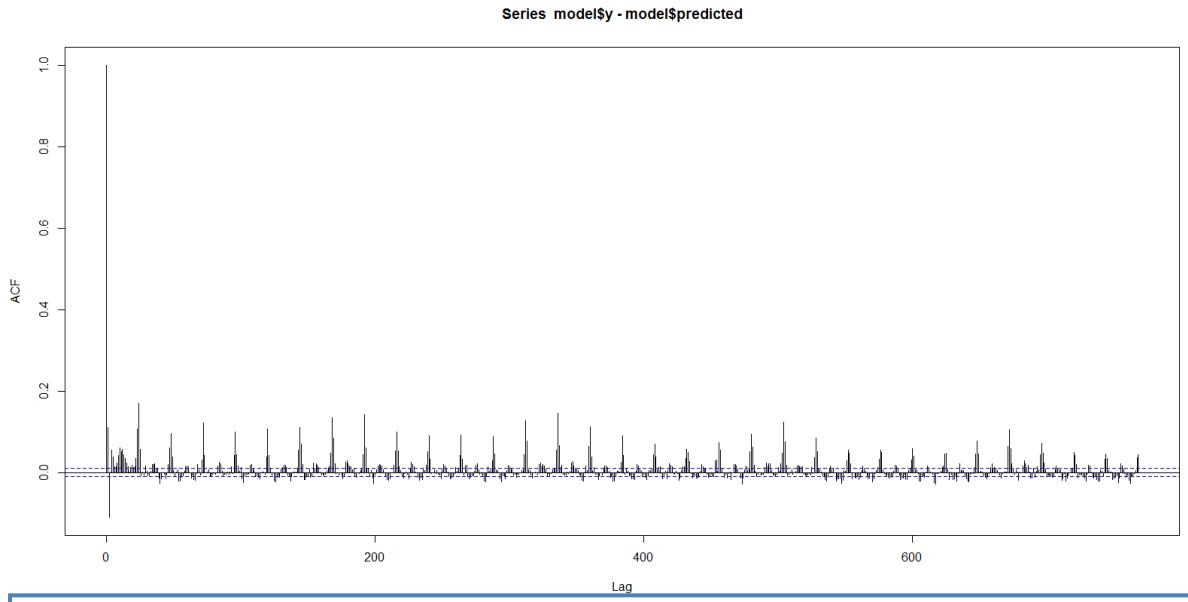


Figure 1.2.2 : acf de de la forêt aléatoire (consommation ~ toutes les covariables + terme autorégressif)

On constate que l'effet d'autocorrélation a diminué. Mais l'effet d'autocorrélation journalière n'est pas bien ajusté.

2. Modèle additif généralisé

2.1 Notion

(Référence. Article GEFCom2012: Electric load forecasting and backcasting with semi-parametric models, Raphael Nedellec, Jairo Cugliari, Yannig Goude)

Supposons qu'on a n observations. Soit y_i des valeurs de variable réponse. On souhaite exprimer y_i en fonction non linéaire des covariables $x_i = (x_{1,i} \dots x_{p,i})$:

$$y_i = f_1(x_{1,i}) + \cdots + f_p(x_{p,i}) + \epsilon_i, \qquad \epsilon_i \text{ est le bruit .}$$

Pour $\boldsymbol{f_q(x)} = \sum_{j=1}^{k_q} \boldsymbol{\beta_{q,j} b_{q,j}(x)}$. Les $\boldsymbol{b_{q,j}(x)}$ sont les fonctions de base spline, $\boldsymbol{k_q}$ est la dimension de base spline.

Soit B matrice de concaténation de tous les vecteurs de valeurs des fonctions des bases splines, le problème devient :

$$Y = B\beta + \epsilon \, .$$

Pour Y vecteur des y_i, β vecteur des $\beta_{q,j}$, ϵ vecteur des ϵ_i.

On résout souvent ce problème par la méthode de Ridge.

$$\hat{\beta} = argmin_{\beta} \left(||Y - B\beta||^2 \right) + \sum_{i=1}^{p} \lambda_i \int ||f_i''(x)||^2 dx \,.$$

$\lambda = (\lambda_1 \dots \lambda_p)$ estimé par VC, AIC, BIC, Cp, GCV…

2.2 Construction d'un modèle additif généralisé

La construction d'un modèle additif généralisé ajusté des données passe par plusieurs étapes en rajoutant une ou plusieurs covariables. Chaque étape se compose de 3 parties :

- Choix de covariable,
- Analyse de covariable,
- Evaluation du modèle

Choix de covariable

A la première étape, on cherche la variable la plus importante pour la prédiction de la consommation électrique et on construit le modèle initial g0 par cette covariable. Comme l'effet de cette variable est ajusté par le modèle g0, elle ne sera plus importante pour g0$residuals. Ce qui nous amène dans l'étape suivante à chercher la variable la plus importante pour g0$residuals et construire g1. Ainsi, de la même façon, on choisit la variable pour l'étape k+1 à partir de gk$residuals.

Pour évaluer l'importance des covariables à l'étape k+1, on construit une forêt aléatoire en considérant gk$residuals comme variable explicative et utilise le critère «gini-importance» de ce modèle. Ce critère est expliqué ci-dessous.

Soit T un arbre dans la forêt aléatoire, on calcule d'abord $L(v,T)$ la valeur de Gini-importance (cas de régression) de covariable v associé à l'arbre T.

$$L(v,T) = \sum_{\omega \in N(T,v)} \Delta Loss(\omega) = \sum_{\omega \in N(T,v)} \left(\mathrm{SCR}(\omega) - \mathrm{SCR}(\omega_g) - \mathrm{SCR}(\omega_d) \right) .$$

Où :

$N(T,v)$: l'ensemble des nœuds de l'arbre T répartis par la variable v.

ω_g, ω_d : le sous fils gauche et le sous fils droite de ω respectivement.

Pour N un nœud quelconque, $SCR(N) = \sum_{i \in N} \left(Y_i - \frac{1}{\#N} \sum_{i \in N} Y_i \right)^2$.

Enfin, la valeur de Gini-importance de la forêt aléatoire calculée par :

$$L(v) = \frac{1}{\#Forêt} \sum_{T \in Forêt} L(v,T).$$

Analyse de covariable

Après avoir choisi une covariable, on peut exprimer cette variable dans le modèle GAM par plusieurs façons : une régression linéaire simple, un ANOVA, une base spline simple (thin plate splines, cyclic splines) ou une base spline avec l'intéraction... Les choix sont proposés par l'analyse des données et par l'évaluation de la qualité du modèle et on n'utilise pas une condition mathématique précise pour le choix. Ainsi donc, il existe plusieurs choix de modèle.

Après avoir rajouté cette variable au modèle, on compare ses résidus avec ceux du modèle précédent pour retenir l'effet de cet ajustement.

Evaluation

L'évaluation du modèle est effectuée par le critère R2-ajusté décrit ci-dessous.

Soit Y vecteur de variable de réponse. (ω) modèle d'identité dont $\hat{Y}_\omega$ un estimateur de Y, (Ω) modèle à évaluer dont $\hat{Y}_\Omega$ un estimateur de Y.

On rappelle que :

$$R^2 = \frac{\left|\left|\hat{Y}_\Omega - \hat{Y}_\omega\right|\right|^2}{\left|\left|Y - \hat{Y}_\omega\right|\right|^2} = 1 - \frac{\left|\left|Y - \hat{Y}_\Omega\right|\right|^2}{\left|\left|Y - \hat{Y}_\omega\right|\right|^2}.$$

Pour obtenir R2-ajusté, on rajoute la pénalisation.

$$R^2_{aj} = 1 - \frac{n}{n-p}\frac{\left|\left|Y - \hat{Y}_\Omega\right|\right|^2}{\left|\left|Y - \hat{Y}_\omega\right|\right|^2}.$$

Dans cette formule, n est le nombre d'observations, p est la dimension du modèle. Si la dimension du modèle est trop grande par rapport au nombre d'observations, R^2_{aj} est pénalisé et diminué. Cela évitera le surapprentissage. Donc, plus ce critère s'approche à 1, plus ce modèle a une bonne qualité. Idéalement, on cherchera un modèle tel que $R^2_{aj} > 0.95$.

2.3 Démarche de la construction du modèle

En appliquant la méthode ci-dessus, on passe par les 5 étapes suivantes.

Etape 1

Sélection de variable

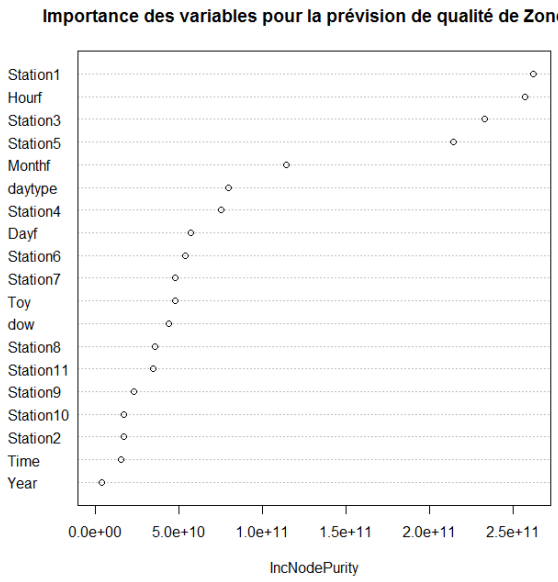


Figure 3.2.1 : L'importance des variables pour la prévision de la consommation électrique

D'après la figure 3.2.1, on note que la covariable Station1 (la température mesurée par la Station1) est la plus importante, alors on choisit cette variable pour rajouter au modèle.

Par ailleurs, on observe que : Parmi les stations, les influences des stations : 1, 3, 5 sont les plus fortes.

Analyse de variable

On interprète d'abord la consommation électrique en fonction de température pour tous les mois.

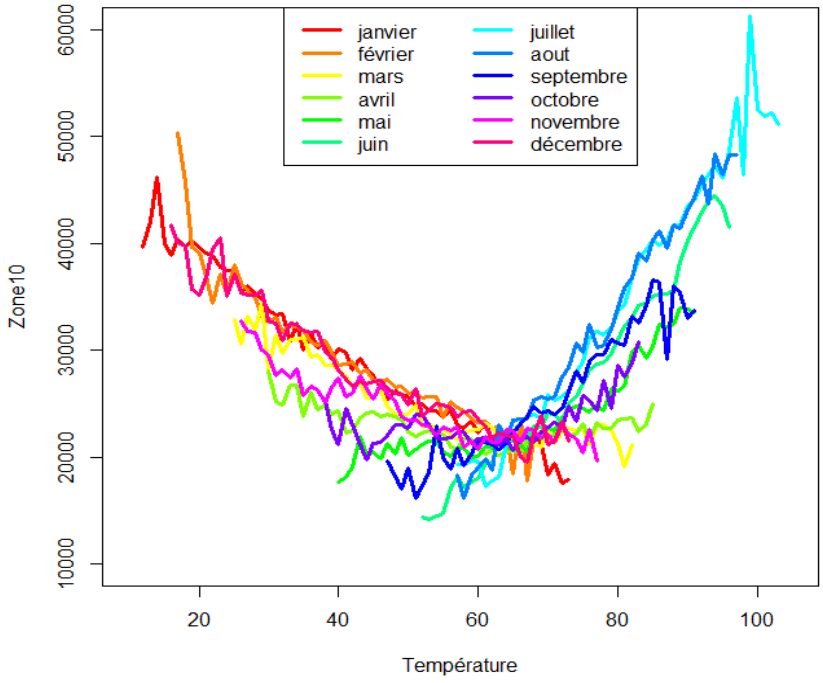


Figure 3.2.2 Graphe de la consommation électrique en fonction de température pour tous les mois

D'après la figure 3.2.2, on remarque que les 12 séries ne sont pas parallèles. L'allure des séries au mois « chaud » est très différente de celle des séries au mois « froid ». Alors, il y a une interaction significative entre la température et le mois. En d'autre terme, l'effet de température-mois peut être caractérisé par un modèle suivant :

$$E_i^{Zone10,tm} = a_{Monthf_i} + f_{Monthf_i}(t_i) + \epsilon_i^{Zone10,tm} \ .$$

i est l'indice de l'observation, t_i la température à l'instant i, $Monthf_i$ le mois à l'instant i, $\epsilon_i^{Zone10,tm}$ le bruit, f_{Monthf_i} les fonctions non linéaires.

Donc, pour ajuster l'effet de température, on rajoute au modèle $g0$ une base de thin-plat-spline de température avec l'interaction (12 bases thin-plat-spline simples) du mois :

$$S_i^{tm} = \sum_{j=1}^{K_{temp}^{Monthf_i}} \beta_{j,temp}^{Monthf_i} \phi_{j,temp}^{Monthf_i}(t_i) \ .$$

$K_{temp}^{Monthf_i}$ dimensions des bases splines (à estimer). $\beta_{j,temp}^{Monthf_i}$ coefficients (à estimer), $\phi_{j,temp}^{Monthf_i}$ fonctions de thin-plat-spline.

Dans R : *g0<- gam(Zone10~s(Station1,by=Monthf),data=donnees)*

On tracera le même graphe que 3.2.2 mais pour g0$residuals.

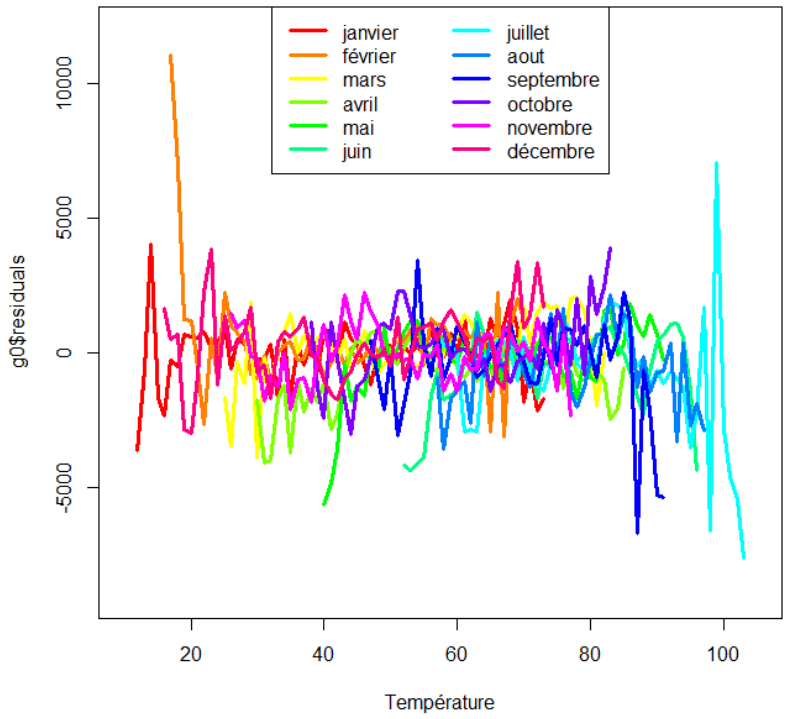


Figure 3.2.3 Graphe de g0$residuals en fonction de température pour tous les mois

En observant la graphe 3.2.3, on remarque que les courbes sont à peu près parallèles et de la forme $y = const$, alors l'effet de la température a été retiré, il ne reste que l'effet du mois :

$$E_i^{g0\$resid,tm} = a_{Monthf_i} + \epsilon_i^{g0\$resid,tm} .$$

Comme ces courbes sont centrées à 0, alors l'effet du mois n'est pas significatif. Cette remarque sera vérifiée dans la partie « Sélection de variable » de l'étape suivante.

Evaluation

R2-ajusté précédent : 0.0

R2-ajusté de g0 : 0.653

Etape 2

Sélection de variable

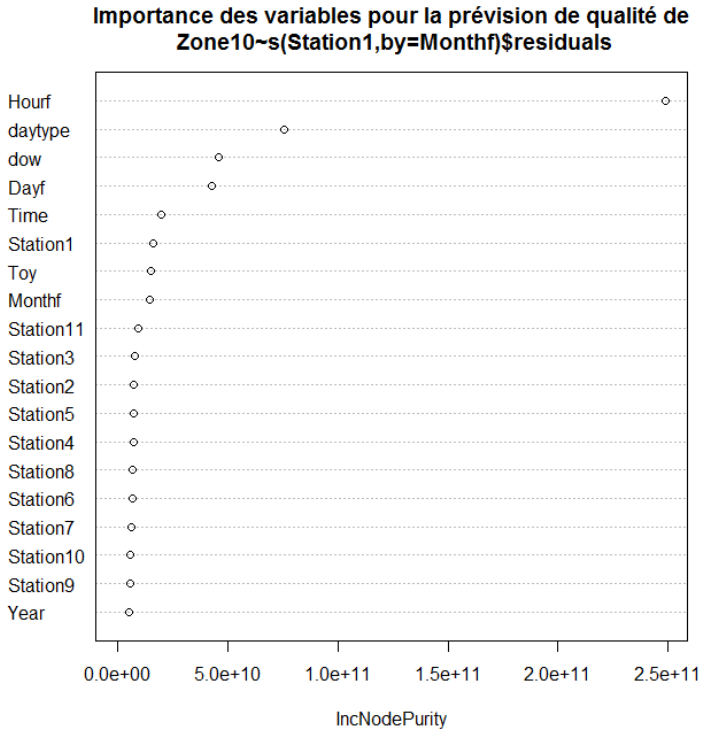


Figure 3.2.4 : L'importance des variables pour la prévision de g0$resid

D'après la figure 3.2.4, on choisit la variable Hourf pour cette étape. Par ailleurs, l'importance de Monthf est faible, cela vérifie la remarque sur l'effet de Monthf dans l'étape 1.

Analyse de variable

On trace d'abord g0$resid en fonction de l'heure pour tous les jours de la semaine.

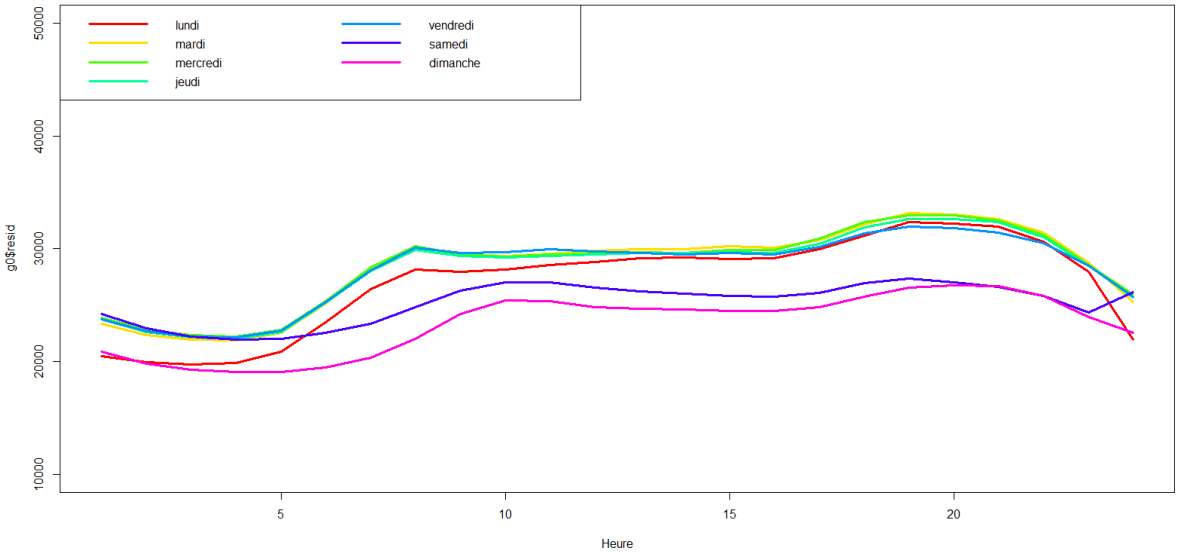


Figure 3.2.5 : g0$resid en fonction de l'heure pour tous les jours de la semaine

En observant la figure 3.2.5, on note que ces 7 séries ne sont pas parallèles. On trouve la différence deux à deux entre l'allure des séries de lundi, mardi-mercredi-jeudi, samedi et dimanche. Alors, il y a une interaction significative entre l'heure et le jour de la semaine. En considérant l'heure comme variable quantitative, l'effet heure-dow peut s'écrire comme suit:

$$E_i^{g0\$resid,hdow} = a_{dow_i} + f_{dow_i}(h_i) + \epsilon_i^{g0\$resid,hdow} \quad .$$

i est l'indice de l'observation, h_i l'heure à l'instant , dow_i le jour de la semaine à l'instant i, $\epsilon_i^{Zone10,hdow}$ le bruit, f_{dow_i} les fonctions non linéaires.

Donc, pour ajuster l'effet de l'heure, on rajoute au modèle $g1$ une base de thin-plat-spline de l'heure avec l'interaction (7 bases thin-plat-spline simples) du jour de la semaine :

$$S_i^{hdow} = \sum_{j=1}^{K_{hour}^{dow_i}} \beta_{j,hour}^{dow_i} \phi_{j,hour}^{dow_i}(h_i) \, .$$

$K_{hour}^{dow_i}$ dimensions des bases splines (à estimer). $\beta_{j,hour}^{dow_i}$ coefficients (à estimer), $\phi_{j,hour}^{dow_i}$ fonctions de thin-plate-spline. Par ailleurs on doit convertir la variable d'heure à la forme quantitative pour mettre à la base spline.

***Dans R**: g1<- gam(Zone10~s(Hour,by=dow)+s(Station1,by=Monthf),data=donnees)*

On retracera le graphe 3.2.5 mais pour g1$residuals.

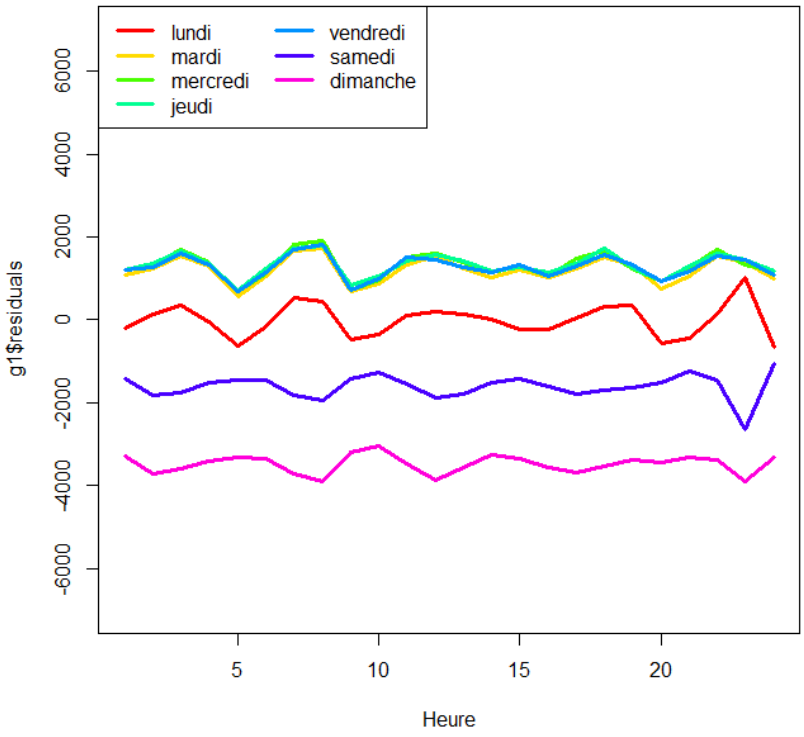


Figure 3.2.6 : g1$resid en fonction de l'heure pour tous les jours de la semaine

D'après la figure 3.2.6, les séries sont à peu près parallèles et de la forme $y = const$, on note que l'effet de l'heure avec son interaction avec les jours de la semaine a été retiré. Il ne reste que l'effet des jours de la semaine :

$$E_i^{g1\$resid,hdow} = a_{dow_i} + \epsilon_i^{g1\$resid,hdow} \quad .$$

Mise à part la série de lundi, toutes les autres séries ne sont pas centrées à 0. Donc, on constate que l'effet de dow est fort pour g1$resid. On va vérifier cette remarque par l'importance des variables dans l'étape 3.

Evaluation

R2-ajusté précédent : 0.653

R2-ajusté de g1 : 0.823

Etape 3

Sélection de variable

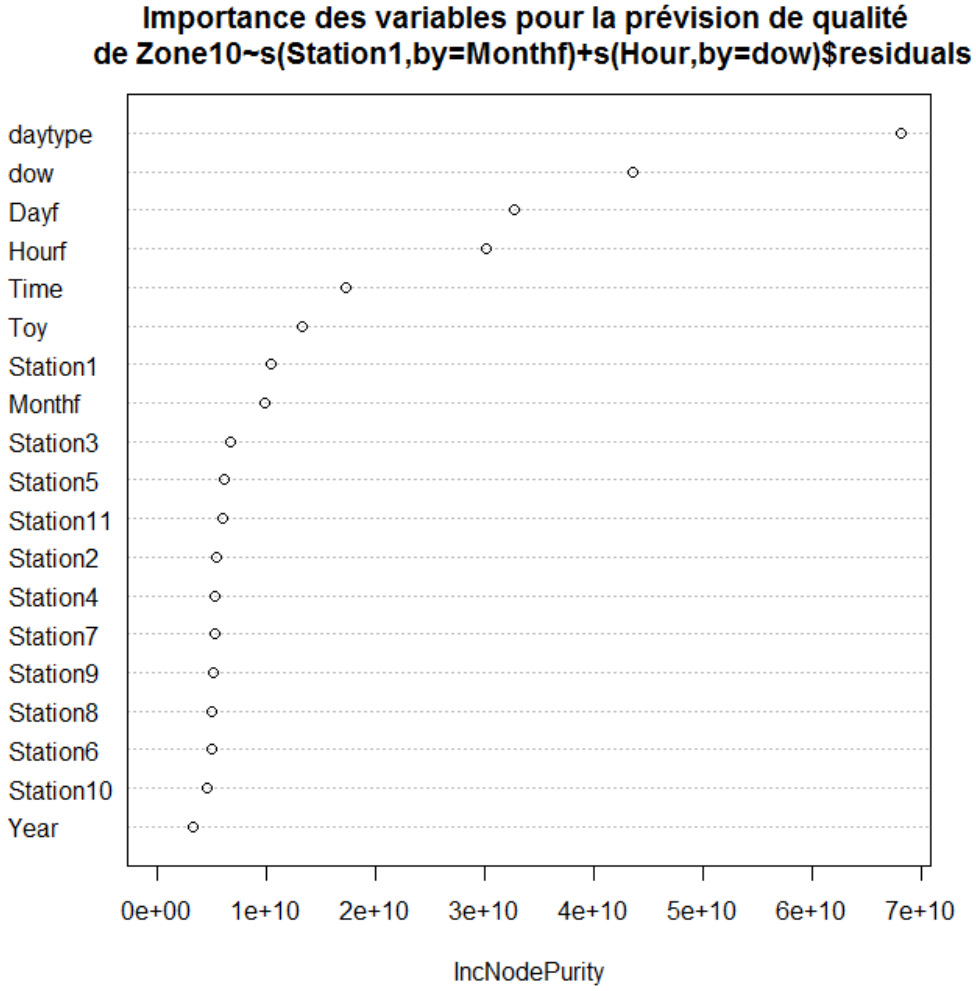


Figure 3.2.7: L'importance des variables pour la prévision de g1$resid

D'après la figure 3.2.7, on trouve que la variable dow est très importante pour la prévision de g1$resid, cela vérifie la remarque sur l'effet de dow dans l'étape 2.

Pourtant, elle est moins importante que la variable daytype. Parce que, les valeurs de variable dow sont incluses dans la variable daytype. On rappelle que la variable daytype se compose : des jours fériés et des jours de la semaine.

Analyse de variable

Il suffit qu'on ajuste l'effet de daytype par un ANOVA(1)

$$A_i^{daytype} = a_{daytype_i}.$$

i est l'indice de l'observation, $daytype_i$ le type du jour à l'instant i. Puis on rajoute ce terme au modèle gam pour obtenir un nouveau modèle g2.

***Dans R**: g2<- gam(Zone10~s(Hour,by=dow)+s(Station1,by=Monthf)+daytype,data=donnees)*

Retournons à l'étape 2, on trace maintenant g2$resid en fonction de l'heure pour tous les jours de la semaine.

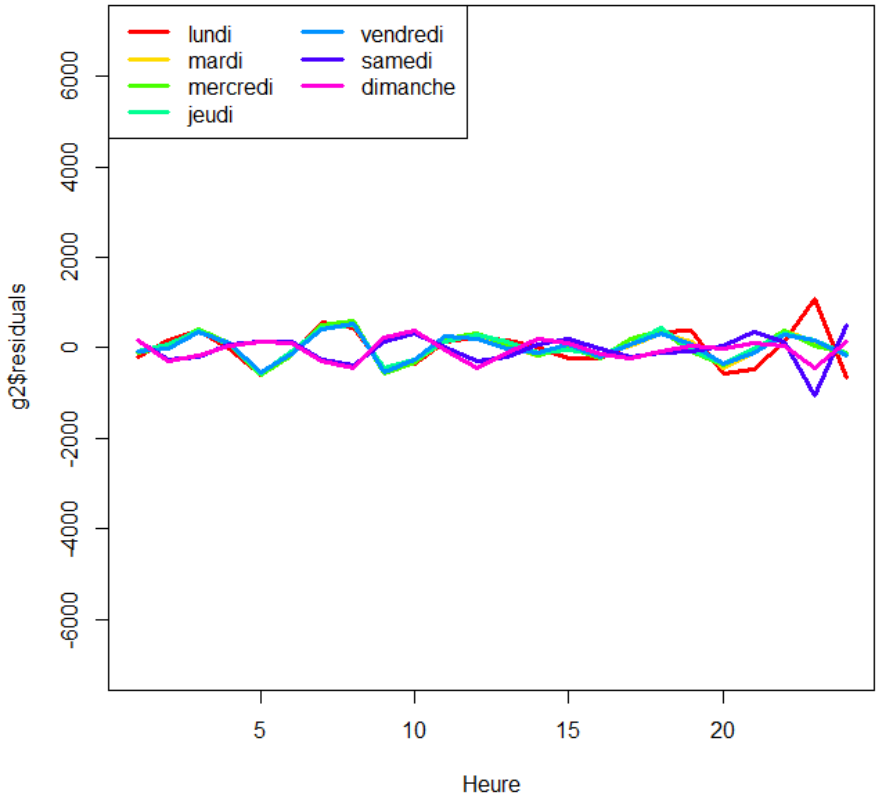


Figure 3.2.8: g2$resid en fonction de l'heure pour tous les jours de la semaine

Comme les valeurs de dow sont incluses dans daytype. Alors quand on ajuste l'effet de daytype, on ajuste aussi l'effet de dow. Comme il n'y a plus l'effet de dow dans g2$residuals, les courbes dans la figure 3.2.7 sont toutes centrées à 0.

Evaluation

R2-ajusté précédent : 0.823

R2-ajusté de g2 : 0.891

Etape 4 :

Sélection de variable

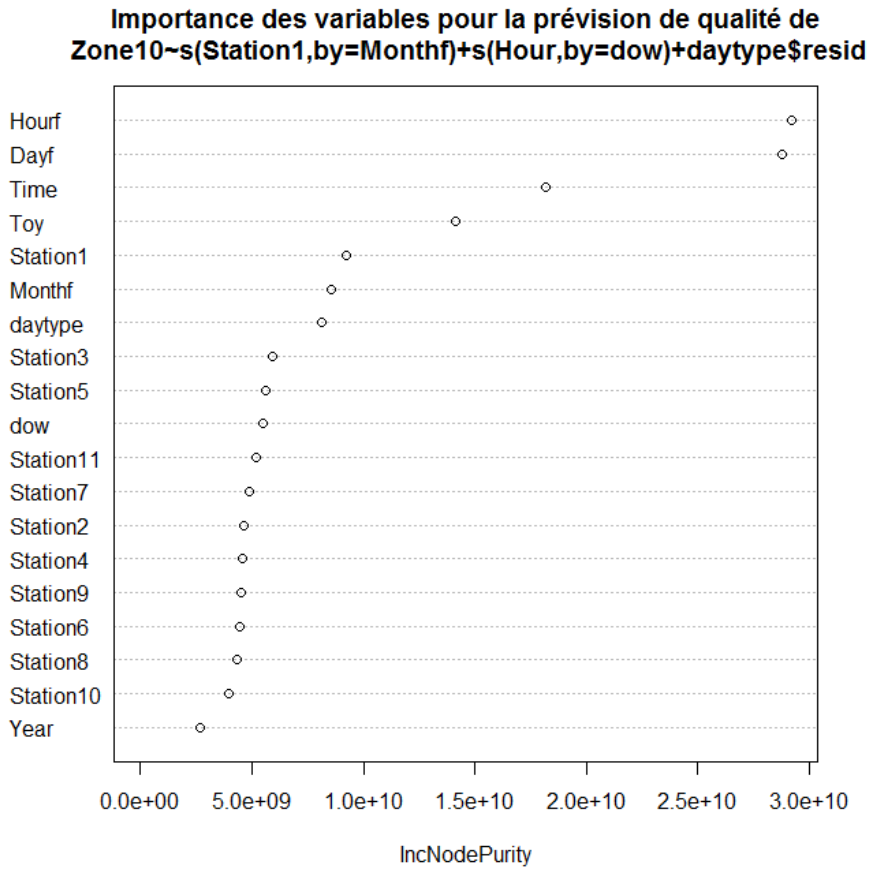


Figure 3.2.9: L'importance des variables pour la prévision de g2$resid

Dans le graphe de l'importance des variables, on retombe sur la variable Hourf qui est déjà ajustée dans l'étape 2. C'est-à-dire l'effet des covariables pour la prévision de g2$resid est déjà très faible. Alors l'ajustement de la consommation électrique par des covariables sera arrêté après cette étape.

Dans cette étape, on va choisir 4 covariables les plus fortes: Hourf, Dayf, Time, Toy

Analyse de variable

On va considérer la variable de l'heure comme une variable quantitative et la mettre dans la base thin-plat-spline avec l'interaction avec les mois. La démarche est similaire à l'étape 2.

Pour la variable Dayf, il suffit d'ajuster par ANOVA(1).

La variable Time (le temps qui coule de 1 à T=30048) sera utilisée pour estimer la tendance des données en mettant dans la base thin-plat-spline.

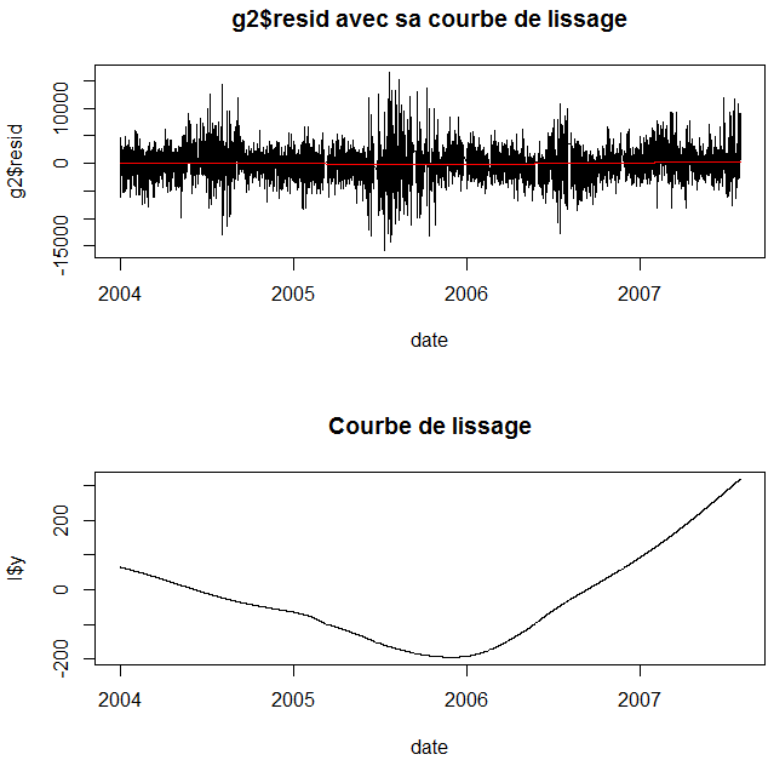


Figure 3.2.10:La tendance de g2$resid

D'après la figure 3.2.10, on trouve la tendance de g2$resid par le lissage. On note qu'elle varie très faiblement et son allure est simple. Donc, dans la base thin-plat-spline, on devrait choisir la dimension K_{time} assez faible.

La variable Toy (qui coule de 0 à 1 pendant 1 année), est utilisée pour estimer la saisonnalité annuelle. En observant la série g2$resid, on note qu'elle présente une saisonnalité annuelle d'allure sinusoïdale. Donc on l'estime en mettant Toy à la base de spline cyclique.

En résumé, ces effets sont caractérisés par :

$$E_i^{g3\$resid,dttt} = a_{Monthf_i} + f_{Monthf_i}(hour_i) + b_{Dayf_i} + g(Time_i) + h(Toy_i) + \epsilon_i^{g3\$resid,dttt} \ .$$

Qui sont ajustés par

$$S_i^{dttt} = \sum_{j=1}^{K_{hour}^{Monthf_i}} \beta_{j,hour}^{Monthf_i} \phi_{j,hour}^{Monthf_i}(hour_i) + b_{Dayf_i} + \sum_{j=1}^{K_{Time}} \beta_{j,Time} \phi_{j,Time}(Time_i) + \sum_{j=1}^{K_{Toy}} \beta_{j,Toy} \phi_{j,Toy}^{cyclique}(Toy_i) \ .$$

En rajoutant ces 4 termes au modèle g2, on obtient un nouveau modèle : g3.

Dans R: *g3 = gam(Zone10~s(Hour,by=dow)+s(Station1,by=Monthf)+daytype+ s(Hour,by=Monthf)+s(Time,k=3)+s(Toy,bs="cc")+Dayf,data=donnees)*

Evaluation

R2-ajusté précédent : 0.891

R2-ajusté de g3 : 0.918

Etape 5

Choix et analyse des covariables

Dans les 4 étapes précédentes, la consommation électrique a été ajustée en fonction des covariables. Donc, la série g4$resid ne présente que le bruit et le terme autorégressif. Cette étape vise à identifier et ajuster ce terme.

Pour identifier le terme autorégressif, on va observer la fonction acf de g3$resid.

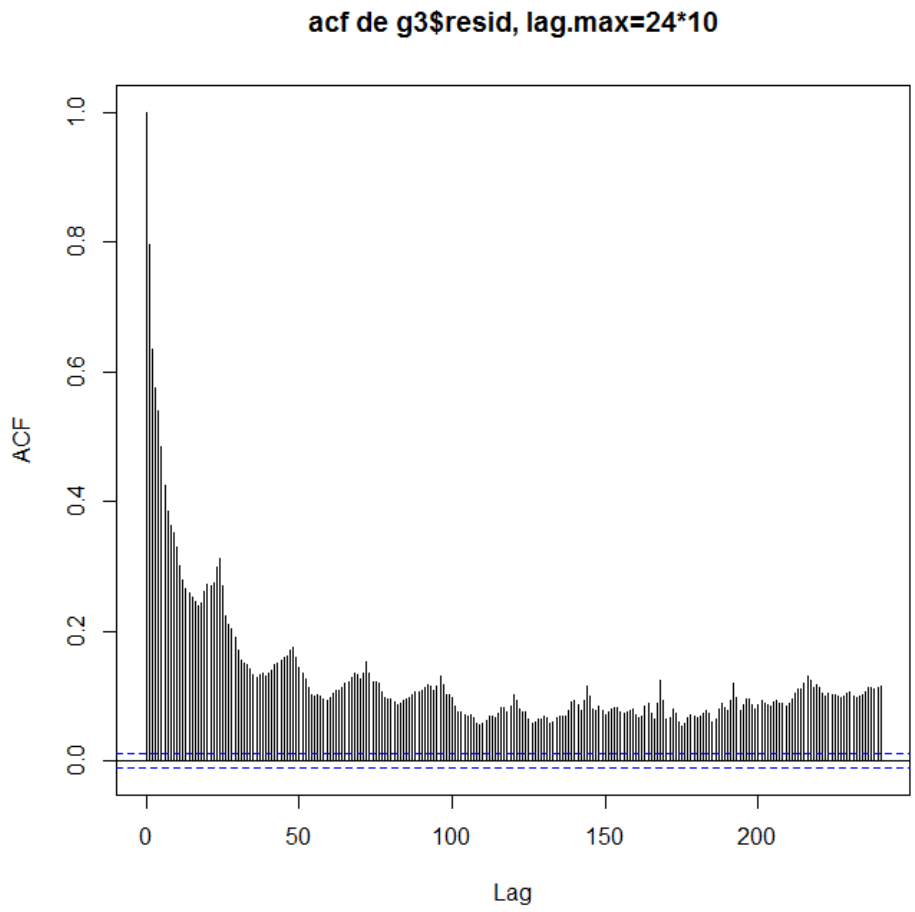


Figure 3.2.11: acf de g3$resid

D'après la figure 3.2, l'acf présente une forte corrélation entre les observations voisines et des piques journalières. Alors la consommation électrique au moment t dépend des moments précédents $t-1$, $t-2$, $t-3$ …. $t-24$, $t-48$ …

Ainsi donc, on va estimer la consommation actuelle en fonction des consommations passées en les mettant à la base thin-plat-spline. Enfin, on rajoute ces termes au modèle pour construire le modèle g4.

Dans R :

g4=gam(Zone10~s(Hour,by=dow)+s(Station1,by=Monthf)+daytype+s(Hour,by=Monthf)+Dayf+s(Time,k=3)+s(Toy,bs="cc")+s(lag1)+s(lag2)+s(lag3)+s(lag4)+s(lag5)+s(lag24)+s(lag48)+s(lag72)+s(lag96)+s(lag120)+s(lag144)+s(lag168),data=donneesr)

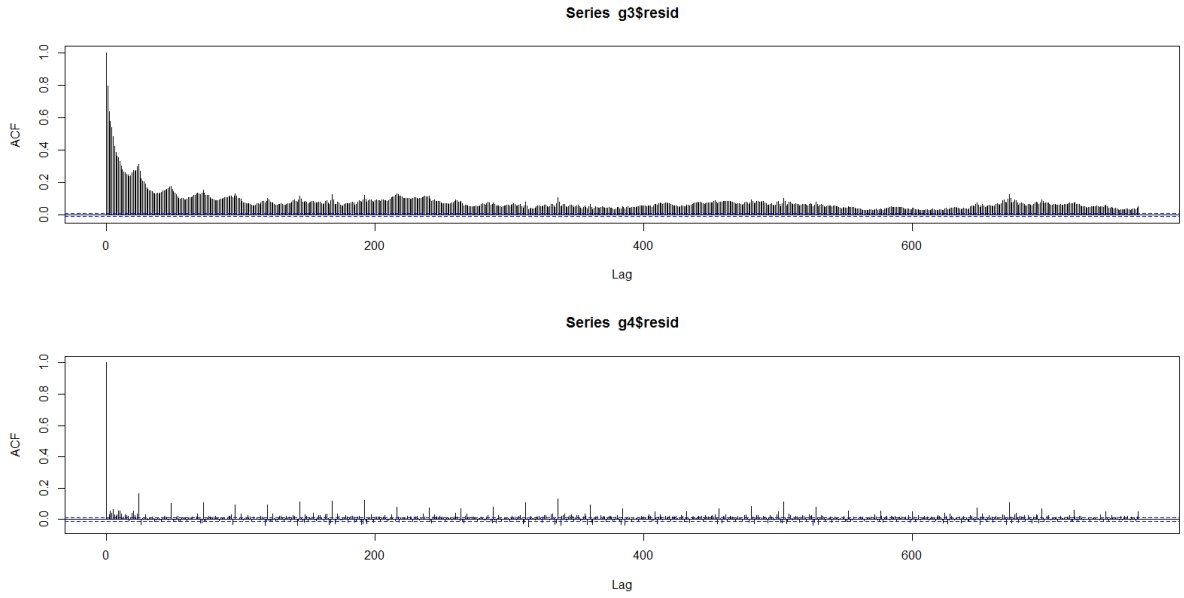


Figure 3.2.12: acf de g3$resid et g4$resid

En comparant entre l'acf de g3$resid et l'acf de g4$resid, on constate que l'effet d'autocorrélation a diminué. Mais l'effet d'autocorrélation journalière n'est pas bien ajusté.

Evaluation

R2-ajusté précédent : 0.918

R2-ajusté de g4 : 0.98

Modèle GAM de Station3 et Station5

Dans l'étape 1, on a vu que l'influence des stations 3 et 5 est très significative. Pourtant, la modalité de ces stations est similaire à celle de la station 1 (cf. figure 2.2.2), alors lorsqu'on ajuste l'effet de la station 1, l'effet des stations 3 et 5 est annulé. Cela nous pousse à penser à l'idée de la construction d'autres modèles GAM. Par la même démarche, on obtient les 2 modèles GAM suivants :

*g4s3=gam(Zone10~s(Hour,by=dow)+s(**Station3**,by=Monthf)+daytype+s(Hour,by=Monthf)+Dayf+s(Time,k=3)+s(Toy,bs="cc")+s(lag1)+s(lag2)+s(lag3)+s(lag4)+s(lag5)+s(lag24)+s(lag48)+s(lag72)+s(lag96)+s(lag120)+s(lag144)+s(lag168),data=donneesr)*

R2-ajusté de g4s3: 0.979

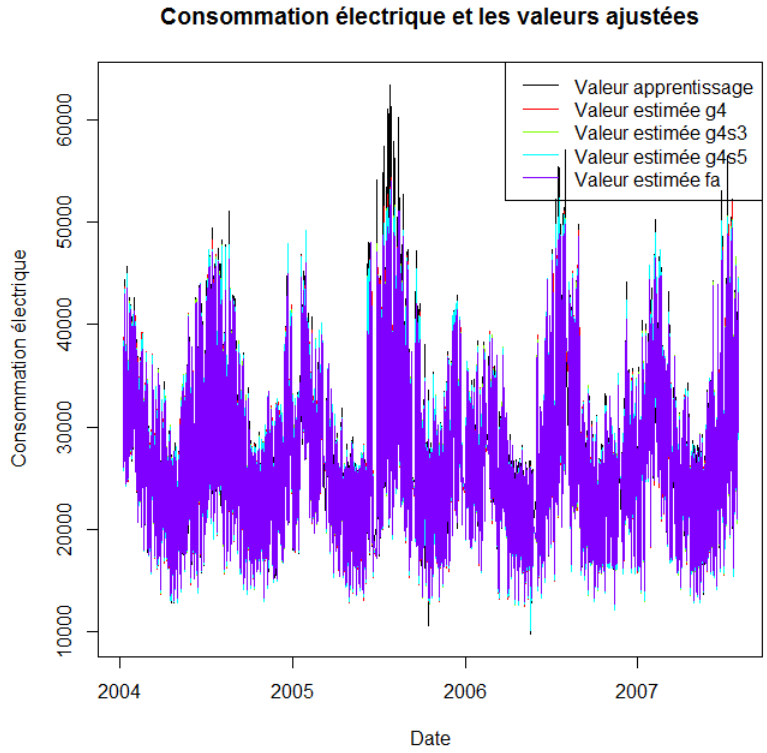
*g4s5=gam(Zone10~s(Hour,by=dow)+s(**Station5**,by=Monthf)+daytype+s(Hour,by=Monthf)+Dayf+s(Time,k=3)+s(Toy,bs="cc")+s(lag1)+s(lag2)+s(lag3)+s(lag4)+s(lag5)+s(lag24)+s(lag48)+s(lag72)+s(lag96)+s(lag120)+s(lag144)+s(lag168),data=donneesr)*

R2-ajusté de g4s5: 0.979

3. Interprétation graphique

On trace les vraies valeurs d'apprentissage en ajustant avec les valeurs estimées des 4 modèles: **g4** est le modèle GAM avec la station 1, **g4s3** est le modèle GAM avec la station 3, **g4s5** est le modèle GAM avec la station 5, et **fa** la forêt aléatoire.

.



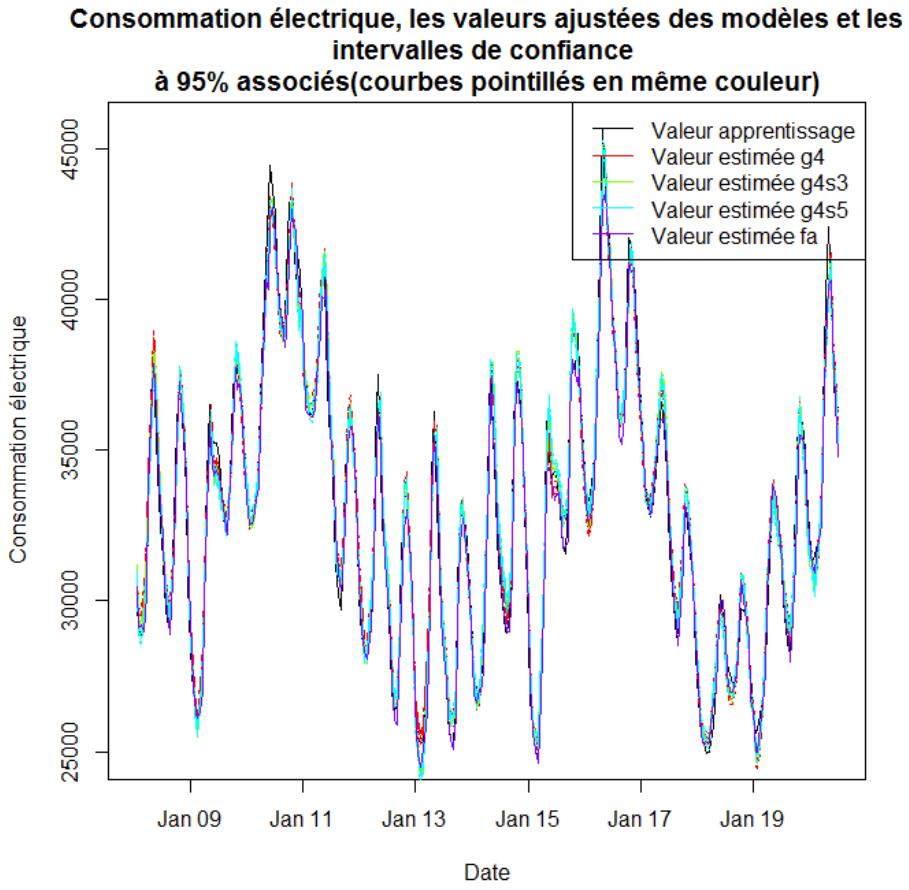


Figure 3.3.1 : Estimation

D'après la figure 3.3.1, on note que,

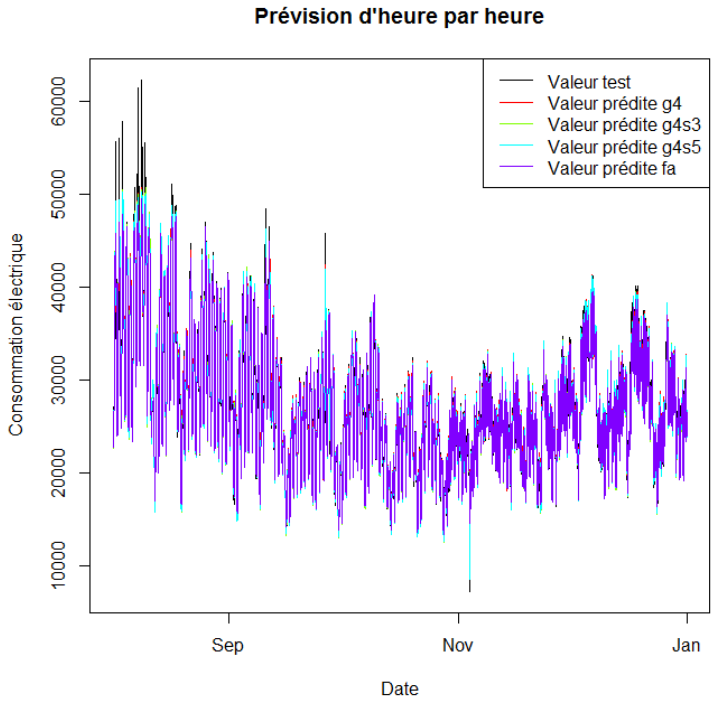
- Globalement, la qualité d'estimation est bonne.
  - Les courbes d'estimation collent bien à la courbe des vraies valeurs.
  - Les intervalles de confiance associés sont petits, ça signifie que les estimateurs sont stables (sa variance est faible), alors ils ont des bonnes qualités.
- Cependant, ces modèles estiment mal la consommation électrique exceptionnelle (au mi 2005 et mi 2006).

IV. Prédiction

Dans cette partie, on va utiliser les modèles précédents pour la prévision à 3 périodes. Prévision à très court terme (on prédit à une heure en utilisant les données actuelles), prévision à court terme (on prédit à une journée en utilisant les données actuelles) et prévision à moyen terme (on prédit à une semaine en utilisant les données actuelles). Par contre, dans les données de test, il n'y a que 3 mois, alors c'est difficile d'évaluer la prévision à long terme (mois par mois). On interprétera les résultats par 2 façons : la présentation graphique et les critères numériques (MAPE et RMSE). On rappelle que : **g4** est le modèle GAM avec la station 1, **g4s3** est le modèle GAM avec la station 3, **g4s5** est le modèle GAM avec la station 5, et **fa** la forêt aléatoire.

1. Présentation graphique

D'abord, on va représenter la prévision à très court terme (heure par heure).



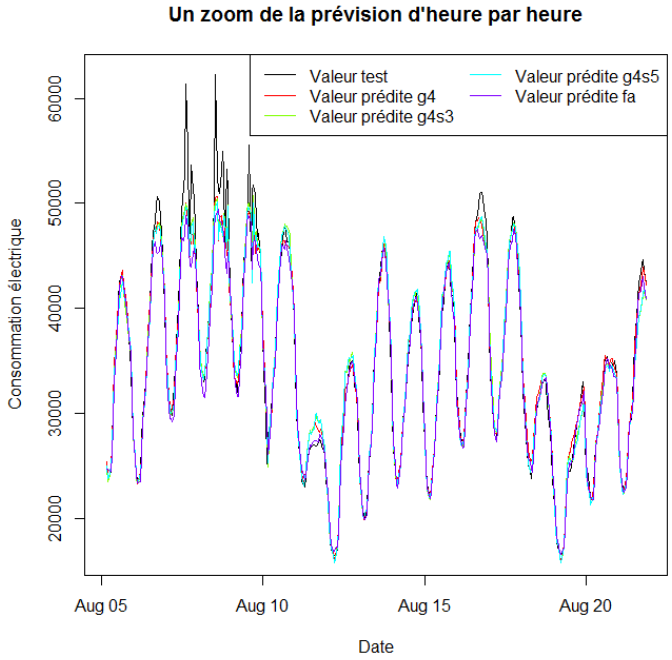


Figure 4.1.1 : Prévision d'heure par heure

D'après la figure 4.1.1, on constate que :

- Globalement, la qualité de la prévision à très court terme est bonne. Les courbes de prédiction collent bien à la courbe des vraies valeurs.
- Cependant, ces modèles ne prédisent pas bien les moments où la consommation électrique est exceptionnelle.
- Quand la consommation électrique est stable, les valeurs prédites de ces modèles s'approchent. Quand la consommation électrique est fluctuante, on trouve une différence entre les valeurs prédites des modèles.

Ensuite, on va observer la prévision à court terme (jour par jour).

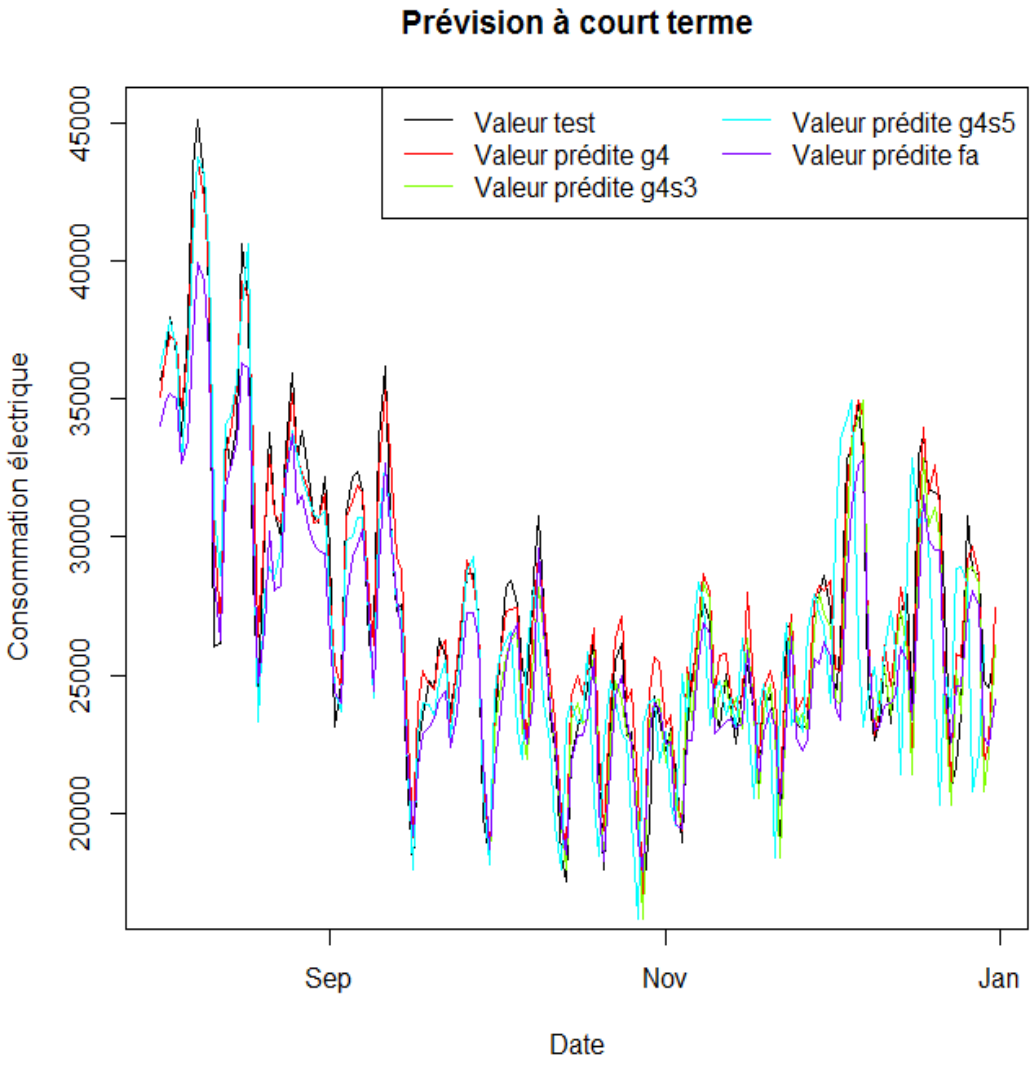


Figure 4.1.2 : Prévision de jour par jour

Dans la figure 4.1.2, on a moyenné la consommation électrique et ses valeurs prédites pendant une journée pour mieux observer la prévision de jour par jour.

D'après la figure 4.1.2, on constate que :

- Par rapport à la prévision à très court terme, la qualité de la prévision court terme a diminuée. On trouve une différence entre les vraies valeurs avec les valeurs prédites et la différence entre les valeurs prédites des modèles différents.
- Les valeurs prédites de la forêt aléatoire (courbe violette) sont souvent plus faibles que celles des autres modèles et plus faibles que les vraies valeurs.
- Dans le modèle g4s5 (courbe bleue), depuis mi-septembre, il y a un retard sur l'axe abscisse de ses valeurs prédites par rapport aux vraies valeurs.
- On peut dire que les modèles g4 et g4s3 prédisent mieux la consommation électrique.

Enfin, c'est la prévision à moyen terme (semaine par semaine).

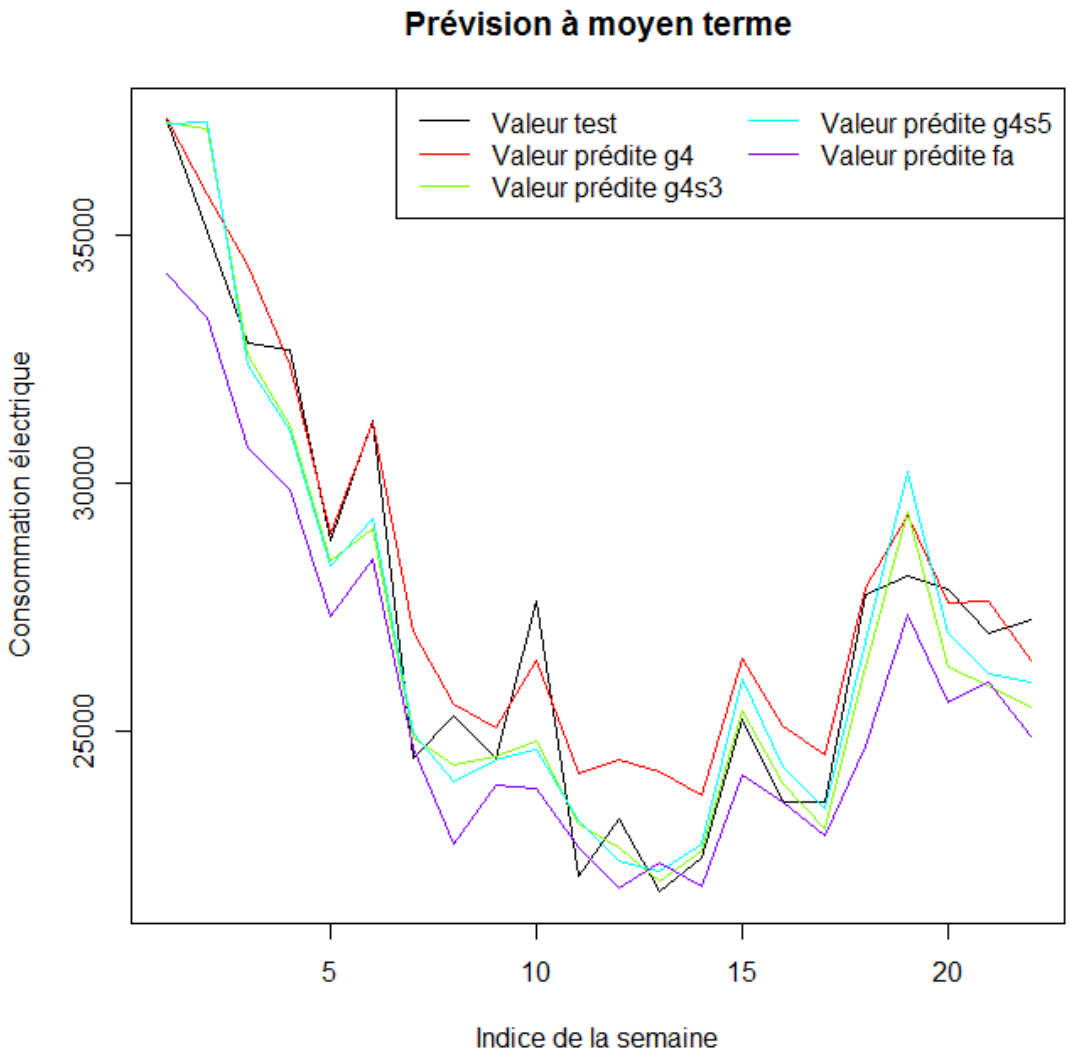


Figure 4.1.3 : Prévision de semaine par semaine

Comme la figure 4.1.2, on a moyenné la consommation électrique et ses valeurs prédites pendant une semaine.

En observant la figure 4.1.2, on note que :

- La qualité de prévision à moyen terme est encore pire.
- Les valeurs prédites de la forêt aléatoire (courbe violette) sont souvent plus faible et les valeurs prédites du modèle g4 (courbe rouge) sont souvent plus fortes que celles des autres modèles et les vraies valeurs.
- Les valeurs prédites du modèle g4s3 et g4s5 s'approchent.

2. Critères numériques

On utilise les 2 critères RMSE et MAPE définis ci-dessous (on suppose y_i les vraies valeurs et $\hat{y}_i$ valeurs prédites):

2.1 MAPE

$$\%\ MAPE = 100 \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| .$$

Plus MAPE s'approche de 0, plus la prévision est bonne.

2.2 RMSE

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} .$$

Plus RMSE s'approche de 0, plus la prévision est bonne.

2.3 Tableau des résultats

Modèle \ Période de prévision		Très court terme	Court terme	Moyen terme
g4	**MAPE %**	2.320	**5.251**	6.591
	RMSE	**1091.938**	**1960.742**	**2207.126**
g4s3	**MAPE %**	2.337	5.404	**6.309**
	RMSE	1117.951	2190.76	2433.364
g4s5	**MAPE %**	**2.354**	5.550	6.678
	RMSE	**1131.495**	2258.905	2544.081
fa	**MAPE %**	**2.252**	**6.349**	**7.802**
	RMSE	1123.427	**2554.847**	**3009.207**

Figure 4.2.1 : Tableau d'évaluation

Dans le tableau 4.2.1, pour chaque période de prévision, à chaque critère d'évaluation, la meilleure valeur est coloriée en bleu, la pire valeur est coloriée en rouge. D'après ce tableau, on note que :

- La qualité de prévision est diminuée au cours de très court terme à moyen terme. C'est évident, plus le temps de prévision est long, plus cette prévision est moins exacte.
- Globalement, le modèle g4 prédit le mieux la consommation électrique.

- La forêt aléatoire fonctionne bien pour la prévision à très court terme, pourtant, elle fonctionne trop pire pour la prévision à court terme et à moyen terme. Peut-être, parce qu'on a construit pas beaucoup d'arbre (250 arbres) ou la forêt aléatoire marche mieux pour la classification que la régression.
- Parmi 3 modèles GAM, modèle g4 est le meilleur, puis c'est modèle g4s3 et modèle g4s5 est le pire. Ce résultat est compatible au graphe de l'importance des covariables (cf. figure 3.2.1) : l'influence de la station1 > l'influence de la station3 > l'influence de la station5.

V. Conclusion

1. Synthèse

Dans cette partie, on résume les démarches et les résultats obtenus dans notre projet.

Ce projet vise à prédire la consommation électrique sur 3 périodes : heure par heure, jour par jour et semaine par semaine. Pour réaliser cet objectif, on a cherché à construire des modèles. D'où l'utilisation de 2 types de modèle : la forêt aléatoire et le modèle GAM. Dans chaque modèle, on a ajusté la consommation électrique d'abord en fonction des covariables puis en fonction de la consommation électrique au passé. On a aussi évalué ces modèles par le critère R2-ajusté (pour GAM) et variance explained (pour forêt aléatoire). A la fin, on a obtenu 3 modèles GAM avec R2-ajusté égale à 0.98, 0.979 et 0.979 et une forêt aléatoire avec variance explained égale à 0.9801.

Enfin, on utilise ces modèles pour la prévision sur 3 périodes objectif du projet. On a évalué la qualité de prédiction par 2 façons : l'interprétation graphique et les critères numériques (RMSE et MAPE).

2. Perspectives

Comme on a pu le constater à la figure 4.1.1, les modèles ne prédisent pas bien les valeurs exceptionnelles de la consommation électrique. Ces effets peuvent être associés avec une température exceptionnelle (trop chaude ou trop froide). Ainsi donc, nos modèles devraient être améliorés pour prendre en compte ces effets.

Par ailleurs, il est possible d'améliorer la qualité des modèles en mélangeant ces modèles. La méthode de mélange des prédicteurs est expliquée ci-dessous :

$i \in \{1 \dots d\}$ l'ensemble des modèles, chacun propose une prévision $f_{i,t}$ en moment t associée à une poids $\hat{p}_{i,t}$ à estimer. On a

$$\hat{y}_t = \sum_{i=1}^{d} \hat{p}_{i,t} f_{i,t} \text{ le prédicteur de mélange.}$$

3. Compétences acquises

Ce projet nous a permis d'avoir des connaissances sur des méthodes d'ajustement de prédiction les données compliquées et les appliquer aux problèmes précis dans la vie : prédiction de la consommation électrique. De plus, on a accueilli des expériences pour réaliser un projet scientifique : lire des articles scientifiques, traiter des nouveaux problèmes par autonomie, rédiger un rapport. Par ailleurs, c'était une opportunité pour résoudre des gros problèmes sur R, ce qui nous a aidé à renforcer nos compétence de R.

VI.ANNEXE

1. Fonctions

```
# Input: données d'apprentissage, données de test, model, nahead (1 pour prévision heure par heure, 24 pour jour par jour, 24*7 pour semaine par semaine)
# Output: Le vecteur de prédiction
# Forme du modèle: Variable réponse ~ Les covariables + termes autorégressifs (lag1,lag2,lag3,lag4,lag5,lag24,lag48,lag72,lag96,lag120,lag144,lag168)

prediction = function(donapp,dontest,model,nahead){
	napp = length(donapp$Zone10)
	ntest = length(dontest$Zone10)
	ntot = napp+ntest
	j=0
	pred = c()

	while ((napp+nahead)<=ntot){
		j=j+1
		p = numeric(nahead)
		for (i in 1:nahead){
			indice = nahead*(j-1)+i
			# Par exemple: on prédit à 24 heures, alors lag1 à lag24 est pris dans l'ensemble prédit, lag48... lag168 est pris dans l'ensemble des vraies données.
			k=1;	lag1  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=2;	lag2  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=3;	lag3  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=4;	lag4  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=5;	lag5  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=24;	lag24 = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=48;	lag48 = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=72;	lag72 = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=96;	lag96 = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=120;	lag120= if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=144;	lag144= if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			k=168;	lag168= if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
			# Créer un data.frame de 1 élément composant des covariables et des termes autorégressifs pour prédire
			dt= data.frame(dontest[indice,],lag1,lag2,lag3,lag4,lag5,lag24,lag48,lag72,lag96,lag120,lag144,lag168);
			p[i] = predict(model,dt)
		}
		donapp = rbind(donapp,dontest[(nahead*(j-1)+1):(nahead*j),])
		napp = napp + nahead
		pred = c(pred,p)
	}
	# Prédire des éléments restants. Par exemple: on a 100 éléments et prédire de jour par jour. 96 premiers elements sont prédits par la boucle précédente, 4 autres sont prédits par cette boucle
	if (napp+nahead>ntot){
		j = j+1
		p = numeric(ntot-napp)
		for (i in 1:(ntot-napp)){
```

```
                        indice = nahead*(j-1)+i
                        k=1;    lag1  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=2;    lag2  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=3;    lag3  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=4;    lag4  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=5;    lag5  = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=24;   lag24 = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=48;   lag48 = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=72;   lag72 = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=96;   lag96 = if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=120;  lag120= if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=144;  lag144= if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];
                        k=168;  lag168= if (i<=k) donapp$Zone10[napp+i-k] else p[i-k];

        dt=data.frame(dontest[indice,],lag1,lag2,lag3,lag4,lag5,lag24,lag48,lag72,lag96,lag120,lag14
4,lag168);
                        p[i] = predict(model,dt)
                }
                pred = c(pred,p)
        }
        return(pred)
}


# Critère d'evaluation
mape <- function(y, yhat) 100*mean(abs((y - yhat)/y))
rmse <- function(y, yhat) sqrt(mean((y - yhat)^2))
```

2. Programmes

```
#------------------------------------------------------------------
#INITIALISATION
#------------------------------------------------------------------

rm(list=objects())
library(mgcv)
library(date)
library(randomForest)

chemin<-"C:/TP/Datamining/DONNEES US/"
fich<-"data_elec0.txt"
Data0<-read.table(paste(chemin,fich,sep=""),sep='\t',header=TRUE,dec='.')
fich<-"data_elec1.txt"
Data1<-read.table(paste(chemin,fich,sep=""),sep='\t',header=TRUE,dec='.')
Data0$date=as.POSIXct(Data0$date,format="%Y-%m-%d %H:%M:%S", tz = "UTC")
Data1$date=as.POSIXct(Data1$date,format="%Y-%m-%d %H:%M:%S", tz = "UTC")

names(Data0)
# [1] "Year"     "Month"    "Day"      "Hour"     "Time"     "Toy"      "dow"      "daytype"  "Zone1"
"Zone2"
```

```
#[11] "Zone3"    "Zone4"    "Zone5"    "Zone6"    "Zone7"    "Zone8"    "Zone9"    "Zone10"
"Zone11"   "Zone12"
#[21] "Zone13"   "Zone14"   "Zone15"   "Zone16"   "Zone17"   "Zone18"   "Zone19"   "Zone20"
"Station1"  "Station2"
#[31] "Station3"  "Station4"  "Station5"  "Station6"  "Station7"  "Station8"  "Station9"  "Station10"
"Station11"

# Créer Hourf Yearf Monthf Dayf la forme qualitative des covariables Hour Year Month Day
Data0 = data.frame(Data0,Hourf= as.factor(Data0$Hour),Yearf= as.factor(Data0$Year),Monthf =
as.factor(Data0$Month),Dayf = as.factor(Data0$Day))

# Comme j'ai un problème de données test, je dois diviser les données apprentissage pour faire le
projet
# On peut voir ce problème en executant ce code
plot(c(Data0$Zone10,Data1$Zone10))
# Pourtant, dans la partie statistique exploratoire, j'utilise toutes les données Data0
split = 30048
donnees=Data0[1:split,]
test = Data0[(split+1):length(Data0$date),]

#--------------------------------------------------------------------------
# STATISTIQUE EXPLORATOIRE
#--------------------------------------------------------------------------
attach(Data0)

# Tendance et saisonnalité
par(mfrow=c(2,1))
# Lissage
l = lowess(date,Zone10)
plot(date,Zone10,type="l",main="La consommation électrique avec sa courbe de lissage")
lines(l,col="red",lwd=3)
plot(l,type="l",main="Courbe de lissage")

# Saisonnalité de la consommation électrique comparant avec celle des stations de météo
fac = Monthf
#Moyenner par mois
x=tapply(Zone10,fac,mean)
s1 = tapply(Station1,fac,mean)
s3 = tapply(Station3,fac,mean)
s5 = tapply(Station5,fac,mean)
par(mfrow=c(2,1))
plot(x,type="l",main="La consommation électrique moyenne pour chaque mois")
plot(s1,type="l",ylim=c(30,90),main="La température moyenne pour chaque mois",col="green")
lines(s3,col="red")
lines(s5,col="blue")
legend("topleft",legend=c("Station1","Station3","Station5"),col=c("green","red","blue"),lty=1)

# ACF
acf(Zone10,lag.max=24*35,main="ACF")

# Saisonnalité hebdomadaire
```

```
fac = dow:Hourf
#Moyenner toutes les 1 heure de lundi, toutes les 2 heure de lundi…toutes les 00h de samedi…
x=tapply(Zone10,fac,mean)
nomJour = c("lundi","mardi","mercredi","jeudi","vendredi","samedi","dimanche")
col = rainbow(7) #creer automatiquement un vecteur des couleurs en utilisant les couleurs de l'arc en ciel.
# Chercher les valeurs moyennes des heures à lundi
sel = grep(nomJour[1],names(x))
plot(x[sel],type="l",col=col[1],,xlab="Heure",ylab="Zone10",ylim=c(15000,40000))
# Idem pour les jours après
for(i in 2:7){
  sel = grep(nomJour[i],names(x))
  lines(x[sel],type="l",col=col[i])
}
legend("topleft",legend=nomJour,col=col,lty="solid",ncol=4)

detach(Data0)

#------------------------------------------------------------
#Random Forest
#------------------------------------------------------------

attach(donnees)

# Foret1 (sans terme autoregressif)
model <- randomForest(Zone10 ~ Station1 + Station2 + Station3 + Station4 + Station5 +
Station6+Station7+Station8+Station9+Station10+Station11+Year+Monthf+Dayf+Hourf+Toy+dow+da
ytype+Time, data=donnees, na.action=na.omit,ntree=250)
#ACF de résidus
resid = Zone10-model$predicted
acf(resid,lag=32*24,main="foret1$resid, lag.max=24*32")

# Foret2 (avec terme autoregressif)
# Préparation des valeurs de la consommation électrique au passé
lagtot = 168
nr = length(Zone10)
lag1 = Zone10[lagtot:(nr-1)];
lag2 = Zone10[(lagtot-1):(nr-2)];
lag3 = Zone10[(lagtot-2):(nr-3)];
lag4 = Zone10[(lagtot-3):(nr-4)];
lag5 = Zone10[(lagtot-4):(nr-5)];
lag24 = Zone10[(lagtot-23):(nr-24)];
lag48 = Zone10[(lagtot-47):(nr-48)];
lag72 = Zone10[(lagtot-71):(nr-72)];
lag96 = Zone10[(lagtot-95):(nr-96)];
lag120 = Zone10[(lagtot-119):(nr-120)];
lag144 = Zone10[(lagtot-143):(nr-144)];
lag168 = Zone10[(lagtot-167):(nr-168)];
#Ajouter dans data.frame
```

```
donneesr =
data.frame(donnees[(lagtot+1):nr,],lag1,lag2,lag3,lag4,lag5,lag24,lag48,lag72,lag96,lag120,lag144,la
g168)
# Foret2
model <- randomForest(Zone10 ~ Station1 + Station2 + Station3 + Station4 + Station5 +
Station6+Station7+Station8+Station9+Station10+Station11+Year+Monthf+Dayf+Hourf+Toy+dow+da
ytype+Time+lag1+lag2+lag3+lag4+lag5+lag24+lag48+lag72+lag96+lag120+lag144+lag168,
data=donneesr, importance=TRUE, na.action=na.omit,ntree=250)
#ACF de résidus
acf(Zone10-model$predicted,lag.max=24*32)

#-------------------------------------------------------------
#Model GAM
#-------------------------------------------------------------

#ETAPE 1

#Importance des covariables
# Option Importance = TRUE pour calculer l'importance
model <- randomForest(Zone10 ~ Station1 + Station2 + Station3 + Station4 + Station5 +
Station6+Station7+Station8+Station9+Station10+Station11+Year+Monthf+Dayf+Hourf+Toy+dow+da
ytype+Time, data=donnees, importance=TRUE, na.action=na.omit,ntree=150)
# Tracer graphe de l'importance, type=1 : l'importance par comparaison de la prédiction avant et
après une permutation des valeurs out of bag, type=2 : l'importance de Gini
varImpPlot(model, type = 2, main = "Importance des variables pour la prévision de qualité de
Zone10")

#Analyse l'effet temperature-mois
col = rainbow(12)
sel = which(Monthf==1)
#Moyenner et tracer la consommation des mois en fonction de température
sf = as.factor(Station1[sel])
m = tapply(Zone10[sel],sf,mean)
plot(as.numeric(levels(sf)),m,type="l",lwd=3,col=col[1],xlab="Température",ylab="Zone10",xlim=c(1
0,105),ylim=c(10000,60000))
# Idem pour les mois 2 à 12
for (i in 2:12){
sel = which(Monthf==i)
sf = as.factor(Station1[sel])
m = tapply(Zone10[sel],sf,mean)
lines(as.numeric(levels(sf)),m,col=col[i],lwd=3)
}
nomMois =
c("janvier","février","mars","avril","mai","juin","juillet","aout","septembre","octobre","novembre","
décembre")
legend("top",legend=nomMois,col=col,lty="solid",ncol=2,lwd=3)


#Construction g0
# Base thin-plat-spline avec l'interaction s(covariable,by=variable_interaction)
# Base thin-plat-spline simple s(covariable)
```

```
# Il existe l'option "bs" pour préviser le type du base spline (par défaut : thin plate spline), « k » pour limiter la dégre liberté maximale (par défaut : k=10).
g0<- gam(Zone10~s(Station1,by=Monthf),data=donnees)
summary(g0)

#Re-analyse l'effet temperature-mois (idem précédent, juste remplacer « Zone10 » par « g0$resid »)
col = rainbow(12)
sel = which(Monthf==1)
sf = as.factor(Station1[sel])
m = tapply(g0$resid[sel],sf,mean)
plot(as.numeric(levels(sf)),m,type="l",lwd=3,col=col[1],xlab="Température",ylab="g0$residuals",xlim=c(10,105),ylim=c(-9000,12000))
for (i in 2:12){
sel = which(Monthf==i)
sf = as.factor(Station1[sel])
m = tapply(g0$resid[sel],sf,mean)
lines(as.numeric(levels(sf)),m,col=col[i],lwd=3)
}
nomMois =
c("janvier","février","mars","avril","mai","juin","juillet","aout","septembre","octobre","novembre","décembre")
legend("top",legend=nomMois,col=col,lty="solid",ncol=2,lwd=3)


#-----------------------------------------------------------
#ETAPE 2

#Importance des covariables
model <- randomForest(g0$residuals ~ Station1 + Station2 + Station3 + Station4 + Station5 + Station6+Station7+Station8+Station9+Station10+Station11+Year+Monthf+Dayf+Hourf+Toy+dow+daytype+Time, data=donnees, importance=TRUE, na.action=na.omit, ntree=150)
varImpPlot(model, type = 2, main = "Importance des variables pour la prévision de qualité de\n Zone10~s(Station1,by=Monthf)$residuals")


#Analyse l'effet Hour-dow
#Moyenner toutes les 1 heure de lundi, toutes les 2 heure de lundi…toutes les 00h de samedi
fac = dow:Hourf
x=tapply(g0$resid,fac,mean)
nomJour = c("lundi","mardi","mercredi","jeudi","vendredi","samedi","dimanche")
col = rainbow(7)
#Chercher les valeurs moyennes des heures de lundi
sel = grep(nomJour[1],names(x))
plot(x[sel],type="l",col=col[1],lwd=3,xlab="Heure",ylab="g0$resid",ylim=c(10000,50000))
#Idem pour Mardi à dimanche
for(i in 2:7){
  sel = grep(nomJour[i],names(x))
  lines(x[sel],type="l",col=col[i],lwd=3)
}
legend("topleft",legend=nomJour,col=col,lty="solid",ncol=2,lwd=3)
```

```
#Construction g1
g1<- gam(Zone10~s(Hour,by=dow)+s(Station1,by=Monthf),data=donnees)
summary(g1)

#Re-Analyse l'effet Hour-dow, idem précédent juste remplacer g0$resid par g1$resid
fac = dow:Hourf
x=tapply(g1$residuals,fac,mean)
nomJour = c("lundi","mardi","mercredi","jeudi","vendredi","samedi","dimanche")
col = rainbow(7)
sel = grep(nomJour[1],names(x))
plot(x[sel],type="l",lwd=3,col=col[1],xlab="Heure",ylab="g1$residuals",ylim=c(-7000,7000))
for(i in 2:7){
  sel = grep(nomJour[i],names(x))
  lines(x[sel],type="l",col=col[i],lwd=3)
}
legend("topleft",legend=nomJour,col=col,lty="solid",ncol=2,lwd=3)

#-------------------------------------------------------
#ETAPE 3

#Importance des covariables
model <- randomForest(g1$residuals ~ Station1 + Station2 + Station3 + Station4 + Station5 +
Station6+Station7+Station8+Station9+Station10+Station11+Year+Monthf+Dayf+Hourf+Toy+dow+da
ytype+Time, data=donnees, importance=TRUE, na.action=na.omit, ntree=200)
varImpPlot(model, type = 2, main = "Importance des variables pour la prévision de qualité \n de
Zone10~s(Station1,by=Monthf)+s(Hour,by=dow)$residuals")

#Construction g2
g2<- gam(Zone10~s(Hour,by=dow)+s(Station1,by=Monthf)+daytype,data=donnees)
summary(g2)

#Re-Analyse l'effet Hour-dow, idem précédent, juste remplacer g1$resid par g2$resid
fac = dow:Hourf
x=tapply(g2$residuals,fac,mean)
nomJour = c("lundi","mardi","mercredi","jeudi","vendredi","samedi","dimanche")
col = rainbow(7)
sel = grep(nomJour[1],names(x))
plot(x[sel],type="l",lwd=3,col=col[1],xlab="Heure",ylab="g2$residuals",ylim=c(-7000,7000))
for(i in 2:7){
  sel = grep(nomJour[i],names(x))
  lines(x[sel],type="l",col=col[i],lwd=3)
}
legend("topleft",legend=nomJour,col=col,lty="solid",ncol=2,lwd=3)

#-------------------------------------------------------
#ETAPE 4

#Importance des covariables
model <- randomForest(g2$residuals ~ Station1 + Station2 + Station3 + Station4 + Station5 +
Station6+Station7+Station8+Station9+Station10+Station11+Year+Monthf+Dayf+Hourf+Toy+dow+da
ytype+Time, data=donnees, importance=TRUE, na.action=na.omit, ntree=200)
```

```
varImpPlot(model, type = 2, main = "Importance des variables pour la prévision de qualité de\n
Zone10~s(Station1,by=Monthf)+s(Hour,by=dow)+daytype$resid")

# Analyse la saison et la tendance
par(mfrow=c(2,1))
#Courbe de lissage
l = lowess(date,g2$resid)
plot(date,g2$resid,type="l",main="g2$resid avec sa courbe de lissage")
lines(date,l$y,col="red",lwd=1)
plot(date,l$y,type="l",main="Courbe de lissage")

# Construction g3
# bs=”cc” base cyclique, k=3 limité le degré liberté de la variable Time => limiter la dimension de
spline
g3<-
gam(Zone10~s(Hour,by=dow)+s(Station1,by=Monthf)+daytype+s(Hour,by=Monthf)+s(Time,k=3)+s(T
oy,bs="cc"),data=donnees)
summary(g3)

#-------------------------------------------------------------
#ETAPE 5

# ACF
acf(g3$residuals,lag.max=24*10,main="acf de g3$resid, lag.max=24*10")

# Rajouter les termes autoregressifs
lagtot = 168
nr = length(Zone10)
lag1 = Zone10[lagtot:(nr-1)];
lag2 = Zone10[(lagtot-1):(nr-2)];
lag3 = Zone10[(lagtot-2):(nr-3)];
lag4 = Zone10[(lagtot-3):(nr-4)];
lag5 = Zone10[(lagtot-4):(nr-5)];
lag24 = Zone10[(lagtot-23):(nr-24)];
lag48 = Zone10[(lagtot-47):(nr-48)];
lag72 = Zone10[(lagtot-71):(nr-72)];
lag96 = Zone10[(lagtot-95):(nr-96)];
lag120 = Zone10[(lagtot-119):(nr-120)];
lag144 = Zone10[(lagtot-143):(nr-144)];
lag168 = Zone10[(lagtot-167):(nr-168)];
donneesr =
data.frame(donnees[(lagtot+1):nr,],lag1,lag2,lag3,lag4,lag5,lag24,lag48,lag72,lag96,lag120,lag144,la
g168)

#Construction g4
g4<-
gam(Zone10~s(Hour,by=dow)+s(Station1,by=Monthf)+daytype+s(Hour,by=Monthf)+Dayf+s(Time,k=
3)+s(Toy,bs="cc")+s(lag1)+s(lag2)+s(lag3)+s(lag4)+s(lag5)+s(lag24)+s(lag48)+s(lag72)+s(lag96)+s(lag1
20)+s(lag144)+s(lag168),data=donneesr)
summary(g4)
```

```
#-------------------------------------------------------------------------------
#Construction g4s3
g4s3<-
gam(Zone10~s(Hour,by=dow)+s(Station3,by=Monthf)+daytype+s(Hour,by=Monthf)+Dayf+s(Time,k=
3)+s(Toy,bs="cc")+s(lag1)+s(lag2)+s(lag3)+s(lag4)+s(lag5)+s(lag24)+s(lag48)+s(lag72)+s(lag96)+s(lag1
20)+s(lag144)+s(lag168),data=donneesr)
summary(g4s3)

#Construction g4s5
g4s5<-
gam(Zone10~s(Hour,by=dow)+s(Station5,by=Monthf)+daytype+s(Hour,by=Monthf)+Dayf+s(Time,k=
3)+s(Toy,bs="cc")+s(lag1)+s(lag2)+s(lag3)+s(lag4)+s(lag5)+s(lag24)+s(lag48)+s(lag72)+s(lag96)+s(lag1
20)+s(lag144)+s(lag168),data=donneesr)
summary(g4s5)

#-------------------------------------------------------------------------------
#Interpretation graphique
detach(donnees)
attach(donneesr)

# se.fit=TRUE pour calculer l'estimateur de l'ecart type
p4 = predict(g4,se.fit=TRUE)
p4s3 = predict(g4s3,se.fit=TRUE)
p4s5 = predict(g4s5,se.fit=TRUE)
# Calculer l'interval de confiance à 95%
upr4 = p4$fit + 1.96*p4$se.fit
lwr4 = p4$fit - 1.96*p4$se.fit
upr43 = p4s3$fit + 1.96*p4s3$se.fit
lwr43 = p4s3$fit - 1.96*p4s3$se.fit
upr45 = p4s5$fit + 1.96*p4s5$se.fit
lwr45 = p4s5$fit - 1.96*p4s5$se.fit

# Superposer les donnees + valeurs ajustées
col=rainbow(4)
plot(date,Zone10,type="l",xlab="Date",ylab="Consommation électrique",main="Consommation
électrique et les valeurs ajustées")
lines(date,p4$fit,col=col[1])
lines(date,p4s3$fit,col=col[2])
lines(date,p4s5$fit,col=col[3])
lines(date,model$pred,col=col[4])
legend("topright",legend=c("Valeur apprentissage","Valeur estimée g4","Valeur estimée
g4s3","Valeur estimée g4s5","Valeur estimée fa"),lty=1,col=c(1,col))

# Zoomer + interval de confiance à 95%
plot(date[1:300],Zone10[1:300],type="l",xlab="Date",ylab="Consommation
électrique",main="Consommation électrique, les valeurs ajustées des modèles et les\nintervalles de
confiance associés(courbes pointillés en même couleur)")
lines(date[1:300],p4$fit[1:300],col=col[1])
lines(date[1:300],upr4[1:300],col=col[1],lty=2)
lines(date[1:300],lwr4[1:300],col=col[1],lty=2)
lines(date[1:300],p4s3$fit[1:300],col=col[2])
```

```
lines(date[1:300],upr43[1:300],col=col[2],lty=2)
lines(date[1:300],lwr43[1:300],col=col[2],lty=2)
lines(date[1:300],p4s5$fit[1:300],col=col[3])
lines(date[1:300],upr45[1:300],col=col[3],lty=2)
lines(date[1:300],lwr45[1:300],col=col[3],lty=2)
lines(date[1:300],model$pred[1:300],col=col[4])
legend("topright",legend=c("Valeur apprentissage","Valeur estimée g4","Valeur estimée
g4s3","Valeur estimée g4s5","Valeur estimée fa"),lty=1,col=c(1,col))



#-------------------------------------------------------
#PREDICTION
#-------------------------------------------------------

detach(donneesr)
attach(test)
source("C:/TP/Datamining/DONNEES US/fonction/prediction.R")
source("C:/TP/Datamining/DONNEES US/fonction/mape.R")
source("C:/TP/Datamining/DONNEES US/fonction/rmse.R")

load("C:/TP/Datamining/DONNEES US/g4")
p = predict(g4,se.fit=TRUE)

# Prédiction jour par jour
predjg4 = prediction(donnees,test,g4,24)
predjg4s3 = prediction(donnees,test,g4s3,24)
predjg4s5 = prediction(donnees,test,g4s5,24)
predjfa = prediction(donnees,test,model,24)

# Prédiction heure par heure
predhg4 = prediction(donnees,test,g4,1)
predhg4s3 = prediction(donnees,test,g4s3,1)
predhg4s5 = prediction(donnees,test,g4s5,1)
predhfa = prediction(donnees,test,model,1)

# Prédiction semaine par semaine
predsg4 = prediction(donnees,test,g4,24*7)
predsg4s3 = prediction(donnees,test,g4s3,24*7)
predsg4s5 = prediction(donnees,test,g4s5,24*7)
predsfa = prediction(donnees,test,model,24*7)


# Interpretation graphique de prédiction à très court terme (l'heure par l'heure)
col=rainbow(4)
plot(date,Zone10,type="l",xlab="Date",ylab="Consommation électrique",main="Prévision d'heure
par heure")
lines(date,predhg4,col=col[1])
lines(date,predhg4s3,col=col[2])
lines(date,predhg4s5,col=col[3])
lines(date,predhfa,col=col[4])
```

```
legend("topright",legend=c("Valeur test","Valeur prédite g4","Valeur prédite g4s3","Valeur prédite g4s5","Valeur prédite fa"),lty=1,col=c(1,col))
# Zoom
plot(date[100:500],Zone10[100:500],type="l",xlab="Date",ylab="Consommation électrique",main="Un zoom de la prévision d'heure par heure")
lines(date[100:500],predhg4[100:500],col=col[1])
lines(date[100:500],predhg4s3[100:500],col=col[2])
lines(date[100:500],predhg4s5[100:500],col=col[3])
lines(date[100:500],predhfa[100:500],col=col[4])
legend("topright",legend=c("Valeur test","Valeur prédite g4","Valeur prédite g4s3","Valeur prédite g4s5","Valeur prédite fa"),lty=1,col=c(1,col),ncol=2)

# Interpretation graphique de prédiction à courte terme (jour par jour)
fac = as.factor(Year):as.factor(test$Month):Dayf #si on utilise Monthf ici, on a les niveaux de 1..12 au lieu de 8...12, idem Yearf niveaux de 2004 2007 au lieu de 2007
# Changer le facteur au type date
datej = as.POSIXct(levels(fac),format="%Y:%m:%d")
datej = datej[!is.na(datej)] # Car il donne NA pour 31/11, 31/9 (il n'existe pas ces jours)
# Moyenner la consommation d'électrique pendant une journée, supprimer les NA pour 31/11 et 31/9
Zone10jm = tapply(Zone10,fac,mean)
Zone10jm = Zone10jm[!is.na(Zone10jm)]
predjg4m = tapply(predjg4,fac,mean)
predjg4m = predjg4m[!is.na(predjg4m)]
predjg4s3m = tapply(predjg4s3,fac,mean)
predjg4s3m = predjg4s3m[!is.na(predjg4s3m)]
predjg4s5m = tapply(predjg4s5,fac,mean)
predjg4s5m = predjg4s3m[!is.na(predjg4s5m)]
predjfam = tapply(predjfa,fac,mean)
predjfam = predjfam[!is.na(predjfam)]
# Présentation graphique
col =rainbow(4)
plot(datej,Zone10jm,type="l",xlab="Date",ylab="Consommation électrique",main="Prévision à court terme")
lines(datej,predjg4m,col=col[1],type="l")
lines(datej,predjg4s3m,col=col[2],type="l")
lines(datej,predjg4s5m,col=col[3],type="l")
lines(datej,predjfam,col=col[4],type="l")
legend("topright",legend=c("Valeur test","Valeur prédite g4","Valeur prédite g4s3","Valeur prédite g4s5","Valeur prédite fa"),lty=1,col=c(1,col),ncol=2)

# Interpretation graphique de prédiction à moyen terme (semaine par semaine)
facsem = numeric(length(Zone10))
j=1;
#Creer un vecteur facsem de facteur c(1,1 ,1,1,1,1,1,2,2,2,2,2,2,2…) pour calculer la moyenne par semaine
for (i in (1:length(Zone10))){ facsem[i] = j; if (i%%(24*7)==0) j = j+1;}
facsem = as.factor(facsem)
#Moyenner la consommation électrique pendant une semaine
Zone10sm = tapply(Zone10,facsem,mean)
predsg4m = tapply(predsg4,facsem,mean)
```

```
predsg4s3m = tapply(predsg4s3,facsem,mean)
predsg4s5m = tapply(predsg4s5,facsem,mean)
predsfam = tapply(predsfa,facsem,mean)
col=rainbow(4)
#Présentation graphique
plot(Zone10sm,type="l",main="Prévision à moyen terme",xlab="Indice de la
semaine",ylab="Consommation électrique")
lines(predsg4m,type="l",col=col[1])
lines(predsg4s3m,type="l",col=col[2])
lines(predsg4s5m,type="l",col=col[3])
lines(predsfam,type="l",col=col[4])
legend("topright",legend=c("Valeur test","Valeur prédite g4","Valeur prédite g4s3","Valeur prédite
g4s5","Valeur prédite fa"),lty=1,col=c(1,col),ncol=2)

#---------------------------------------------------------------------------
# CRITERE D'EVALUATION MAPE et RMSE
#---------------------------------------------------------------------------

# HEURE
mape(Zone10,predhg4);          rmse(Zone10,predhg4)
mape(Zone10,predhg4s3);        rmse(Zone10,predhg4s3)
mape(Zone10,predhg4s5);        rmse(Zone10,predhg4s5)
mape(Zone10,predhfa);          rmse(Zone10,predhfa)


# JOUR
mape(Zone10,predjg4);          rmse(Zone10,predjg4)
mape(Zone10,predjg4s3);        rmse(Zone10,predjg4s3)
mape(Zone10,predjg4s5);        rmse(Zone10,predjg4s5)
mape(Zone10,predjfa);          rmse(Zone10,predjfa)

#SEMAINE
mape(Zone10,predsg4);          rmse(Zone10,predsg4)
mape(Zone10,predsg4s3);        rmse(Zone10,predsg4s3)
mape(Zone10,predsg4s5);        rmse(Zone10,predsg4s5)
mape(Zone10,predsfa);          rmse(Zone10,predsfa)
```