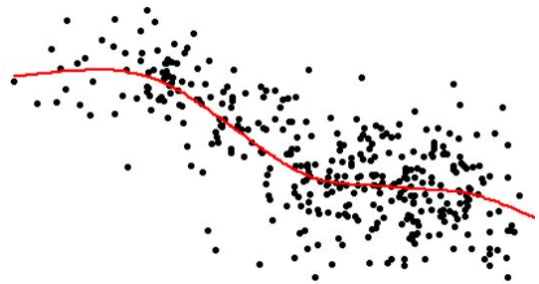
GAM en pratique: le package mgcv



Yannig Goude

EDF R&D yannig.goude@edf.fr

le package *mgcv* développé par Simon Wood (University of Bath) (voir [Wood(2006)] notamment) est une implémentation efficace et simple d'usage des modèles GAM

- le chargement du package se fait par la commande

  ```
  library(mgcv)
  ```

- les fonctions principales sont *gam* et son alternative pour les gros volumes de données *bam*

Nous verrons dans ce cours les fonctions utiles à la mise en place de méthodes de prévision de consommation électrique. Les possibilités du package sont bien plus vastes.

Syntaxe de base de la fonction *gam*:

```
gam(y~x0+s(x1)+s(x2)+s(x3,x4),data=dat)
```

- le modèle est entré grâce à l'objet *formula* de R -utilisé par exemple dans la commande *lm* de la régression-
- les effets non-linéaire mono ou bivariés sont indiqués par la fonction *s*
- les données d'estimation sont précisées par l'instruction *data*= et doivent être de type *data.frame*

Syntaxe de la fonction *s*:

```
s(x, k=10,bs="tp",sp=0.01)
```

- *k* correspond au nombre de degrés de libertés max de la fonction -estimated degrees of freedom est contraint à être inférieur à cette valeur-
- *bs* permet de choisir le type de spline de régression parmi: "tp", "ts", "ds", "cr", "cs", "cc", "ps", "cp", "sos"
- "tp": thin plate splines, par défaut dans la fonction *s*, calcul rapide mais impossibilité de positionnement des noeuds, "ts" est la version "shrinkage" de "tp" -permet de mettre un l'effet d'une variable à zéro-
- "ds": spline de duchon, généralisation des "tp" dans le cas multivarié
- "cr": cubic regression splines, "cs" est la version "shrinkage" de "cr"
- "cc": cyclic splines, adapté aux effets périodiques
- "ps": penalised splines, B-spline avec pénalisation sur les coefficients -contraintes de régularité-
- "cp: cyclic penalised splines
- "sos": thin plate splines sur la sphère
- *sp* permet de fixer une valeur de pénalisation -valeur optimisée automatiquement sinon-

Positionnement des noeuds: argument *knots* de la fonction *gam*

```
gam(y~s(x1,k=3),knots=list(x1=c(5,10,20)))
```

par défaut les noeuds sont équirépartis sur une grille $[\min(x), \max(x)]$

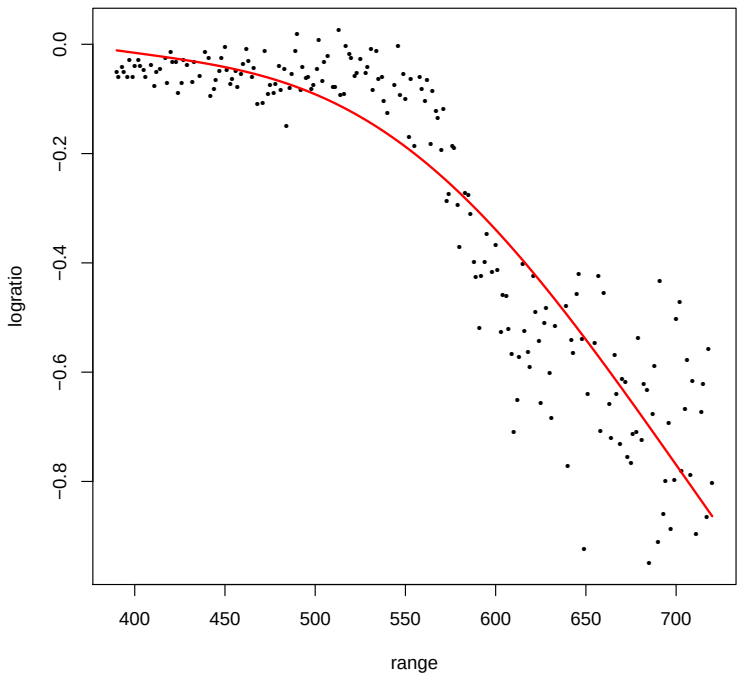
Choix de k, une *recette de cuisine* -par défaut $k = 10$-

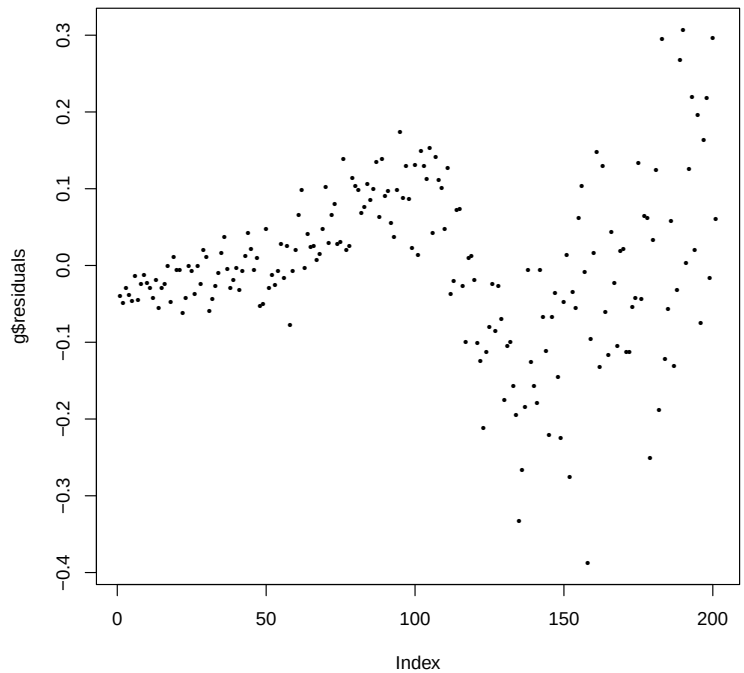
- observation du nuage de point, évaluation du nombre de degrés de liberté nécessaire
- choisir un k "élevé" par rapport à ce nombre
- estimer le modèle et regarder les degrés de libertés estimés
- si ces degrés de libertés sont proches de k augmenter k
- si ces degrés de libertés sont significativement plus faible que k diminuer k

L'analyse du nuage de point et de l'effet estimé:

- fonction non-régulière: probablement trop de noeuds, k est trop grand
- nuage de point présentant une forme: k est trop petit

Exemple de k trop faible: lidar data





Effets bi(ou multi)variés:

```
gam(y~s(x3,x4),data=dat)
```

ou

```
gam(y~te(x3,x4),data=dat)
```

- *s*: une pénalité globale
- *te*: une pénalité par axe (par variable) -produit tensoriel de fonctions de base-

Choix des bases, degrés de liberté max:

```
gam(y~s(x3,x4,k=c(10,15),bs=c('cr','cc'),data=dat)
```

```
gam(y~te(x3,x4,k=c(10,15),bs=c('cr','cc')),data=dat)
```

Attention: les degrés de liberté explosent rapidement en multivarié, à moins d'avoir un très grand nombre de données (et dans ce cas calculs lourds) on dépasse rarement la dimension 2.

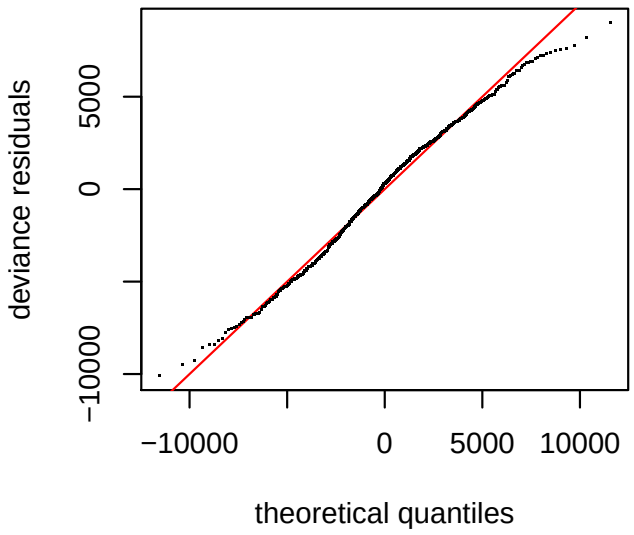
```
g=gam(y~s(x1)+s(x1,x2),data=dat)
```

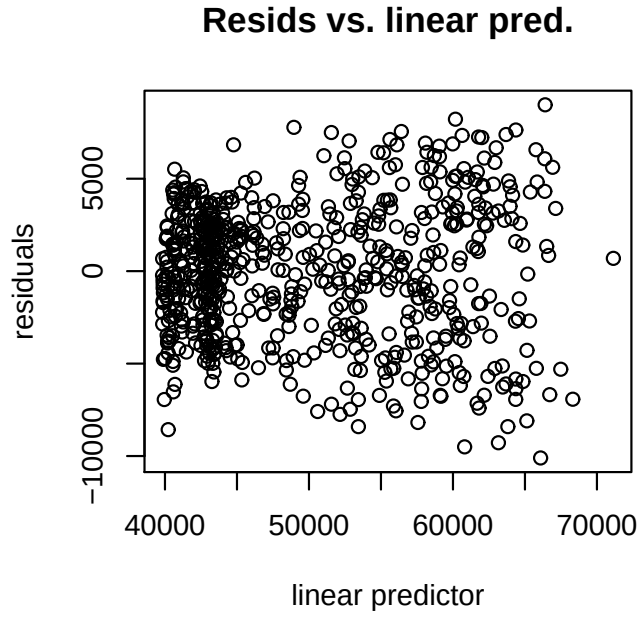
- g est une liste d'éléments divers: effets estimés, données, statistiques de test...L'ensemble des éléments existant est obtenu par l'instruction: names(g)
- pour obtenir les éléments de diagnostique: summary(g)

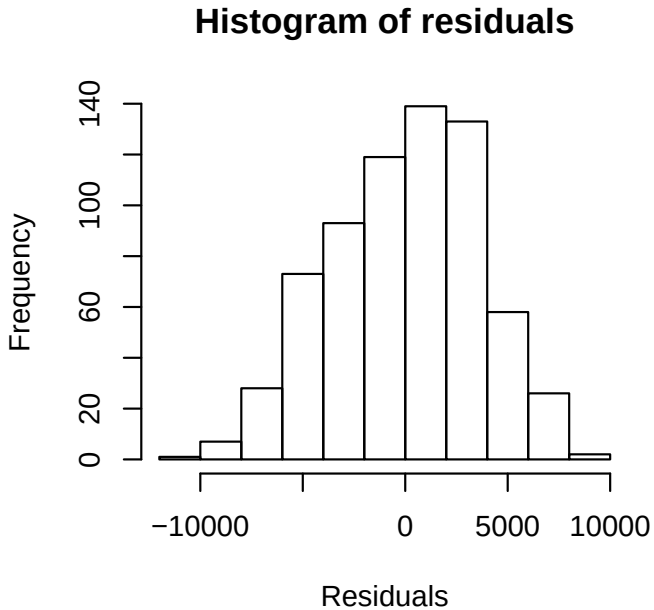
summary(g)

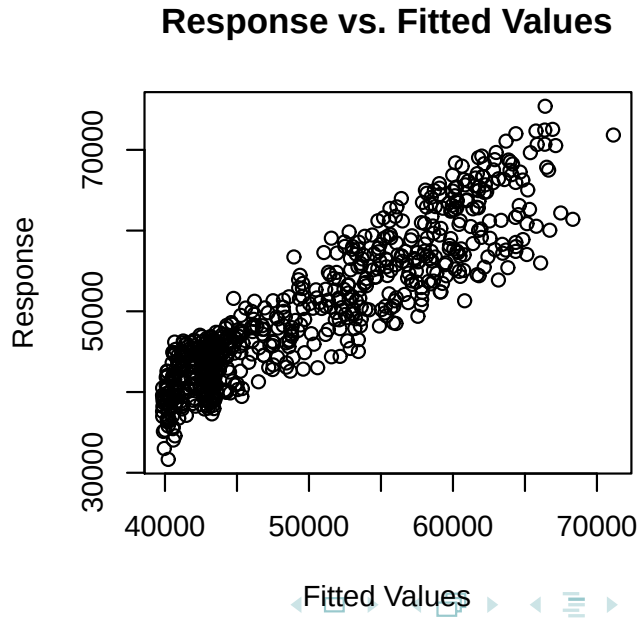
```
> g=gam(Load~ s(NumWeek)+s(Temp),data=data0)
> summary(g)

Family: gaussian
Link function: identity

Formula:
Load ~ s(NumWeek) + s(Temp)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  49832.1      139.4   357.6   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
             edf Ref.df     F p-value
s(NumWeek) 8.589  8.944 14.40  <2e-16 ***
s(Temp)    4.142  5.300 50.98  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.831   Deviance explained = 83.5%
GCV score = 1.346e+07  Scale est. = 1.3188e+07  n = 679
```

Diagnostiques: gam.check(g)









Graphiques

- Graphes des effets non-linéaires:

```
plot(g)
```

- Graphes d'un effet non-linéaire avec intervalle de confiance:

```
plot(g,select=1,shade=TRUE)
```

- Graphe des effets bivariés:

```
vis.gam(g,view=c("x0","x1"),plot.type="persp",box=T
,ticktype="detailed")
```

- Graphe d'un effet superposé aux données:

```
g.terms=predict(g,data,type="terms")
plot(range,logratio)
points(range,g.terms[,1]+attributes(g.terms)$constant,col='red')
```

Prévision

utilisation de la fonction predict.gam:

```
prev=predict(g, data1)
```

Wood, S. (2006), *Generalized Additive Models, An Introduction with R*.