Rappels du premier cours (1)

- Espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$,
- Espace de fonctions de carré intégrable (de dimension infinie) $\mathcal{F} = L^2(T)$, pour $T \subset \mathbb{R}$ intervalle.

Définition

Donnée fonctionnelle *: une réalisation d'une variable fonctionnelle* X*,*

$$\begin{array}{rccc} X : & \Omega & \longrightarrow & L^2(T) \\ & \omega & \longmapsto & X(\omega, \cdot) \end{array}$$

avec

$$\begin{array}{rccc} X(\omega, \cdot) : & T & \longrightarrow & \mathbb{R} \\ & t & \longmapsto & X(\omega, t). \end{array}$$

Exemples : données longitudinales (courbes de températures, taux d'incidence d'une maladie, croissance...), données spectrométriques, données multivariées ou spatiales...

Rappels du premier cours (2)

	Vecteur aléatoire	V.a. fonctionnelle
Données	$X \in \mathbb{R}^d$ $X = {}^t(X_1, \ldots, X_d)$	$X \in L^2(T)$ $X = \{X(t), t \in T\}$
Espace Structure	$(\mathbb{R}^d, \langle \cdot, \cdot \rangle, \| \cdot \|)$ Espace euclidien	$(\mathbb{L}^2(T), \langle \cdot, \cdot \rangle, \| \cdot \|)$ Espace hilbertien
Produit scalaire	$\langle x, y \rangle = \sum_{j=1}^d x_j y_j$	$\langle x, y \rangle = \int_T x(t)y(t)dt$
Norme	$\|x\| = \left(\sum_{j=1}^d x_j^2 \right)^{1/2}$	$\|x\| = \left(\int_T x^2(t)dt \right)^{1/2}$
Bases	Bases orthonormées $(e_j)_{j \in \{1,\ldots,d\}}$	Bases hilbertiennes $(\varphi_j)_{j \geq 1}$
Écriture unique	$x = \sum_{j=1}^d \langle x, e_j \rangle e_j$	$x = \sum_{j \geq 1} \langle x, \varphi_j \rangle \varphi_j.$

Rappels du premier cours (3)

	Vecteur aléatoire	V.a. fonctionnelle
Données	$X \in \mathbb{R}^d$ $X = {}^t(X_1, \ldots, X_d)$	$X \in L^2(T)$ $X = \{X(t), t \in T\}$
Moyenne	vecteur de moyenne $\mathbb{E}[X] = {}^t(\mathbb{E}[X_1], \ldots, \mathbb{E}[X_d])$	courbe de la moyenne $\mathbb{E}[X] = \{\mathbb{E}[X(t)], t \in T\}$
Covariance	matrice $\Sigma_X = (\mathrm{Cov}(X_j, X_k))_{1 \leq j,k \leq d}$	fonction de covariance C_X ou opérateur Γ_X $C_X(s,t) = \mathrm{Cov}(X(s), X(t))$ $\Gamma_X : f \longmapsto \Gamma_X f(\cdot)$ $\Gamma_X f(t) = \mathbb{E}[\langle X, f \rangle X(t)] = \int_T C(s,t) f(s) ds$

Statistique pour données fonctionnelles.
Chapitre 3. Des données aux fonctions (lisses)

Gaëlle Chagny
CNRS, Labo. de Maths. R. Salem, Univ. Rouen,

Université Paris Dauphine – Executive Master Statistique et Big data, 2020

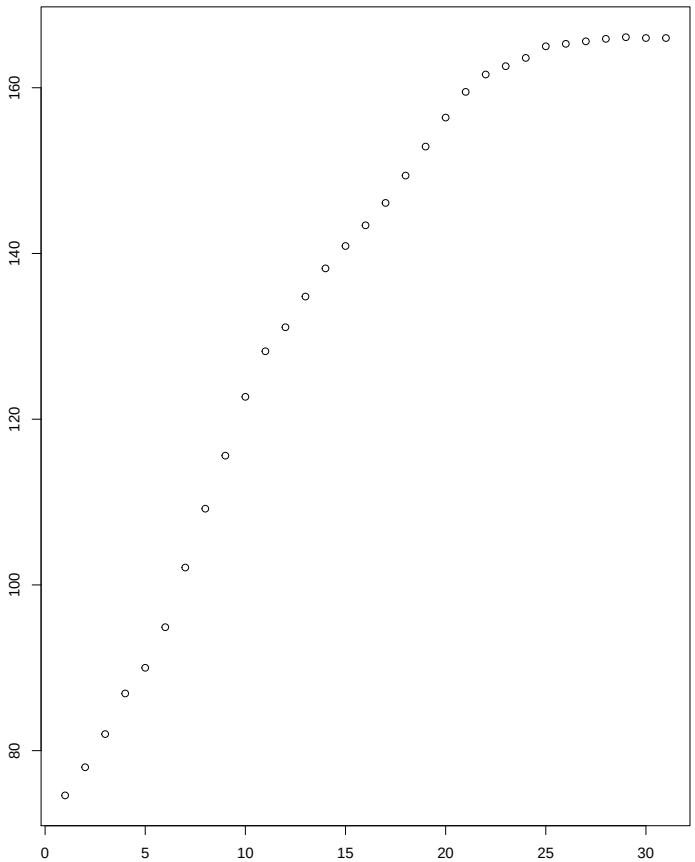
Plan

Plan

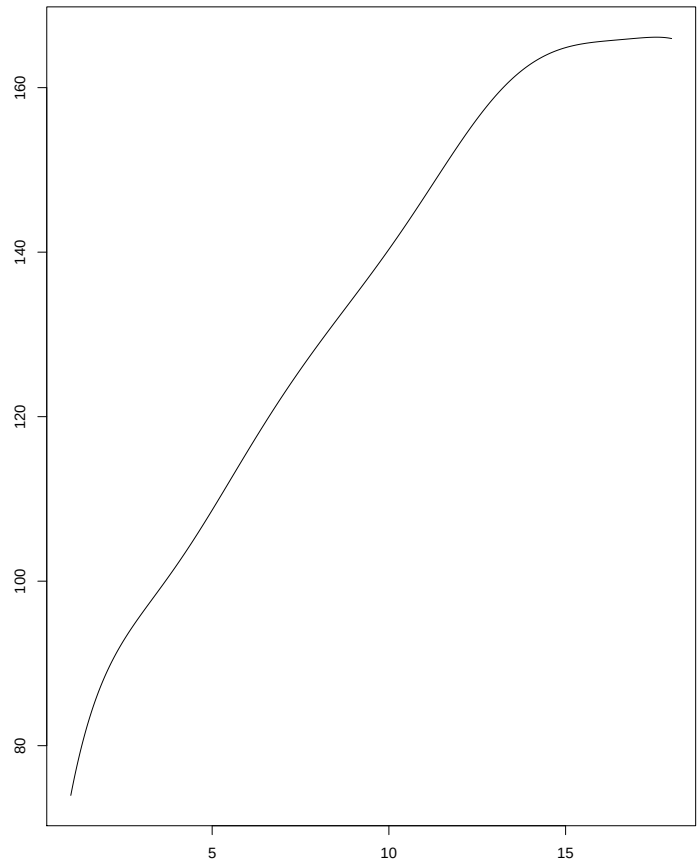
Problématique (1)

Comment passer

des données brutes



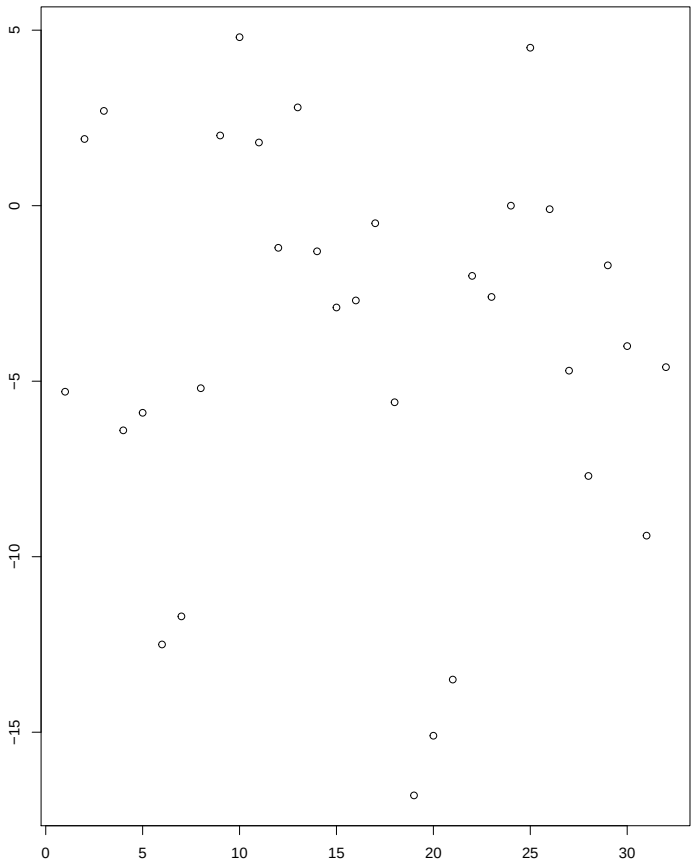
à une courbe lisse ?



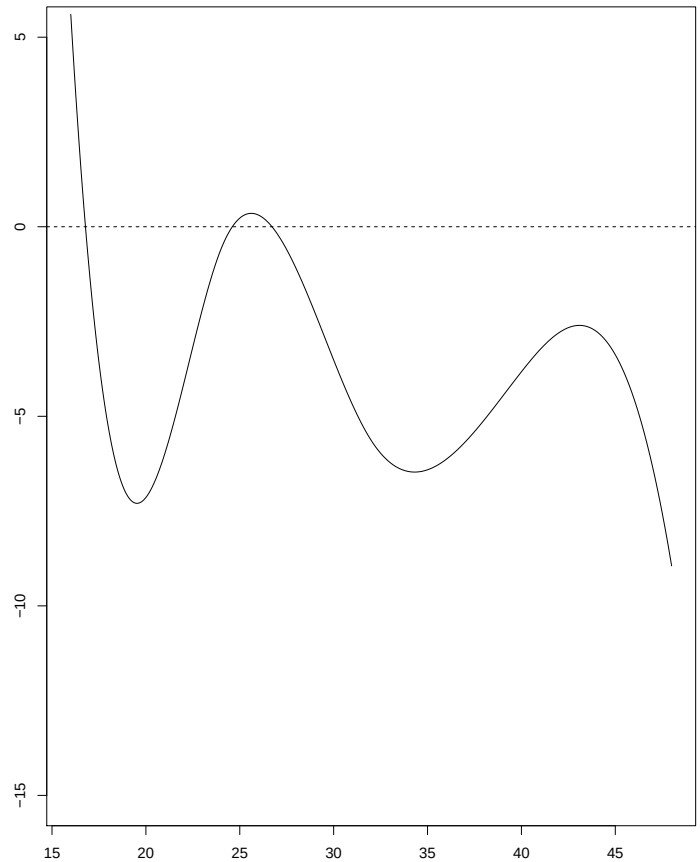
Problématique (2)

Comment passer

des données brutes



à une courbe lisse ?



Problématique (3)

1ère étape de toute étude de données fonctionnelles

Comment passer

- des données brutes sous forme discrétisées

$$\forall i \in \{1, \ldots, n\}, \quad y_i = {}^t(y_{i,1}, \ldots, y_{i,p_i})$$

à

- des données sous leur forme continue

$$\forall i \in \{1, \ldots, n\}, \quad \{x_i(t), \quad t \in T\}$$

$\longrightarrow$ Conversion de mesures discrètes en une fonction continue.

Notation : $y_{i,j}$ $j-$ième mesure de l'individu i.

Problématique (4)

Quelle modélisation ?

- **1er cas.** Données brutes collectées sans erreur (ou avec erreurs négligeables)

$$\forall i \in \{1, \dots, n\}, \quad y_{i,j} = x_i(t_{i,j}) \quad j = 1, \dots, p_i.$$

$\longrightarrow$ interpolation

- **2nd cas.** Données bruitées.

$$\forall i \in \{1, \dots, n\}, \quad y_{i,j} = x_i(t_{i,j}) + \varepsilon_{i,j} \quad j = 1, \dots, p_i,$$

$\varepsilon_{i,j}$ variables non observées centrées, admettant une variance.

$\longrightarrow$ Modèle de régression non-paramétrique. Méthodes de Lissage

Problématique (5)

- **Cadre.** $y = {}^t(y_1, \ldots, y_p) \in \mathbb{R}^p$

$$y_j = x(t_j) + \varepsilon_j, \quad j = 1, \ldots, p.$$

ε_j v.a.r. non observées *i.i.d.* centrées, admettant une variance (inconnue).

- **Objectif.** Construire une approximation/estimation $\widehat{x}$ de x.

- **Exigences possibles (en analyse de données fonctionnelles).**
  - $\widehat{x}(t_j)$ proche de y_j pour tout j
    $\longrightarrow$ lissage par moindres carrés (critère global).
  - $\widehat{x}$ régulière
    $\longrightarrow$ lissage par moindres carrés pénalisés (critère global).
  - $\widehat{x}(t)$ construit en tenant plus compte des y_j tq t_j est proche de t.
    $\longrightarrow$ lissage par méthodes locales

Plan

Lissage par moindres carrés (1)

$$y_j = x(t_j) + \varepsilon_j, \quad j = 1, \ldots, p.$$

Définition

- *critère des moindres carrés pour le problème de régression lié aux données* $(t_j, y_j)_{j=1,\ldots,p}$ *:*

$$Crit_{LS}(x) = \sum_{j=1}^{p} (y_j - x(t_j))^2.$$

- *estimateur de type moindres carrés*

$$\widehat{x} \in \arg\min_x Crit_{LS}(x).$$

Question : sur quel espace de fonctions calculer le minimum ?
$\longrightarrow$ Recherche de $\widehat{x}$ sous la forme d'une C.L. de fonctions linéairement indépendantes.

Lissage par moindres carrés (2)

- Fonctions linéairement indépendantes : $(\varphi_k)_{k=1,\dots,D}$.
- Espace d'approximation, ou modèle ("sieve") : $S_D = \text{Vect}\{\varphi_1, \dots, \varphi_D\}$
- Minimisation du critère sur S_D

$$\begin{aligned}
\min_{x \in S_D} \text{Crit}_{LS}(x) &= \min_{x \in S_D} \sum_{j=1}^{p} (y_j - x(t_j))^2 \\
&= \min_{\theta \in \mathbb{R}^D} \sum_{j=1}^{p} \left(y_j - \sum_{k=1}^{D} \theta_k \varphi_k(t_j) \right)^2 \\
&:= \min_{\theta \in \mathbb{R}^D} \text{Crit}_{LS,\theta}(\theta).
\end{aligned}$$

Lemme

Si $\widehat{x}$ est la solution du problème de minimisation précédent sur S_D,

$$\widehat{x}(t) = \sum_{k=1}^{D} \widehat{\theta}_k \varphi_k(t), \quad t \in T, \quad \textit{avec } \widehat{\theta} = (\theta_k)_{k \in \{1,\dots,D\}} \in \arg\min_{\theta \in \mathbb{R}^D} \textit{Crit}_{LS,\theta}(\theta).$$

Si $\Phi(t) = {}^t(\varphi_1(t), \dots, \varphi_D(t))$, $\widehat{x}(t) = {}^t\widehat{\theta}\Phi(t)$.

Lissage par moindres carrés (3)

$$\min_{\theta\in\mathbb{R}^D} \sum_{j=1}^{p} \left(y_j - \sum_{k=1}^{D} \theta_k \varphi_k(t_j) \right)^2 = \min_{\theta\in\mathbb{R}^D} \mathrm{Crit}_{LS,\theta}(\theta) = \min_{\theta\in\mathbb{R}^D} {}^t(y - \Phi\theta)(y - \Phi\theta)$$

Proposition

La solution $\widehat{\theta}$ au problème d'optimisation satisfait

$$\widehat{\theta} = ({}^t\Phi\Phi)^{-1}\,{}^t\Phi y,$$

- $y = {}^t(y_1, \ldots, y_p)$
- Φ *matrice à* p *lignes et* D *colonnes :* $\Phi = (\varphi_k(t_j))_{\substack{1\le j\le p,\\ 1\le k\le D}}$.

Le vecteur $\widehat{y} = {}^t(\widehat{y}_1, \ldots, \widehat{y}_p) = {}^t(\widehat{x}(t_1), \ldots, \widehat{x}(t_p))$ *des valeurs ajustées est donc*

$$\widehat{y} = \Phi\widehat{\theta} = \Phi({}^t\Phi\Phi)^{-1}\,{}^t\Phi y.$$

Lissage par moindres carrés (4) - Choix de la dimension de l'espace d'approximation

$$\widehat{x} \in \arg\min_{x \in S_D} \mathrm{Crit}_{LS}(x).$$

Question posée par la méthode. Choix de la dimension D de l'espace d'approximation $S_D = \mathrm{Vect}\{\varphi_1, \ldots, \varphi_D\}$.

- Si D est trop petit : biais de l'estimation trop grand, peu de flexibilité
  $\longrightarrow$ données sous-ajustées (*underfitting*)

- Si D est trop grand : bcp de flexibilité, ms variabilité trop importante
  $\longrightarrow$ données sur-ajustées (*overfitting*)

Attention ! Augmenter D ne signifie pas toujours calculer des coefficients supplémentaires, il faut parfois tout calculer à nouveau.
$\longrightarrow$ Cas des espaces non-emboités (base de splines par exemple).

Lissage par moindres carrés (5) - Code R - Méthode 1

Calcul de l'objet fonctionnel $\widehat{x}$ tq $\widehat{x}(t) = \sum_{k=1}^{D} \widehat{\theta}_k \varphi_k(t), \quad t \in T,$ avec $\widehat{\theta} = ({}^t\Phi\Phi)^{-1}{}^t\Phi y,$

1. **Point de départ**
   - Matrice Φ (voir chapitre précédent)
   - Vecteur des observations y

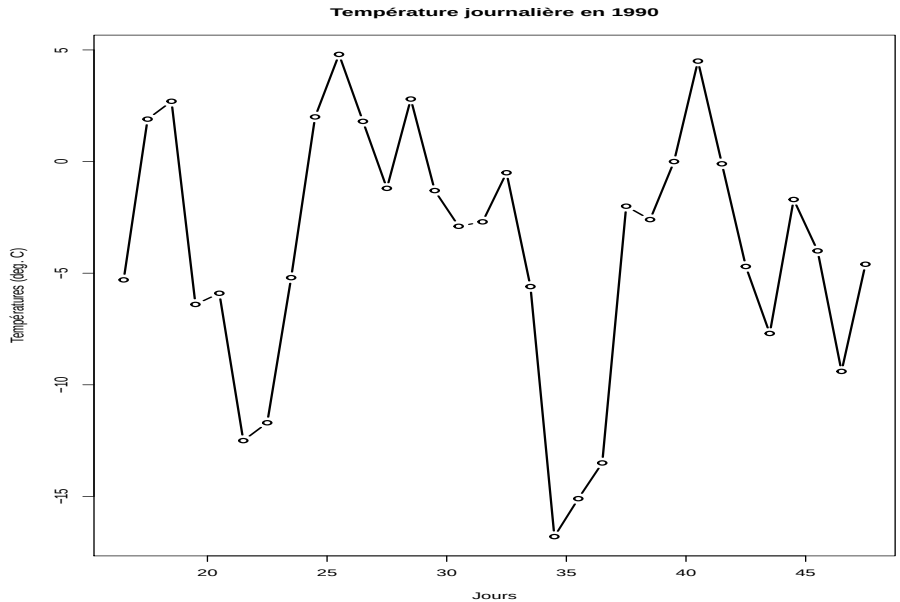
2. **Fonctions utiles**
   - `crossprod`
     - Cas 1 : seul argument, $\Phi \longrightarrow$ renvoie $A = {}^t\Phi\Phi$.
     - Cas 2 : arguments Φ et $y \longrightarrow$ renvoie $b = {}^t\Phi y$.
   - `solve` : arguments A et $b \longrightarrow$ renvoie $\widehat{\theta}$ obtenu par résolution du système $A\widehat{\theta} = b$.

3. **Création de $\widehat{x}$** : fonction `fd` (voir chapitre précédent).

4. **Comparaison de $\widehat{x}$ aux données y** : fonction `plotfit.fd`.

Lissage par moindres carrés (6) - Exemple 1

Lissage des données `MontrealTemp` (package `fda`).



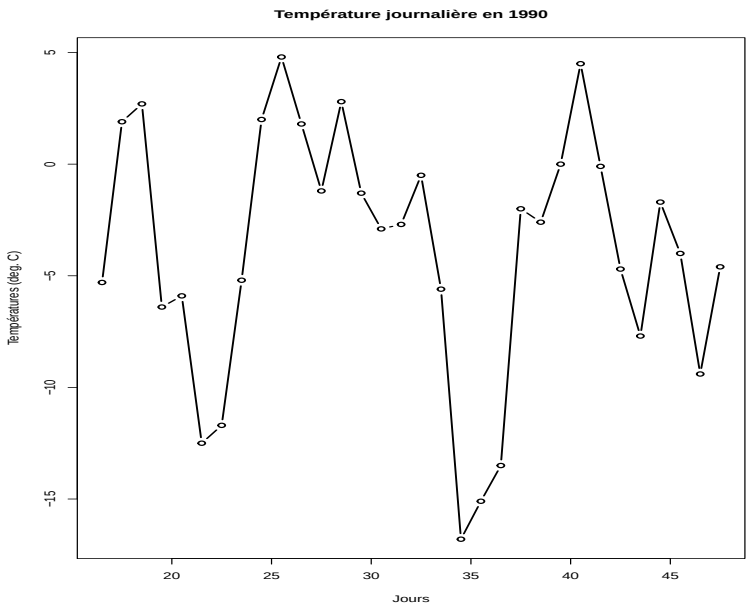
Code R.

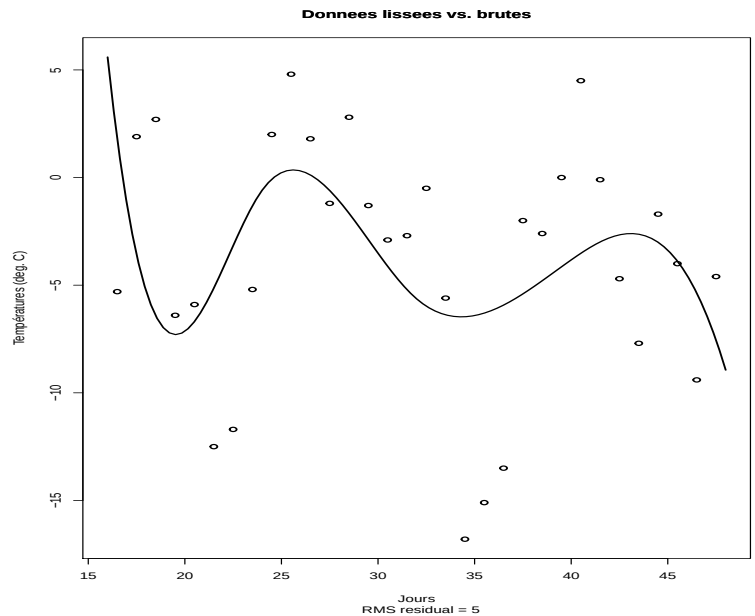
```
Donnees_brutes_temp<- t(MontrealTemp[, 16:47]) #matrice 32*34
Jours <-((16:47)+0.5)
plot(Jours,t(Donnees_brutes_temp[,30]),"b", lwd=2,xlab="Jours",ylab='Températures (deg. C)',main="Température journalière en 1990")
```

–> Étude de l'influence du nombre de fonctions de base (choix de D).

Lissage par moindres carrés (7) - Exemple 1

Lissage des données MontrealTemp - Cas d'une base de 7 fonctions .



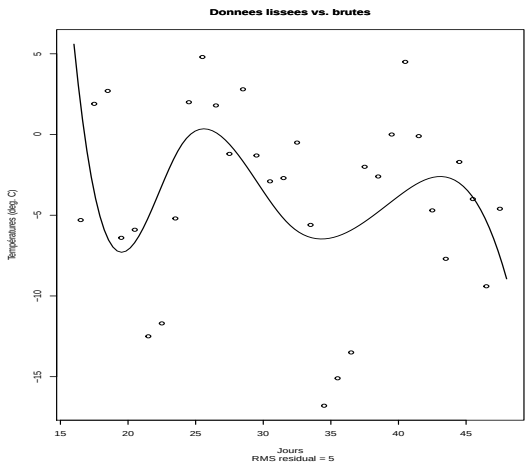


Code R.

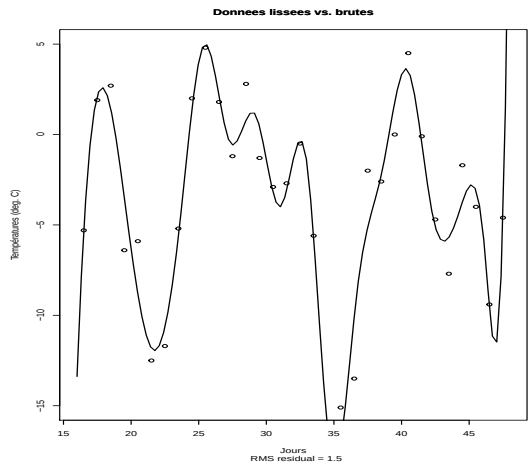
```
Fct_base_temp <- create.bspline.basis(c(16,48),7) #7 Bsplines cubiques
MatPhi_temp <- eval.basis(Jours,Fct_base_temp)
Coeff_MC_temp <- solve(crossprod(MatPhi_temp),crossprod(MatPhi_temp,Donnees_brutes_temp)) #coeff du lissage
Donnees_lisseesMC_temp<-fd(Coeff_MC_temp,Fct_base_temp,list("Jours","années","Température (deg. C)"))
plotfit.fd(Donnees_brutes_temp[,30],Jours,Donnees_lisseesMC_temp[30],lty=1,lwd=2,main='Donnees lissees vs. brutes',
              xlab="Jours",ylab='Températures (deg. C)')
```

Lissage par moindres carrés (8) - Exemple 1

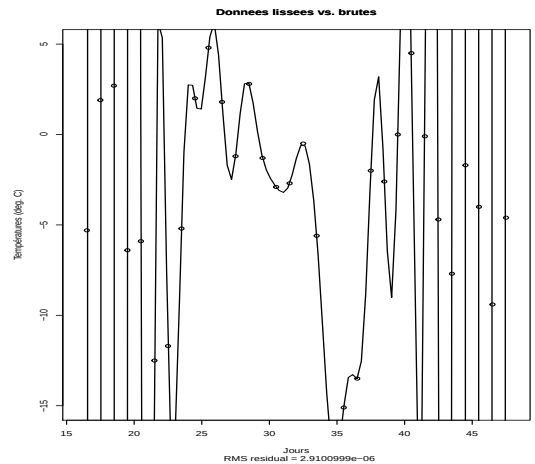
Lissage des données `MontrealTemp` - Choix du nombre de fonctions de base.



$D = 7$



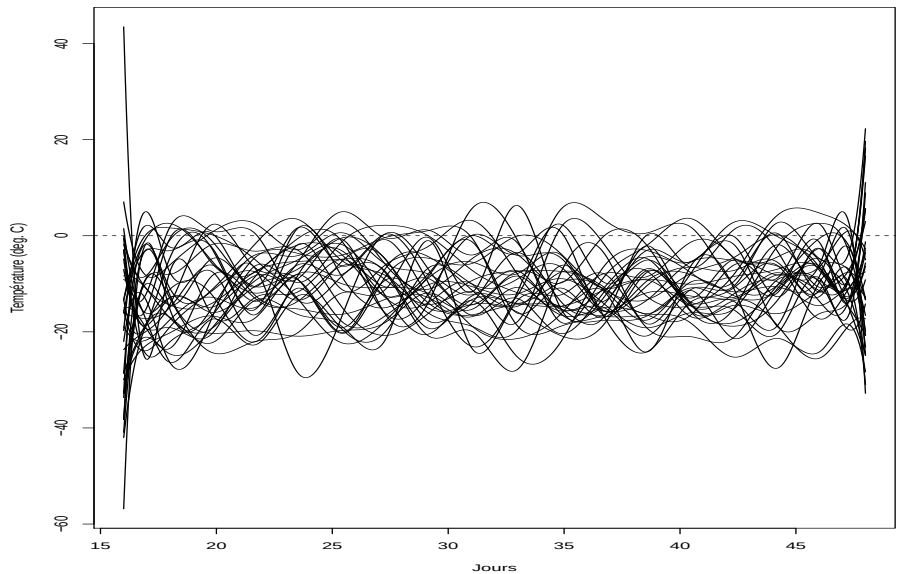
$D = 20$



$D = 32$

Lissage par moindres carrés (9) - Exemple 1

Lissage des données `MontrealTemp` (package `fda`).

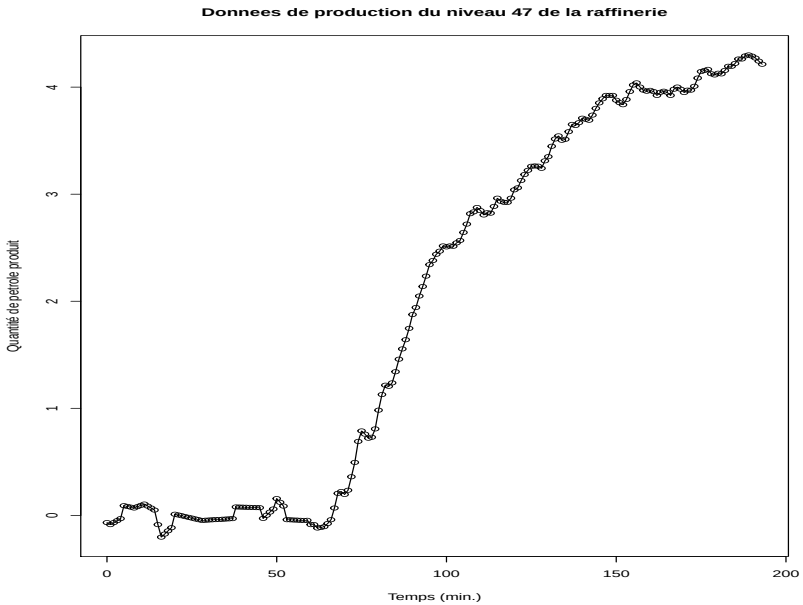


Code R.

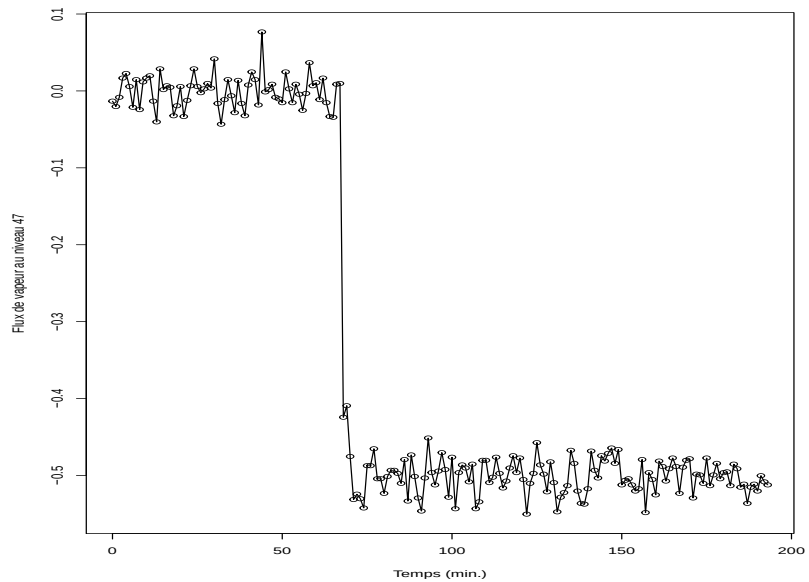
```
plot(Donnees_lisseesMC_temp,lty=1,col=1)
```

Lissage par moindres carrés (10) - Exemple 2

Lissage des données `refinery` (package `fda`).



quantité de pétrole produit
à un certain niveau de la colonne de distillation
en fonction du temps

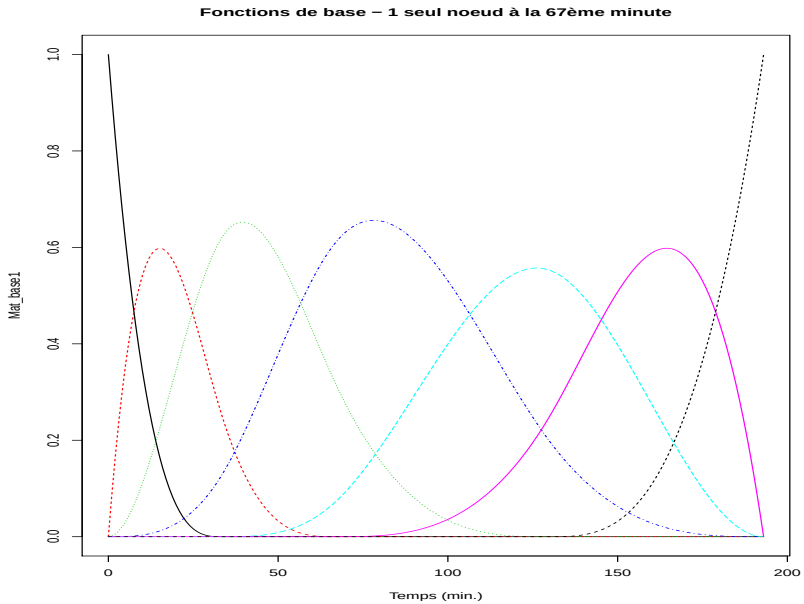


flux de vapeur
dans cette même colonne
en fonction du temps

–> Étude de l'influence du nombre de noeuds d'une base de $B-$splines.

Lissage par moindres carrés (11) - Exemple 2

Lissage des données `refinery` - Base de $B-$splines cubiques, 1 seul noeud à la 67ème minute.

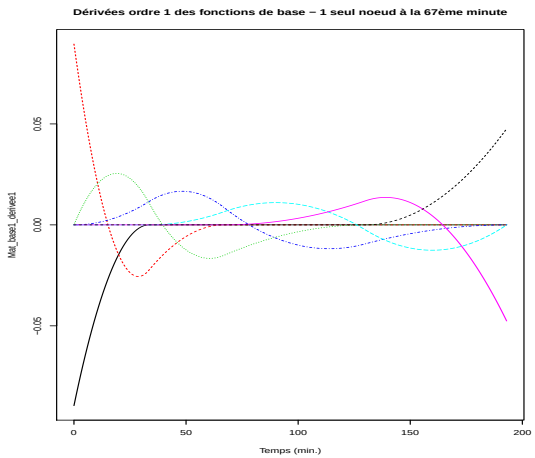


Code R.

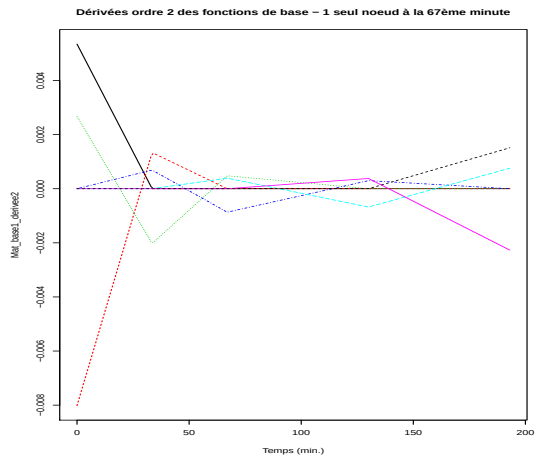
```
Noeuds_Refin1<-c(0,33.5,67,130,193);
Fct_base_Refin1 <- create.bspline.basis(c(0,193),norder=4,breaks=Noeuds_Refin1)
temps <-seq(0,193,1)
Mat_base1 <-eval.basis(refinery$Time,Fct_base_Refin1,Lfdobj = 0)
matplot(refinery$Time,Mat_base1,xlab='Temps (min.)',type='l', main='Fonctions de base
- 1 seul noeud à la 67ème minute')
```

Lissage par moindres carrés (12) - Exemple 2

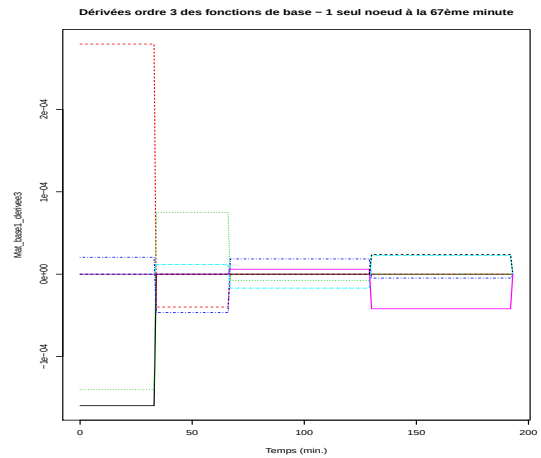
Lissage des données `refinery` - Base de $B-$splines cubiques, 1 seul noeud à la 67ème minute.



Dérivées ordre 1



Dérivées ordre 2



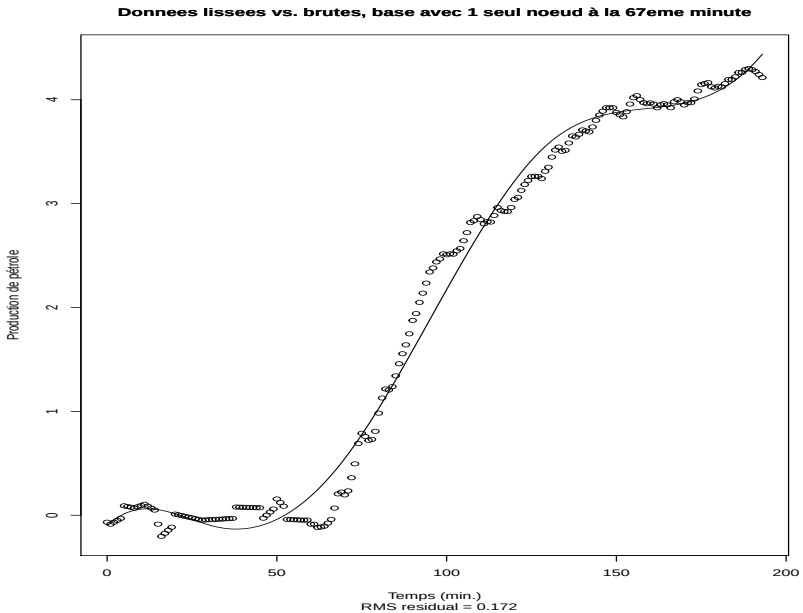
Dérivées ordre 3

Code R.

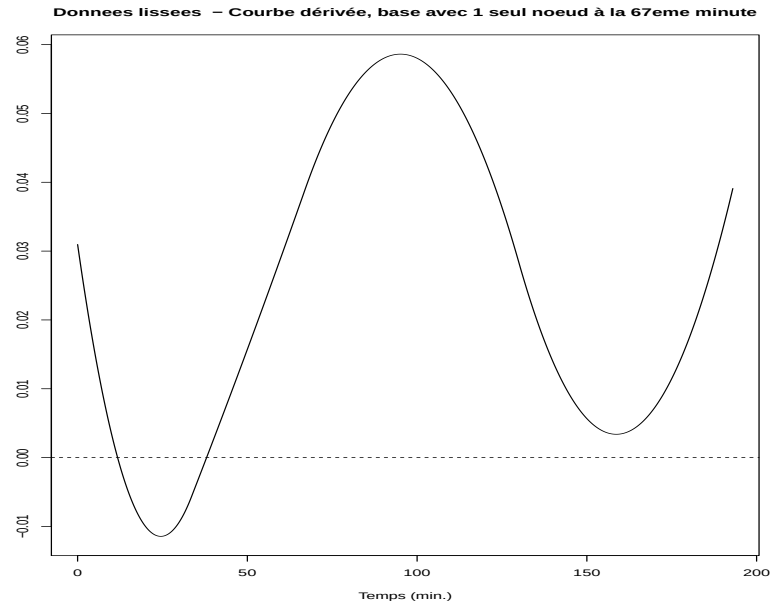
```
Mat_base1_derivee1 <-eval.basis(refinery$Time,Fct_base_Refin1,Lfdobj = 1)
Mat_base1_derivee2 <-eval.basis(refinery$Time,Fct_base_Refin1,Lfdobj = 2)
Mat_base1_derivee3 <-eval.basis(refinery$Time,Fct_base_Refin1,Lfdobj = 3)
matplot(refinery$Time,Mat_base1_derivee1,xlab='Temps (min.)',type='l',main='Dérivées ordre 1 des fonctions de base
- 1 seul noeud à la 67ème minute')
matplot(refinery$Time,Mat_base1_derivee2,xlab='Temps (min.)',type='l',main='Dérivées ordre 2 des fonctions de base
 - 1 seul noeud à la 67ème minute')
matplot(refinery$Time,Mat_base1_derivee3,xlab='Temps (min.)',type='l',main='Dérivées ordre 3 des fonctions de base
- 1 seul noeud à la 67ème minute')
```

Lissage par moindres carrés (13) - Exemple 2

Lissage des données `refinery` - Base de $B-$splines cubiques, 1 seul noeud à la 67ème minute .



Données lissées



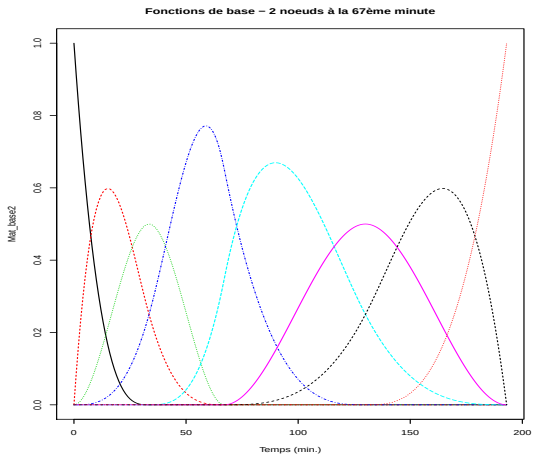
Courbe dérivée

Code R.

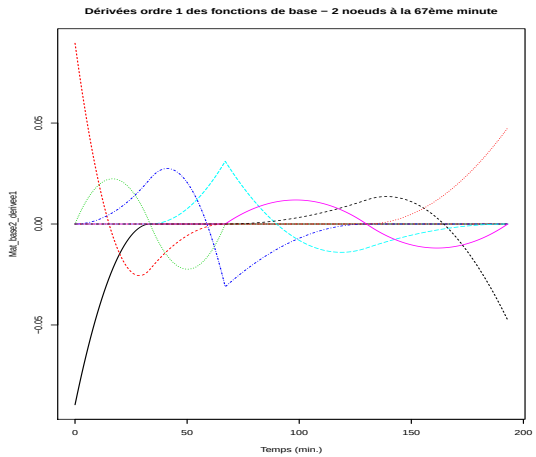
```
Coeff_MC_Refin1<- solve(crossprod(Mat_base1),crossprod(Mat_base1,refinery$Tray47))
Donnees_lisseesMC_Refin1 <-fd(Coeff_MC_Refin1,Fct_base_Refin1)

plotfit.fd(refinery$Tray47,refinery$Time,Donnees_lisseesMC_Refin1,lty=1,lwd=1,
main='Donnees lissees vs. brutes, base avec 1 seul noeud à la 67eme minute',col=1,xlab='Temps (min.)',ylab='Production de pétrole')
plot(Donnees_lisseesMC_Refin1,Lfdobj=1,xlab='Temps (min.)',ylab='',
main='Donnees lissees  - Courbe dérivée, base avec 1 seul noeud à la 67eme minute',col=1)
```

Lissage par moindres carrés (14) - Exemple 2

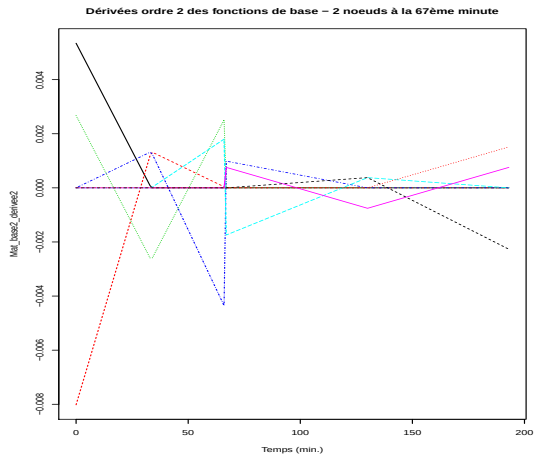
Lissage des données `refinery` - Base de $B-$splines cubiques, 2 noeuds à la 67ème minute.



Fonctions de base



Dérivées ordre 1



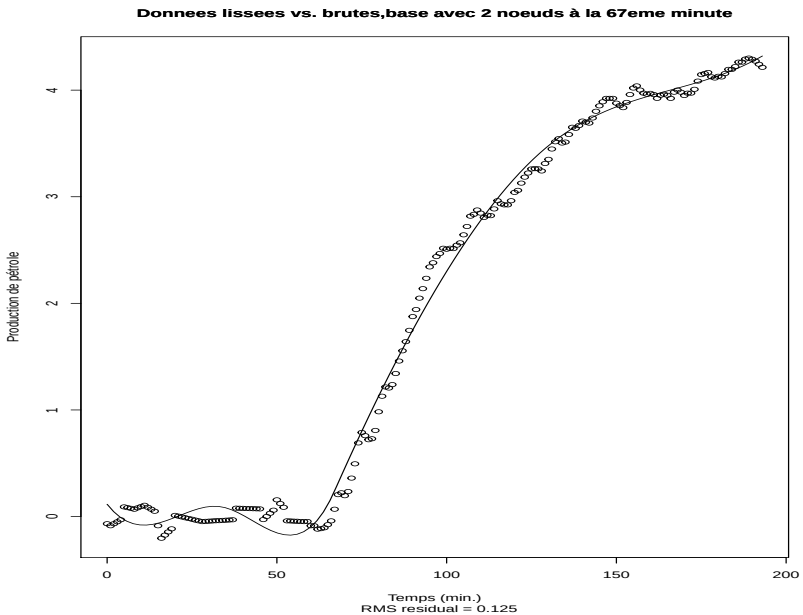
Dérivées ordre 2

Code R.

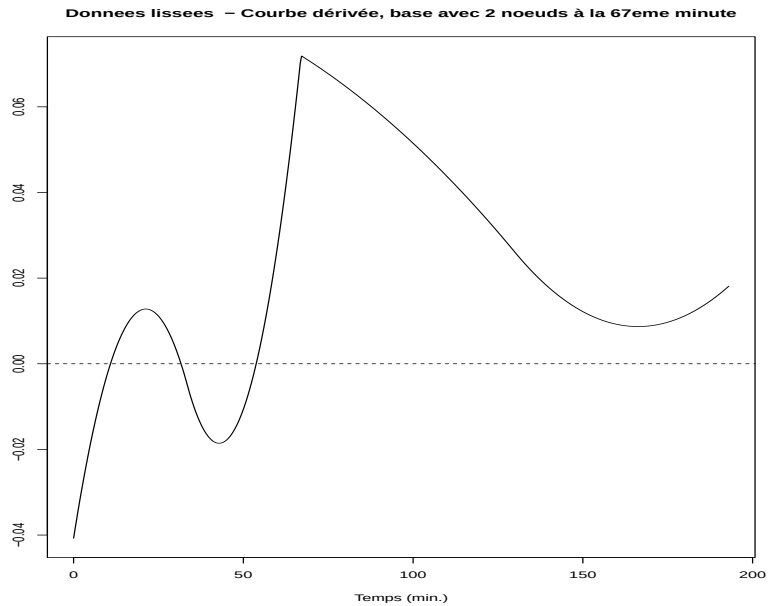
```
Noeuds_Refin2<-c(0,33.5,67,67,130,193);
Fct_base_Refin2 <- create.bspline.basis(c(0,193),norder=4,breaks=Noeuds_Refin2)
temps <-seq(0,193,1)
Mat_base2 <-eval.basis(refinery$Time,Fct_base_Refin2,Lfdobj = 0)
Mat_base2_derivee1 <-eval.basis(refinery$Time,Fct_base_Refin2,Lfdobj = 1)
Mat_base2_derivee2 <-eval.basis(refinery$Time,Fct_base_Refin2,Lfdobj = 2)

matplot(refinery$Time,Mat_base2,xlab='Temps (min.)',type='l',
main='Fonctions de base - 2 noeuds à la 67ème minute')
matplot(refinery$Time,Mat_base2_derivee1,xlab='Temps (min.)',type='l',
main='Dérivées ordre 1 des fonctions de base - 2 noeuds à la 67ème minute')
matplot(refinery$Time,Mat_base2_derivee2,xlab='Temps (min.)',type='l',
main='Dérivées ordre 2 des fonctions de base - 2 noeuds à la 67ème minute')
```

Lissage par moindres carrés (15) - Exemple 2

Lissage des données `refinery` - Base de $B-$splines cubiques, 2 noeuds à la 67ème minute .



Données lissées



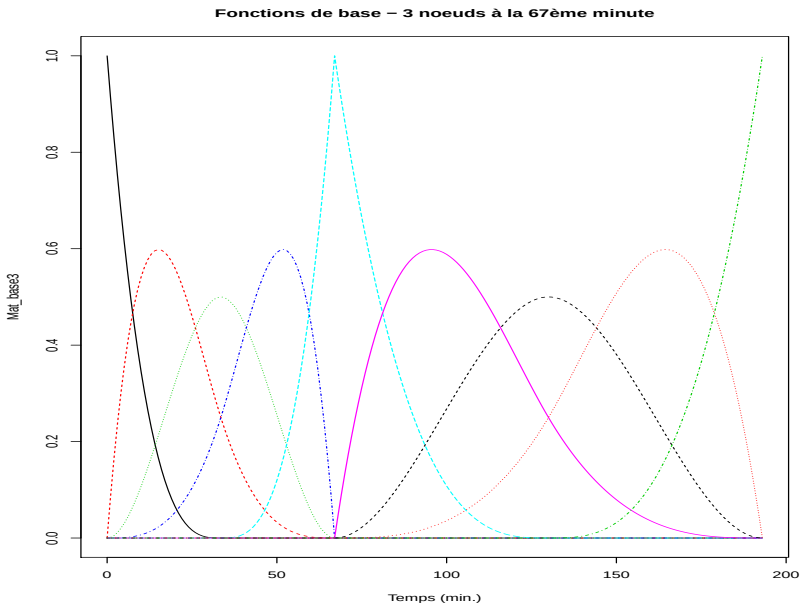
Courbe dérivée

Code R.

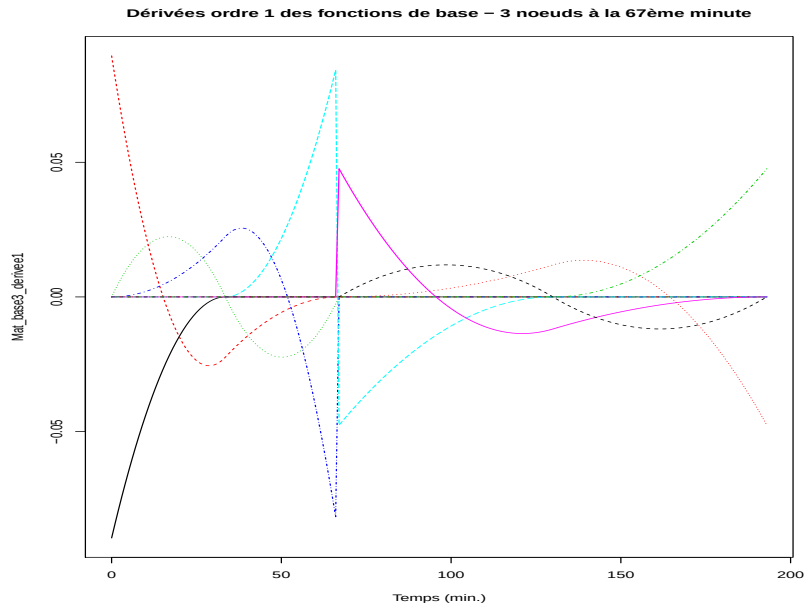
```
Coeff_MC_Refin2<- solve(crossprod(Mat_base2),crossprod(Mat_base2,refinery$Tray47))
Donnees_lisseesMC_Refin2 <-fd(Coeff_MC_Refin2,Fct_base_Refin2)

plotfit.fd(refinery$Tray47,refinery$Time,Donnees_lisseesMC_Refin2,lty=1,lwd=1,
main='Donnees lissees vs. brutes,base avec 2 noeuds à la 67eme minute',col=1,xlab='Temps (min.)',ylab='Production de pétrole')
plot(Donnees_lisseesMC_Refin2,Lfdobj=1,xlab='Temps (min.)',ylab='',
main='Donnees lissees  - Courbe dérivée, base avec 2 noeuds à la 67eme minute',col=1)
```

Lissage par moindres carrés (16) - Exemple 2

Lissage des données `refinery` - Base de $B-$splines cubiques, 3 noeuds à la 67ème minute.



Fonctions de base



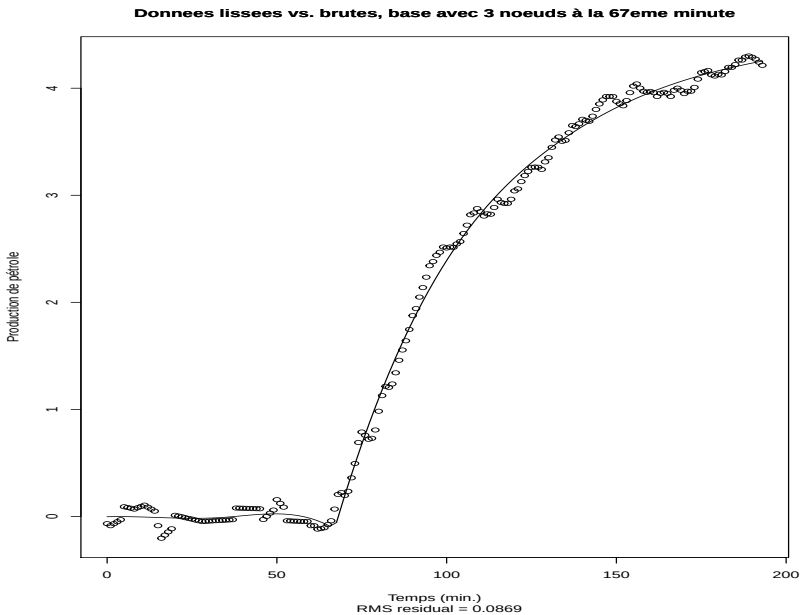
Dérivées ordre 1

Code R.

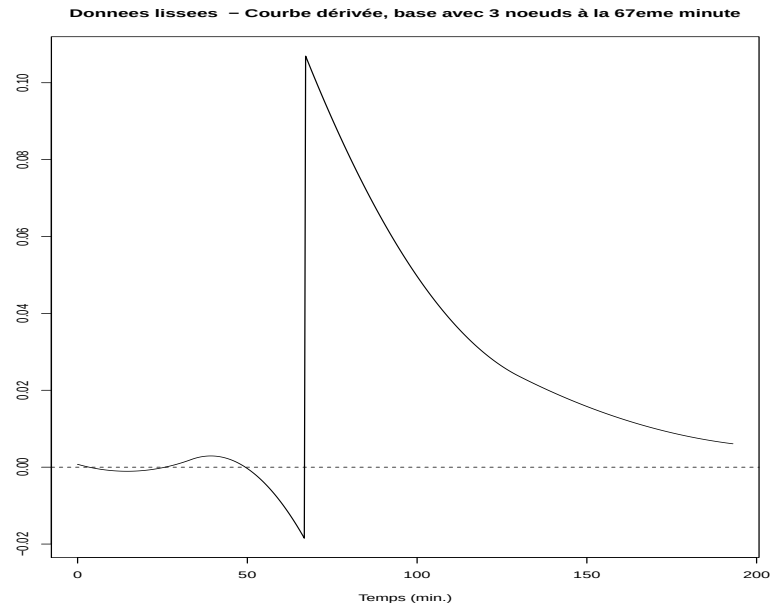
```
Noeuds_Refin3<-c(0,33.5,67,67,67,130,193);
Fct_base_Refin3 <- create.bspline.basis(c(0,193),norder=4,breaks=Noeuds_Refin3)
temps <-seq(0,193,1)
Mat_base3 <-eval.basis(refinery$Time,Fct_base_Refin3,Lfdobj = 0)
Mat_base3_derivee1 <-eval.basis(refinery$Time,Fct_base_Refin3,Lfdobj = 1)

matplot(refinery$Time,Mat_base3,xlab='Temps (min.)',type='l',
    main='Fonctions de base - 3 noeuds à la 67ème minute')
matplot(refinery$Time,Mat_base3_derivee1,xlab='Temps (min.)',type='l',
    main='Dérivées ordre 1 des fonctions de base - 3 noeuds à la 67ème minute')
```

Lissage par moindres carrés (17) - Exemple 2

Lissage des données `refinery` - Base de $B-$splines cubiques, 3 noeuds à la 67ème minute.



Données lissées



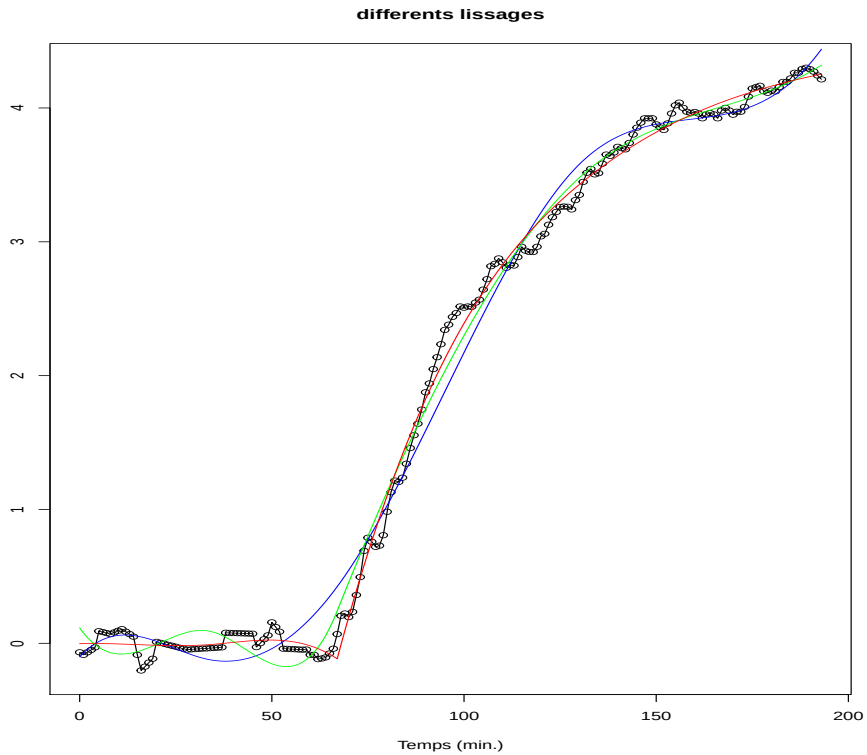
Courbe dérivée

Code R.

```
Coeff_MC_Refin3<- solve(crossprod(Mat_base3),crossprod(Mat_base3,refinery$Tray47))
Donnees_lisseesMC_Refin3 <-fd(Coeff_MC_Refin3,Fct_base_Refin3)

plotfit.fd(refinery$Tray47,refinery$Time,Donnees_lisseesMC_Refin3,lty=1,lwd=1,
main='Donnees lissees vs. brutes, base avec 3 noeuds à la 67eme minute',col=1,xlab='Temps (min.)',ylab='Production de pétrole')
plot(Donnees_lisseesMC_Refin3,Lfdobj=1,xlab='Temps (min.)',ylab='',
main='Donnees lissees  - Courbe dérivée, base avec 3 noeuds à la 67eme minute',col=1)
```

Lissage par moindres carrés (18) - Exemple 2

Lissage des données `refinery` - Synthèse.



Code R.

```
Refin1<-eval.fd(refinery$Time,Donnees_lisseesMC_Refin1,Lfdobj = 0)
Refin2<-eval.fd(refinery$Time,Donnees_lisseesMC_Refin2,Lfdobj = 0)
Refin3<-eval.fd(refinery$Time,Donnees_lisseesMC_Refin3,Lfdobj = 0)

plot(refinery$Time,refinery$Tray47,type='o',xlab='Temps (min.)',ylab='',main='differents lissages')

points(refinery$Time,Refin1,type='l',col="blue")
points(refinery$Time,Refin2,type='l',col="green")
points(refinery$Time,Refin3,type='l',col="red")
```

Lissage par moindres carrés (19) - Code R - Méthode 2

Autre méthode pour le calcul de l'objet fonctionnel $\widehat{x}$ tq

$$\widehat{x}(t) = \sum_{k=1}^{D} \widehat{\theta}_k \varphi_k(t), \quad t \in T, \quad \textbf{avec } \widehat{\theta} = ({}^t\Phi\Phi)^{-1}\,{}^t\Phi y,$$

1. **Point de départ**
   - Matrice Φ (voir chapitre précédent)
   - Vecteur des observations y

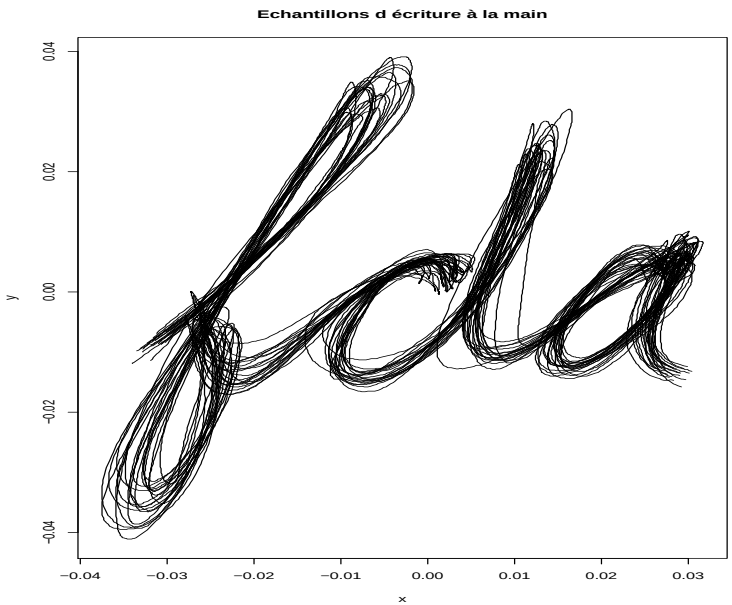
2. **Fonction `smooth.basis`** (à la place de `crossprod` et `solve`)
   - Arguments importants

     `argvals` les valeurs des $(t_j)_{j=1,\ldots,p}$;
     `y` les données $(y_j)_{j=1,\ldots,p}$.
     `fdParObj` objet de la classe `basisfd` (sortie de `create.Nom_base.basis`) dans le cas d'un lissage par MC (non pénalisés)
   - Sortie : si `a<-smooth.basis(...)`
     - `a$fd` objet fonctionnel

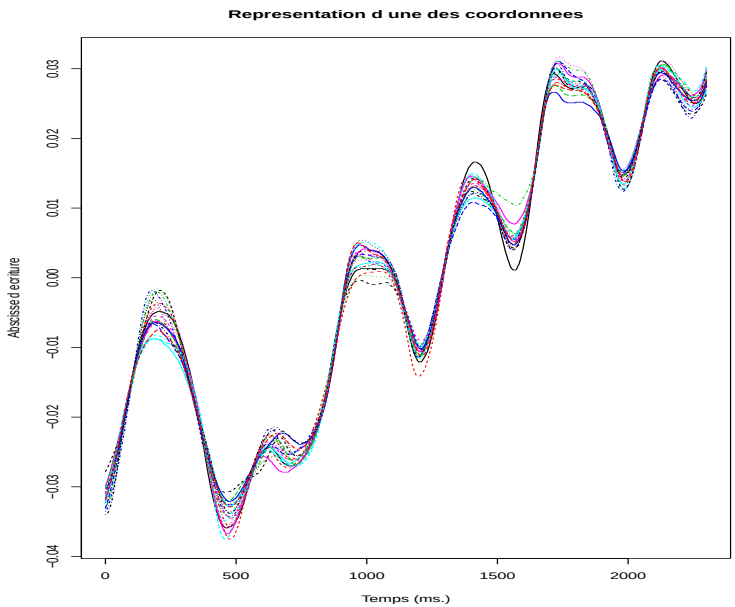
3. **Comparaison de $\widehat{x}$ aux données y** : fonction `plotfit.fd`.

Lissage par moindres carrés (20) - Exemple 3

Lissage des données `handwrit`.



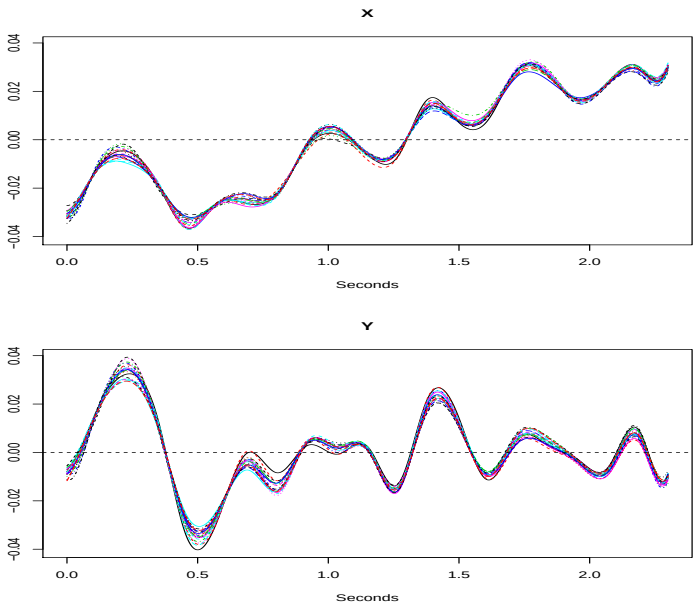
20 tracés
(unité de l'axe des abscisses : cm.)



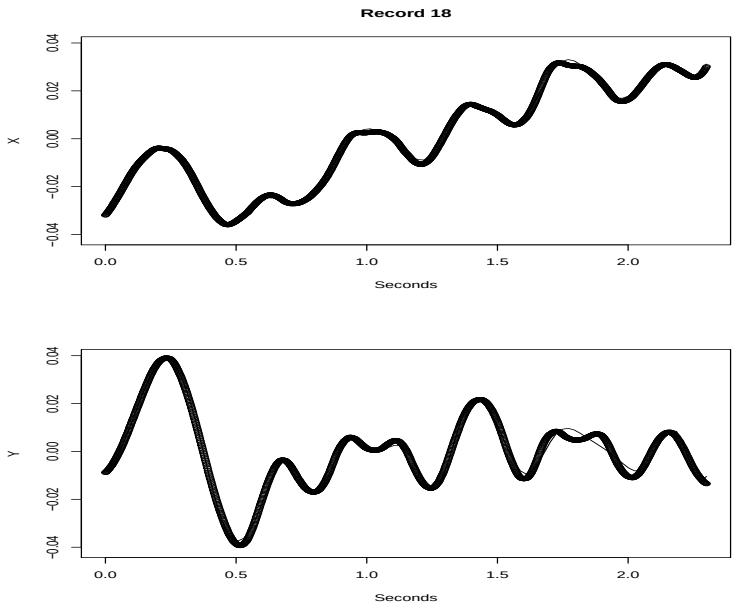
courbes d'évolution des
abscisses au cours du temps (20 courbes)

Lissage par moindres carrés (21) - Exemple 3

Lissage des données `handwrit`. base de B-splines cubiques, avec 21 noeuds (23 fonctions de base)



toutes les courbes lissées



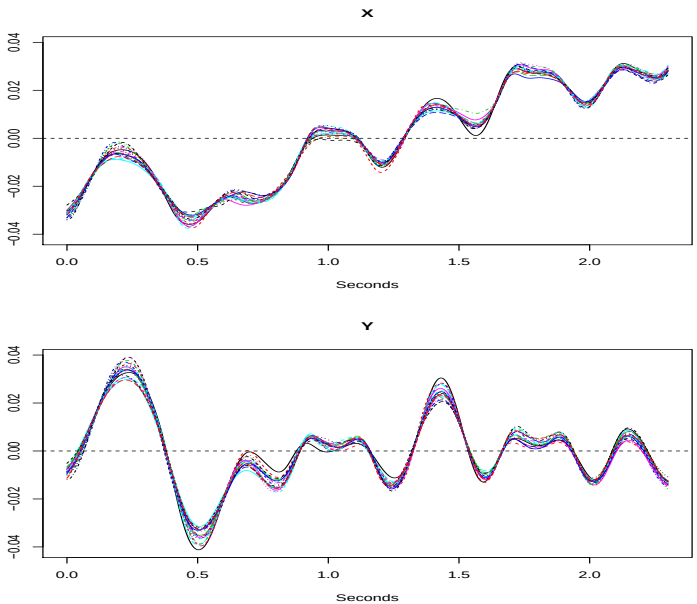
enregistrement 18
données brutes vs. lissées

Code R.

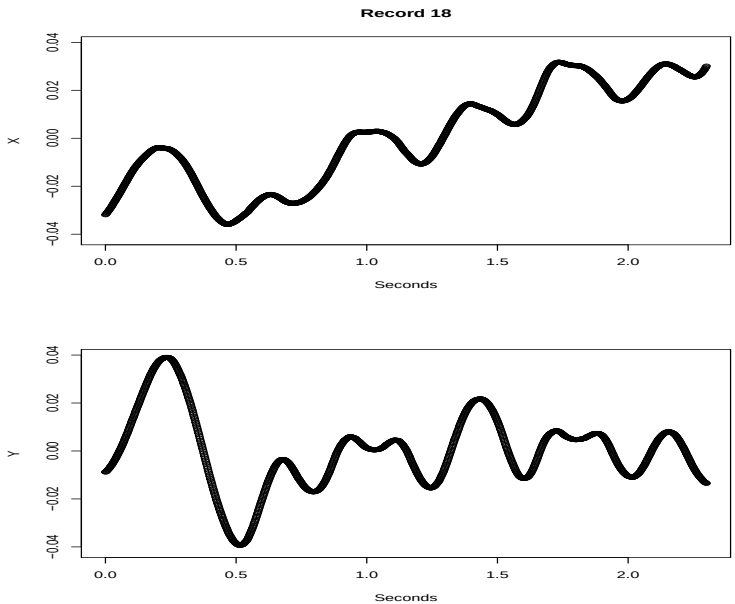
```
pts_rupture_handwrit1 <- seq(0,2.3,length.out=21)
nb_Fct_base_handwrit1 <- 23
Fct_base_handwrit1 <- create.bspline.basis(c(0,2.3),norder=4,breaks=pts_rupture_handwrit1)
tps_handwrit  <- seq(0, 2.3, len=1401)
noms_variables <- list('Seconds',NULL,c('X','Y'))
Donnees_lissees_handwrit1 <- smooth.basis(tps_handwrit,handwrit,Fct_base_handwrit1,fdnames=noms_variables)
par(mfrow=c(2,1))
plot(Donnees_lissees_handwrit1$fd)
plotfit.fd(handwrit,tps_handwrit,Donnees_lissees_handwrit1$fd,type='p')
```

Lissage par moindres carrés (22) - Exemple 3

Lissage des données `handwrit`. base de B-splines d'ordre 6, avec 51 noeuds (55 fonctions de base)



toutes les courbes lissées



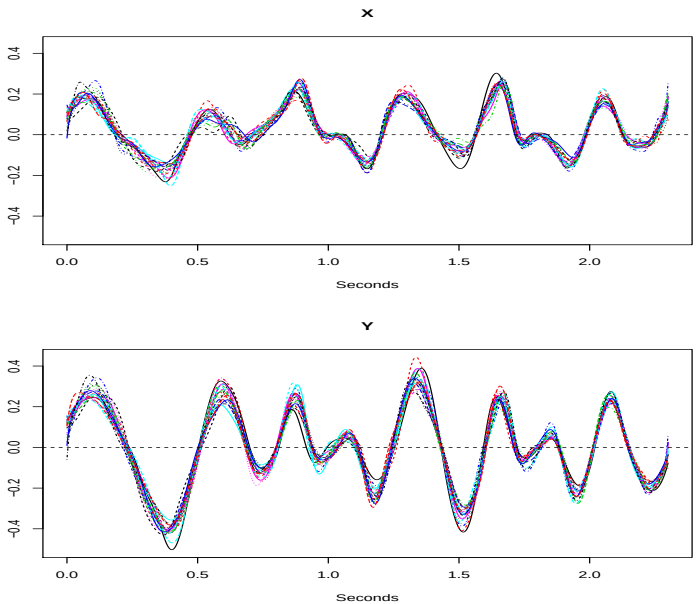
enregistrement 18
données brutes vs. lissees

Code R.

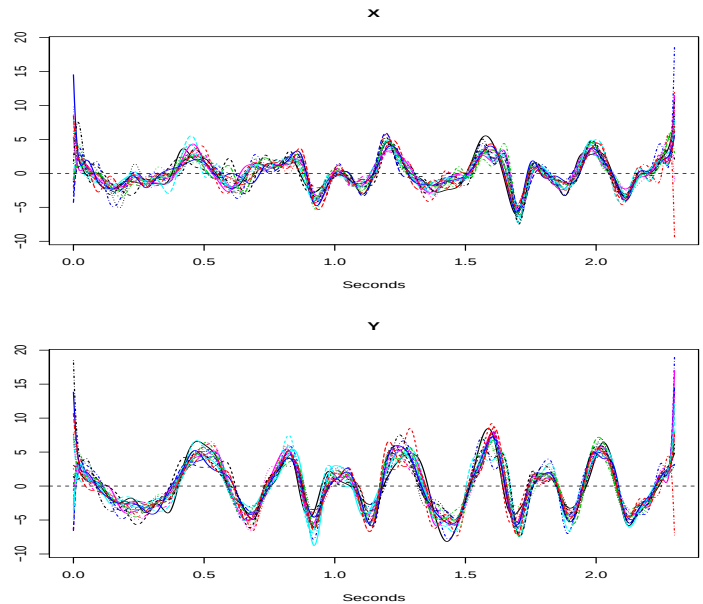
```
pts_rupture_handwrit2 <- seq(0,2.3,length.out=51)
Fct_base_handwrit2 <- create.bspline.basis(c(0,2.3),norder=6,breaks=pts_rupture_handwrit2)
Donnees_lissees_handwrit2 <- smooth.basis(tps_handwrit,handwrit,Fct_base_handwrit2,fdnames=noms_variables)
```

Lissage par moindres carrés (23) - Exemple 3

Lissage des données `handwrit`. base de B-splines d'ordre 6, avec 51 noeuds (55 fonctions de base), courbes dérivées



ordre 1



ordre 2

Code R.

```
par(mfrow=c(2,1))
plot(Donnees_lissees_handwrit2$fd,Lfdobj=1)
plot(Donnees_lissees_handwrit2$fd,Lfdobj=2)
```

Plan

Lissage par moindres carrés pénalisés (1)

$$y_j = x(t_j) + \varepsilon_j, \quad j = 1, \ldots, p.$$

- **Objectif du lissage :**
  - bon ajustement aux données ;
  - pas d'ajustement trop précis (sinon, trop de variabilité)
- **Conséquence :** $\longrightarrow$ minimisation d'un critère de type moindres carrés, mais sous la contrainte que la fonction à estimer est régulière.

- **Mesure de la régularité/rugosité d'une fonction** (*roughness*)

$$\text{pen}_m(x) = \int_T \left(x^{(m)}(t)\right)^2 dt.$$

  - $x^{(1)} = x'$: mesure de la pente de x à chaque instant,
  - $x^{(2)} = x''$: mesure de la courbure de x en chaque point

  $\longrightarrow \int_T (x''(t))^2 \, dt$: énergie.

- **Autres mesures possibles de la régularité** : opérateur d'accélération harmonique - données périodiques de période $2\pi/\omega$.

$$\text{pen}(x) = \int_T (Lx)^2(t)dt, \quad Lx = x^{(3)} + (2\pi/\omega)^2 x'.$$

Lissage par moindres carrés pénalisés (2)

$$y_j = x(t_j) + \varepsilon_j, \quad j = 1, \ldots, p.$$

Définition

- *critère des moindres carrés pénalisé pour le problème de régression lié aux données* $(t_j, y_j)_{j=1,\ldots,p}$*, avec pénalité de lissage :*

$$Crit_{PenLS}(x) = \sum_{j=1}^{p} (y_j - x(t_j))^2 + \lambda pen_m(x).$$

  $\lambda > 0$ *paramètre à calibrer.*

- *estimateur de type moindres carrés pénalisés*

$$\widehat{x} \in \arg\min_x Crit_{PenLS}(x).$$

Question : sur quel espace de fonctions calculer le minimum ? $\longrightarrow$ sur un espace qui ne comporte que des fonctions m fois dérivables de dérivées m-ièmes intégrables sur T.

Lissage par moindres carrés pénalisés (3)

$$y_j = x(t_j) + \varepsilon_j, \quad j = 1, \ldots, p.$$

Notation. $\mathcal{S}^m(T) = \left\{ x : T \to \mathbb{R}, \quad x^{(m)} \text{ existe et est absolument continue} \right\}$

Théorème (Green et Silverman, 1994)

Soit $\widehat{x} \in \arg\min_{x \in \mathcal{S}^m(T)} Crit_{PenLS}(x)$. Alors $\widehat{x}$ est unique, et définie sur $T = [a, b]$ de la façon suivante :

(i) *la restriction de $\widehat{x}$ aux intervalles $[a, t_1]$ et $[t_p, b]$ est un polynôme de degré au plus $m - 1$;*

(ii) *la restriction de $\widehat{x}$ aux intervalles $[t_j, t_{j+1}]$, $j = 1, \ldots, p - 1$, est un polynôme de degré au plus $2m - 1$;*

(iii) *la fonction $\widehat{x}$ est de classe $C^{2m-2}(T)$.*

En particulier, $\widehat{x}$ est une spline dont les noeuds sont les points t_j où il y a des données.

Définition

$\widehat{x}$ est appelée spline naturelle de lissage d'ordre m associée aux données $(t_j, y_j)_{j=1,\ldots,p}$.

Lissage par moindres carrés pénalisés (4) - Cas des splines cubiques

Lemme

Si $m = 2$ alors,

$$\widehat{x} \in \arg \min_{x \in \mathcal{S}^2(T)} Crit_{PenLS}(x)$$

est définie de la façon suivante :

$$\widehat{x}(t) = a_j(t - t_j)^3 + b_j(t - t_j)^2 + c_j(t - t_j) + d_j, \quad t \in [t_j, t_{j+1}],\ j \in \{2, \ldots, p-2\},$$
$$\widehat{x}(t) = e_j(t - t_j) + f_j \quad t \in [t_j, t_{j+1}],\ j \in \{1, p-1\},$$

avec les contraintes

$$\widehat{x}_{|[t_{j-1},t_j]}(t_j) = \widehat{x}_{|[t_j,t_{j+1}]}(t_j),\ \ \widehat{x}'_{|[t_{j-1},t_j]}(t_j) = \widehat{x}'_{|[t_j,t_{j+1}]}(t_j),\ \ \widehat{x}''_{|[t_{j-1},t_j]}(t_j) = \widehat{x}''_{|[t_j,t_{j+1}]}(t_j).$$

Les dérivées d'ordre 2 et 3 en a et b sont nulles.

Lissage par moindres carrés pénalisés (5) - Choix du paramètre de lissage

$$\text{Crit}_{PenLS}(x) = \sum_{j=1}^{p}(y_j - x(t_j))^2 + \lambda \text{pen}_m(x).$$

- Paramètre inévitable.
- Compromis nécessaire.
  - λ petit : peu de poids pr la pénalité dans la minimisation du critère $\longrightarrow$ ajustement aux données privilégié.
    Cas extrême. $\lambda = 0$: interpolation
  - λ grand : bcp de poids pr la pénalité $\longrightarrow$ caractère lisse privilégié
    Cas extrême. $\lambda = \infty$: droite de régression.

$\longrightarrow$ minimisation d'un critère de validation croisée ou validation croisée généralisée.

Lissage par moindres carrés pénalisés (6) - Code R

$$\text{Crit}_{PenLS}(x) = \sum_{j=1}^{p}(y_j - x(t_j))^2 + \lambda \text{pen}_m(x).$$

- **Pas de minimisation sur l'espace $\mathcal{S}^m(T)$ entier.**
- **Définition d'une base $(\varphi_1, \ldots, \varphi_D)$, et minimisation du critère sur l'espace vectoriel engendré**

  $\longrightarrow$ Calcul de l'objet fonctionnel $\widehat{x}$ tq

$$\widehat{x}(t) = \sum_{k=1}^{D} \widehat{\theta}_k \varphi_k(t), \quad t \in T, \qquad \widehat{\theta} \in \arg\min_{\theta \in \mathbb{R}^D} \text{Crit}_{PenLS}\left(\sum_{j=1}^{D} \theta_j \varphi_j\right).$$

- **Point de départ**
  - Base de fonctions, typiquement le résultat de l'appel à la fonction `create.Nom_base.basis`
  - Vecteur des observations y
- **Fonctions essentielles :** `smooth.basis`, précédé d'un appel à `fdPar`

Lissage par moindres carrés pénalisés (7) - Code R

- Fonction `smooth.basis`
  - Arguments importants

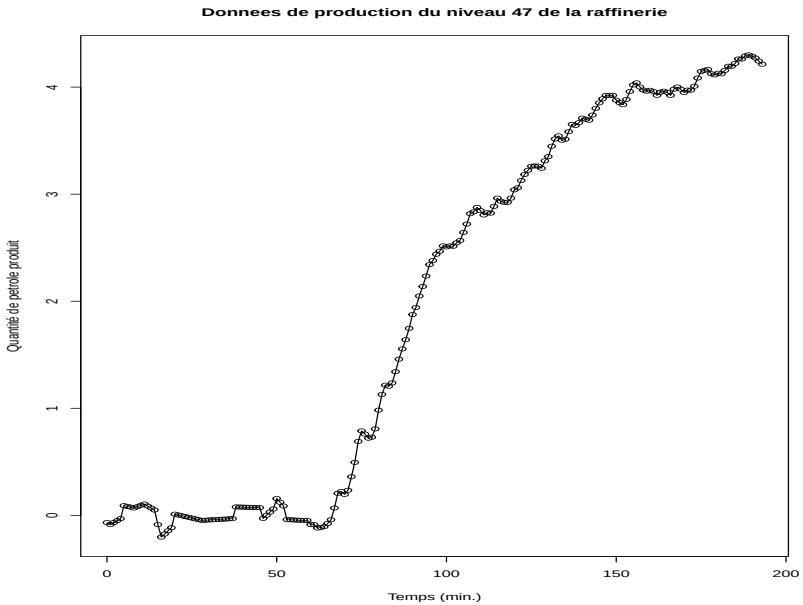
    `argvals` les valeurs des $(t_j)_{j=1,\ldots,p}$;
    `y` les données $(y_j)_{j=1,\ldots,p}$.
    `fdParObj` objet de la classe `basisfd` (sortie de `create.Nom_base.basis`) ou paramètre fonctionnel de la classe `fdPar` (sortie de `fdPar`)
  - Sorties : si `a<-smooth.basis(...)`
    - `a$fd` objet fonctionnel
    - `a$gcv` valeurs du critère de validation croisée généralisée

- Fonction `FdPar` (pour spécifier pénalité - forme et paramètre)
  - Arguments

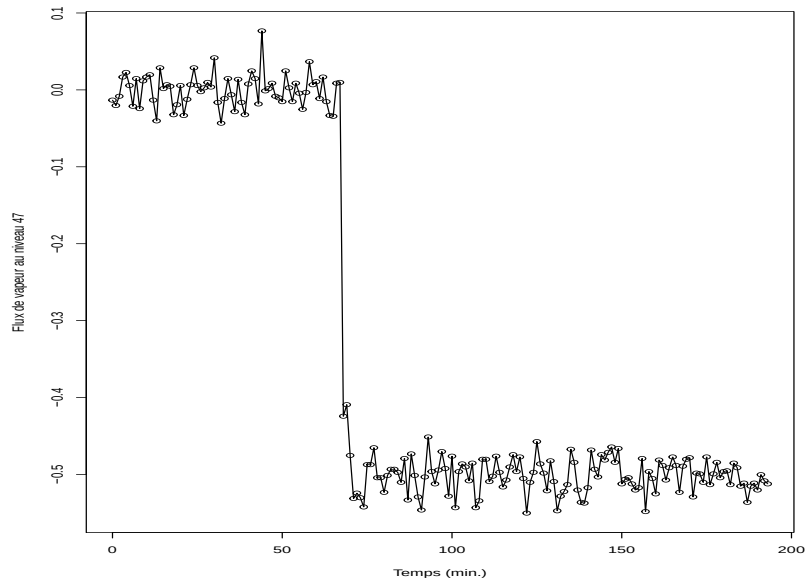
    `fdobj` typiquement, la base de fonctions créée à l'aide de `create.Nom_base.basis` ;
    `Lfdoj` l'ordre m de la dérivée à utiliser si on choisit comme pénalité $\text{pen}_m(x) = \int_T (x^{(m)}(t))^2 dt$ ou alors un opérateur différentiel (défini avec `vec2Lfd` par exemple) ;
    `lambda` le paramètre λ devant la pénalité.
  - Sorties
    - objet de la classe `fdPar`, comportant plusieurs éléments qui mènent à un lissage via `smoothbasis`.

Lissage par moindres carrés pénalisés (8) - Exemple 1

Lissage des données `refinery`.



quantité de pétrole produit
à un certain niveau de la colonne de distillation
en fonction du temps

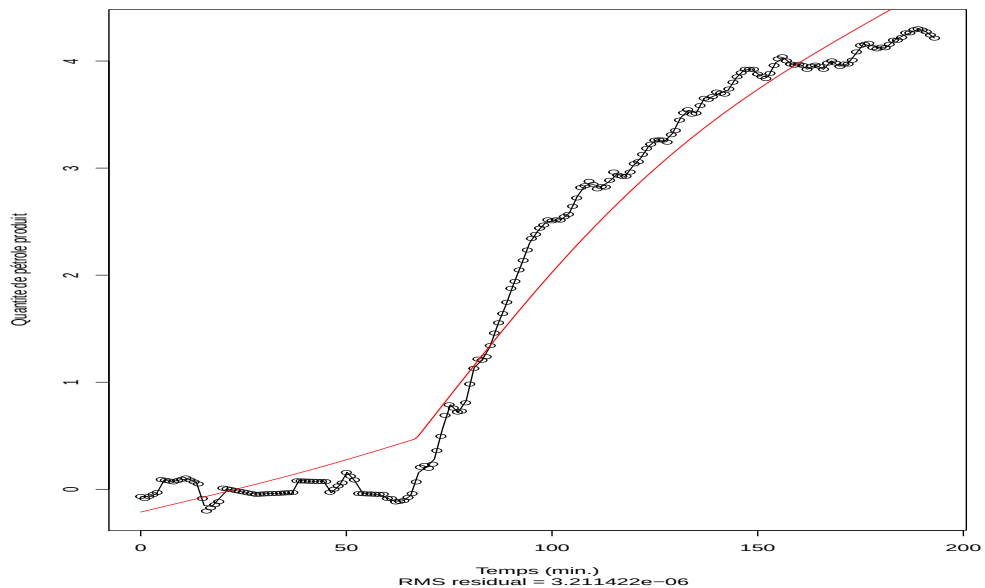


flux de vapeur
dans cette même colonne
en fonction du temps

Lissage par moindres carrés pénalisés (9) - Exemple 1

Lissages des données `refinery`. base de B-splines cubiques avec noeuds aux points d'observation (et un noeud supplémentaire en la discontinuité) et pénalité $\text{pen}_2(x) = \int_T (x''(t))^2 dt$

Courbe noire : obtenue avec $\lambda = 10^{-5}$, Courbe rouge : obtenue avec $\lambda = 10^6$

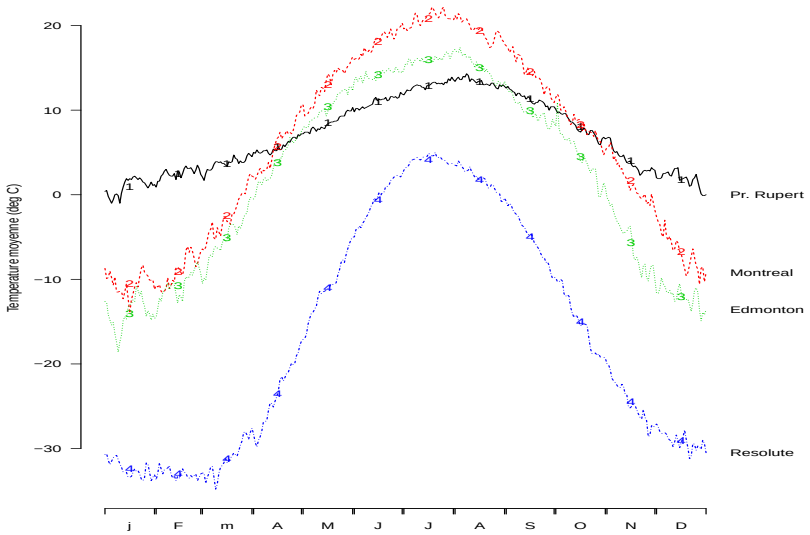


Code R.

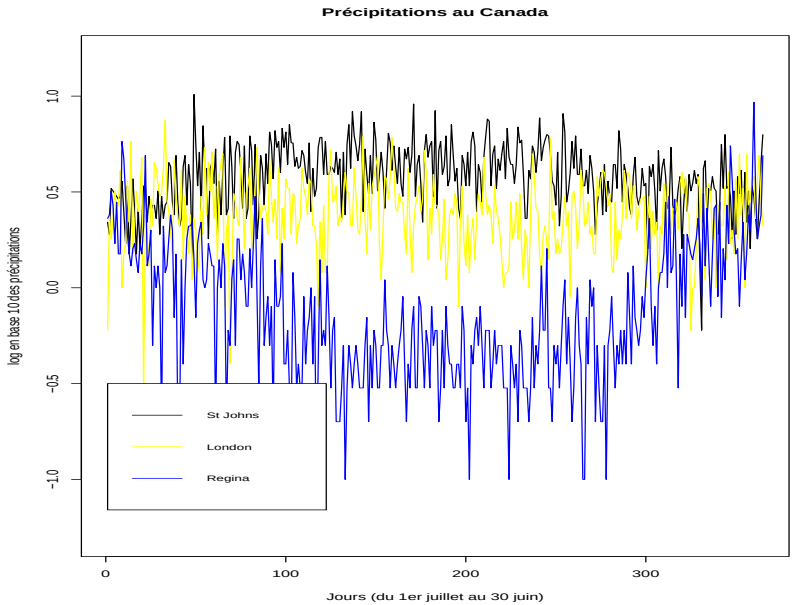
```
Noeuds_refin<-sort(c(refinery$Time,67,67))
Fct_Base_refin<-create.bspline.basis(c(0,193),norder=4,breaks=Noeuds_refin)

fdpar_refin1<-fdPar(fdobj=Fct_Base_refin,Lfdobj = 2,lambda=0.00001)
Donnees_lissees_refin1 <-smooth.basis(argvals=refinery$Time,y=refinery$Tray47,fdParobj=fdpar_refin1)
plotfit.fd(refinery$Tray47,refinery$Time,Donnees_lissees_refin1$fd,xlab='Temps (min.)',ylab="Quantite de pétrole produit")
fdpar_refin2<-fdPar(fdobj=Fct_Base_refin,Lfdobj = 2,lambda=10^6)
Donnees_lissees_refin2 <-smooth.basis(argvals=refinery$Time,y=refinery$Tray47,fdParobj=fdpar_refin2)
lines(Donnees_lissees_refin2$fd,col=2)
```

Lissage par moindres carrés pénalisés (10) - Exemple 2

Lissage des données `CanadianWeather`.



Courbes de température.



Courbes de précipitation.

Lissage par moindres carrés pénalisés (11) - Fonction `vec2Lfd`

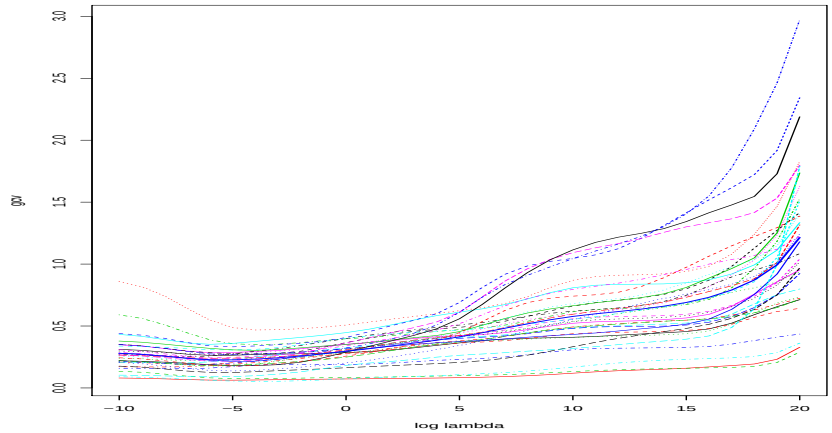
- **Objectifs :** définir un opérateur différentiel de la forme

$$Lx(t) = \beta_0 x(t) + \beta_1 x'(t) + \cdots + \beta_{m-1} x^{(m-1)}(t) + x^{(m)}(t)$$

- **Arguments :**
  - le vecteur des coefficients $(\beta_0, \ldots, \beta_{m-1})$;
  - `rangeval` les extrémités de l'intervalle de définition de l'opérateur.
- **Exemple :** opérateur d'accélération harmonique de période $2\pi/\omega$.
  $Lx = x^{(3)} + (2\pi/\omega)^2 x'$
  premier argument : `c(0,(2*pi/`ω`)^2,0)`.

Lissage par moindres carrés pénalisés (12) - Exemple 2

Lissage des données `CanadianWeather`, températures. base de Fourier de période 365 avec pénalité définie par l'opérateur harmonique, et paramètre λ optimisé par gcv.



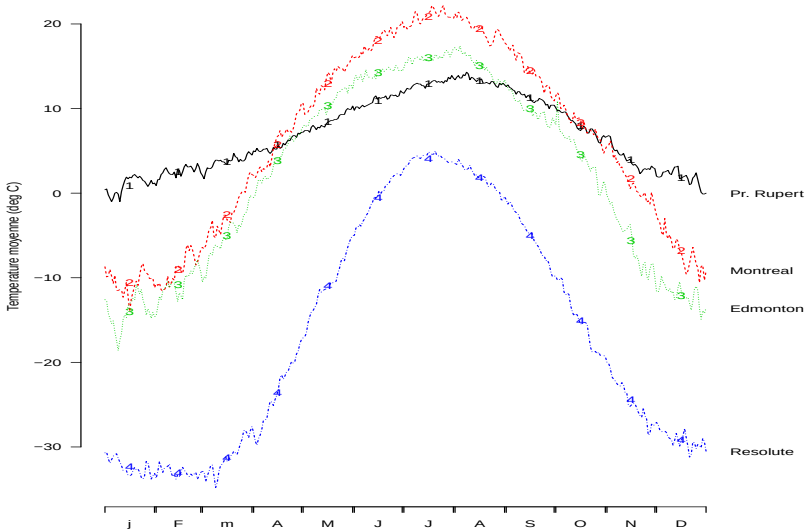
Critère de validation croisée généralisée (gcv) en fonction de λ.

Code R

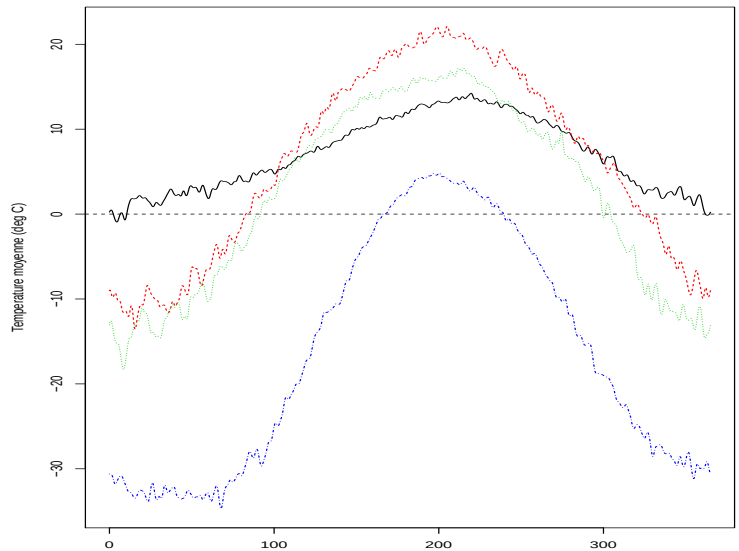
```
temperature = CanadianWeather$dailyAv[,,'Temperature.C']
jours = 1:365 - 0.5
op_harmonique = vec2Lfd( c(0,(2*pi/365)^2,0),c(0,365) )
Fct_base_CanadianW= create.fourier.basis(c(0,365),nbasis=365)

gcvtemp = matrix(0,31,35)
for(i in 1:31){
  lambda = exp(i-10)
  fdPar_CanadianW = fdPar(Fct_base_CanadianW,op_harmonique,lambda)
  gcvtemp[i,] = smooth.basis(jours,temperature,fdPar_CanadianW)$gcv
}
matplot(-10:20,gcvtemp,type='l',xlab='log lambda',ylab='gcv')
mgcvtemp = apply(gcvtemp,1,mean)
lines(-10:20,mgcvtemp,lwd=2,col=4)
```

Lissage par moindres carrés pénalisés (13) - Exemple 2

Lissage des données `CanadianWeather`, températures. base de Fourier de période 365 avec pénalité définie par l'opérateur harmonique, et paramètre λ optimisé par gcv.



Données brutes



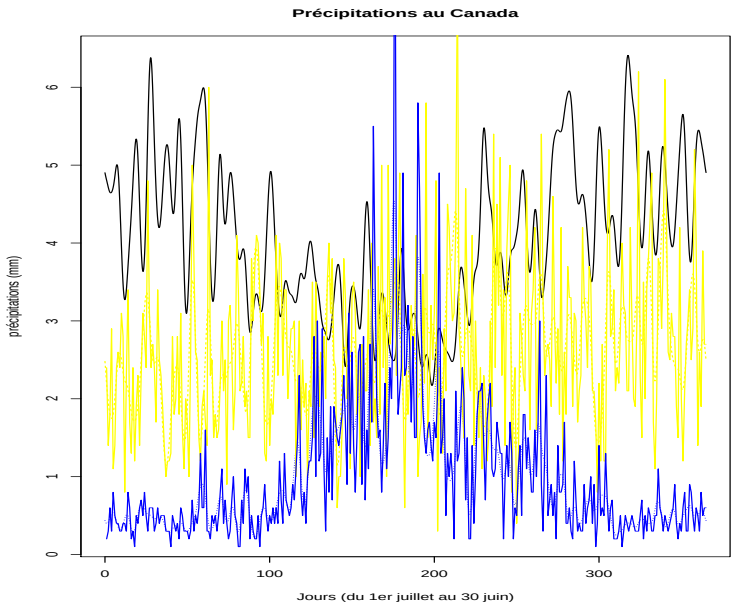
Données lissées

Code R

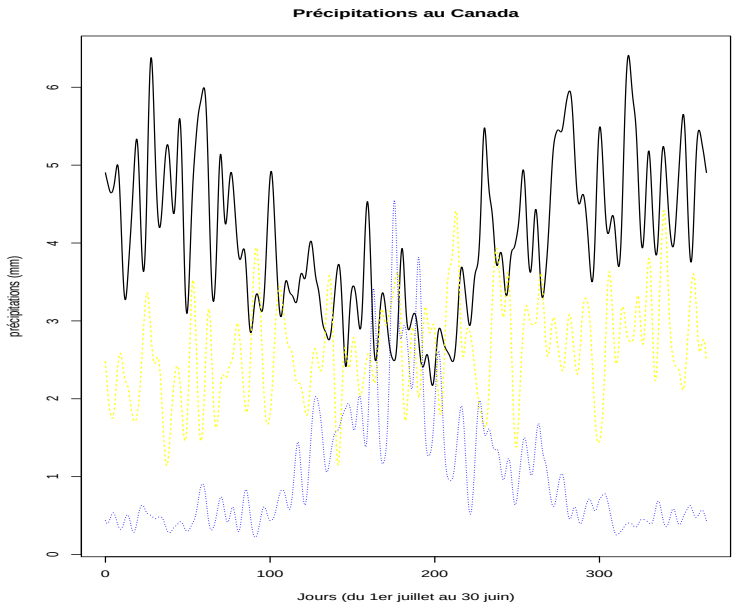
```
i = which.min(mgcvtemp)
lambda_gcv_temp = exp(i-10)
fdPar_CanadianTemp_gcv = fdPar(Fct_base_CanadianW,op_harmonique,lambda_gcv_temp)
Donnees_lissees_temp = smooth.basis(jours,temperature,fdPar_CanadianTemp_gcv)$fd
```

Lissage par moindres carrés pénalisés (14) - Exemple 3

Lissage des données `CanadianWeather`, précipitations. base de Fourier de période 365 avec pénalité définie par l'opérateur harmonique, et paramètre λ optimisé par gcv.



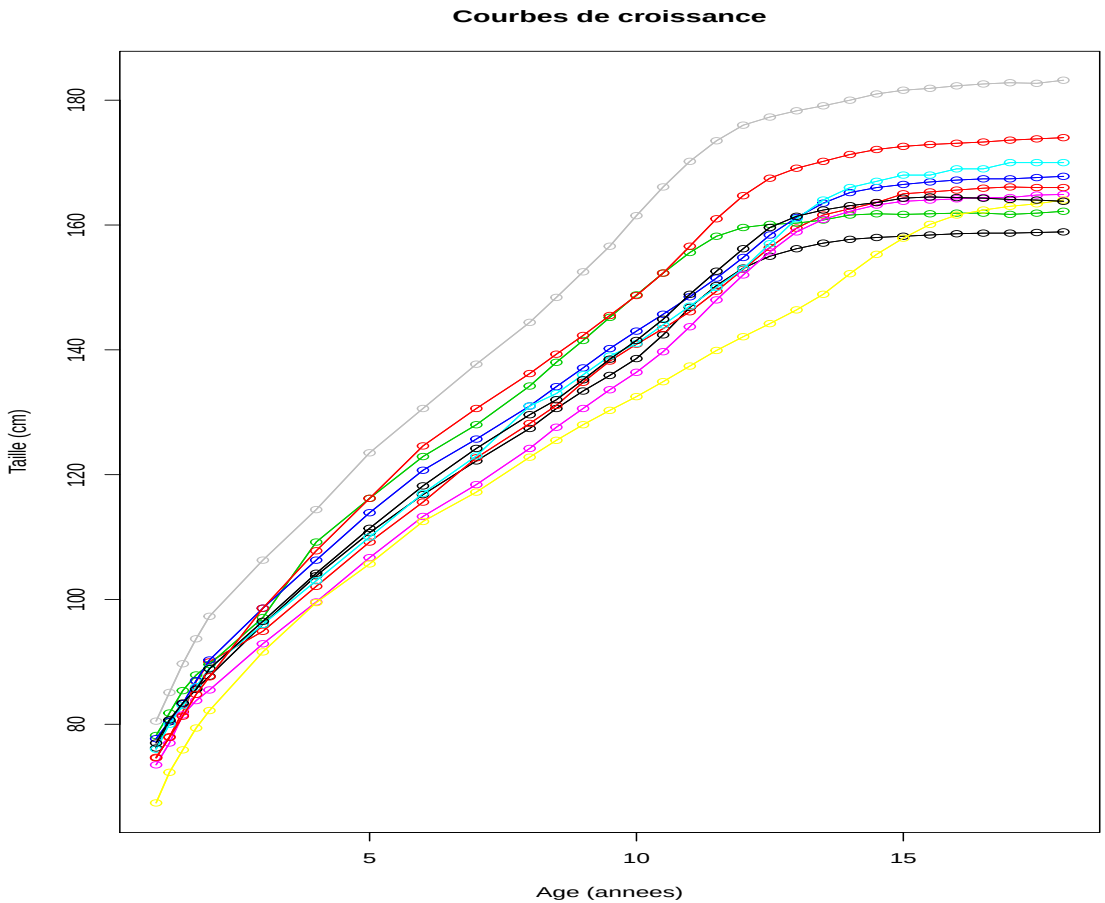
Données brutes



Données lissées

Lissage par moindres carrés pénalisés (15) - Exemple 4

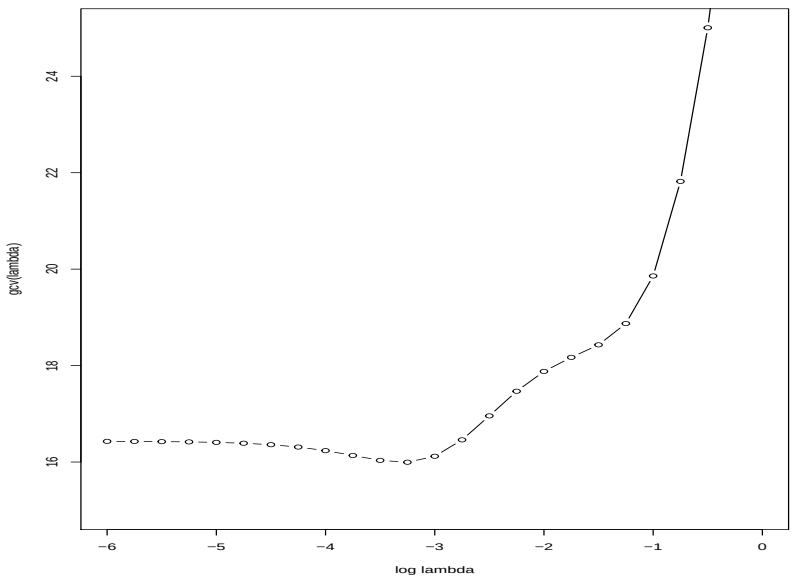
Lissage des données `growth`.



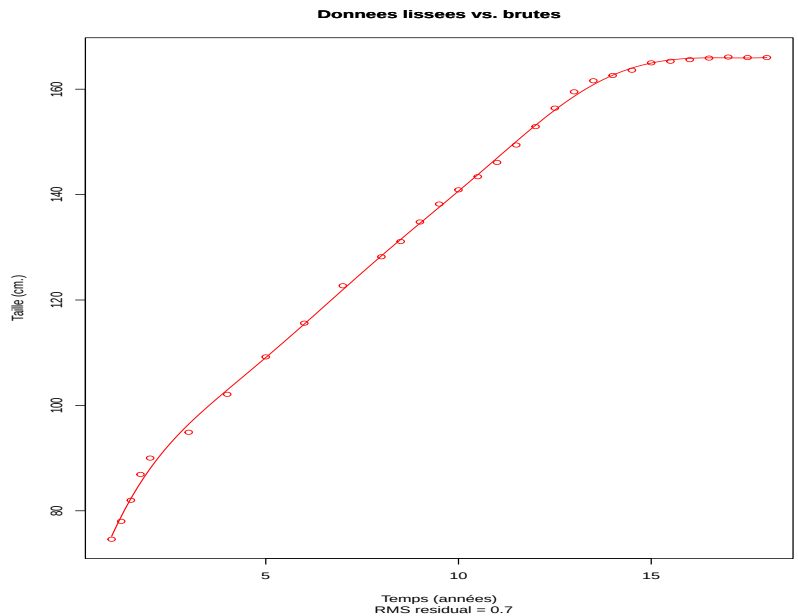
Courbes de croissance de 10 filles, de 1 à 18 ans.

Lissage par moindres carrés pénalisés (20) - Exemple 4

Lissage des données `growth`. base de $B-$splines, d'ordre 6, noeuds à chaque année pénalité $\text{pen}_4(x) = \int_T (x^{(4)}(t))^2 dt$, λ choisi à l'aide du critère gcv.



Somme du critère gcv en fonction de λ pour toutes les filles de l'échantillon



Données lissées vs. données brutes pour 1 individu.