Analyse fonctionnelle de données
Jour 2 : ACP lissée

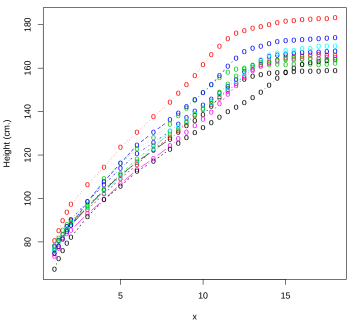
Jean-Marc Lasgouttes, Inria Paris

jean-marc.lasgouttes@inria.fr

Partie I. Données fonctionnelles croissantes

Courbes de croissance

Données une liste contenant la taille de 39 garçons et 54 filles et l'âge auquel elle a été mesurée

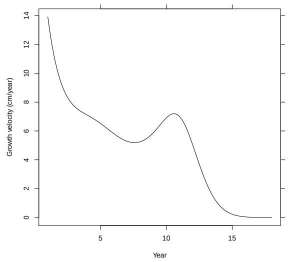


Données mesurées (31 points) pour 10 filles.

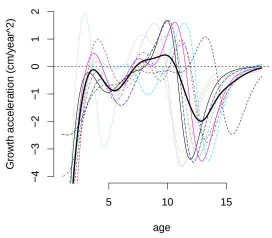
Propriétés
- courbes croissantes
- méthodes paramétriques difficiles d'usage (jusqu'à 8 paramètres)
- croissance rapide au début, puis à l'adolescence
- certains peuvent avoir de l'avance ou du retard
- données très difficiles à collecter

Courbes de croissance : transformation

Calcul de la vélocité on lisse la courbe de taille $H(t)$ et on la dérive pour obtenir la vélocité $V(t)$
- on parle de courbe de croissance, mais on a en fait la taille $H(t)$
- l'expression $V(t_i) = [H(t_{i+1}) - H(t_i)]/[t_{i+1} - t_i]$ est très bruitée



Courbe de vitesse de croissance pour une fille.



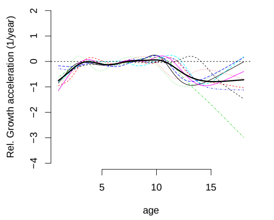
Courbe d'accélération de croissance pour 10 filles (avec la moyenne).

Un modèle de croissance

Écriture du modèle Pour mettre en évidence que l'accélération dépend de la vitesse, on écrit

$$V(t_{i+1}) - V(t_i) = w_i V(t_i)\,(t_{i+1} - t_i)$$

et en passant à la limite $A(t) = w(t)V(t)$



Courbe d'accélération relative pour 10 filles.

Modèle de taille on a en intégrant

$$H(t) = C_2 + C_1 \int_0^t \exp\left[\int_0^u w(s)\,ds\right] du$$

Évaluation on minimise

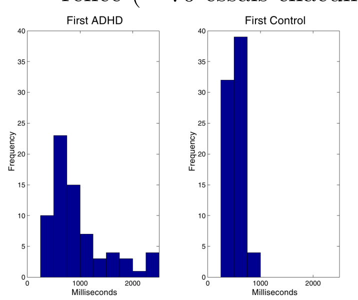
$$\sum_{j=1}^{n} [y_j - H(t_j)]^2 + \lambda \int w'''(t)^2 dt$$

Dépendance non linéaire il faut une optimisation itérative plus lente

Partie II. Cas d'une courbe de densité

Trouble du déficit de l'attention avec hyperactivité

Données
- on affiche un signal à un enfant après un délai d'une dizaine de secondes et il doit appuyer sur un bouton
- 17 enfants souffrant de TDAH et 16 enfants de référence ($\approx$ 70 essais chacun).



Histogrammes pour environ 70 temps de réaction à un signal. Gauche : enfant TDAH ; droite : enfant de référence.

Propriétés
- formes très différentes dans les deux cas
- délai initial fixe de 120ms
- valeurs positives
- somme des valeurs fixée

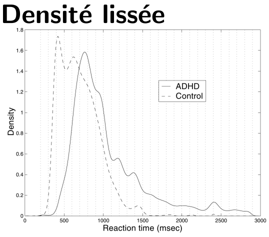
Transformation d'une densité

Problème une densité est contrainte par $\int p(x)\,dx = 1$, on ne peut pas l'approximer.

Réécriture on définit p en fonction d'une nouvelle fonction W sur laquelle il n'y a pas de contrainte

$$p(t) = C \exp W(t), \quad \text{avec } C^{-1} = \int \exp W(x)\,dx$$

Densité lissée



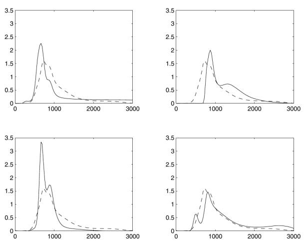
Densité lissée des temps de réaction pour 17 enfants ADHD (ligne pleine) et 16 enfants de contrôle (pointillé).

Méthode maximum de vraisemblance pénalisée par une dérivée 3e

$$\sum_{i=1}^{n} \log_{10} p(t_i) + \lambda \int [W'''(x)]^2 dx$$

Pourquoi W''' ?
- on va s'intéresser à W'
- si $W''' = 0$, alors la distribution est une loi Gaussienne

Modélisation des individus



4 densités de temps de réaction d'enfants TDAH

Action on transforme les temps de réaction

$$z = \log_{10}(t - 120)$$

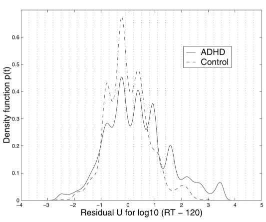
Pourquoi ?
- 120 ms est le temps de réaction minimal observé
- l'impact du TDAH est plutôt multiplicatif qu'additif

Modèle individuel (ANOVA) pour l'essai j de l'enfant i du groupe G_k

$$z_{ijk} = \mu_k + \alpha_{i|k} + U_{ijk}, \qquad \sum_{i \in G_k} \alpha_{i|k} = 0$$

μ_k : performance typique de G_k / $\alpha_{i|k}$: performance typique de l'enfant i de G_k

Les résidus



distribution des résidus

Observations
- $\mu_{\text{TDAH}} = 2.92 > \mu_{\text{ctrl}} = 2.72$ (écart $\approx$ 300ms)
- le temps de réaction de l'enfant i est

$$\tau_i = 120 + 10^{\mu_k + \alpha_{i|k} + U_{ijk}}$$

Forme du résidu
- ce n'est pas une loi normale, il y a plusieurs modes
- on pourrait penser que c'est lié au fonctionnement du cerveau
- en fait c'est seulement lié à la technique de mesure !

Partie III. ACP fonctionnelle

Première approche

Données à chaque individu i est associée une fonction $y_i(t)$ et pour toute fonction $\xi(t)$ telle que $\int \xi(t)^2 dt = 1$, on pose

$$z_i = \int \xi(t) y_i(t)\,dt$$

1re composante principale on cherche la fonction ξ_1 qui maximise la variance empirique de $\mathbf{z} = (z_1, \ldots, z_n)$

$$(\xi_1, \mathbf{z}_1) = \arg\max_{\xi} \operatorname{var} \mathbf{z}, \text{ avec } \int \xi_1(t)^2 dt = 1$$

- $\mathbf{z}_1 = (z_{i1})_{1 \le i \le n}$ est la première composante principale (*score* en anglais)
- $\xi_1(t)$ est le premier facteur principal (*weight* ou *loading function* en anglais)

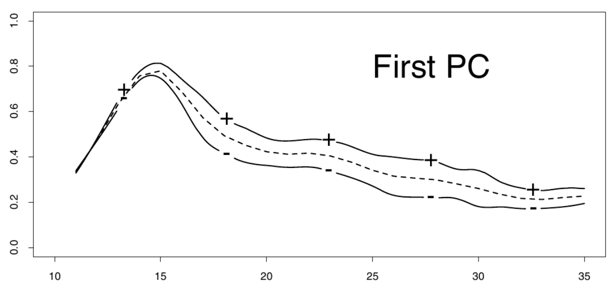
Autres composantes pour la je composante principale, on ajoute les conditions

$$\int \xi_1(t)\xi_j(t)\,dt = \int \xi_2(t)\xi_j(t)\,dt = \cdots = \int \xi_{j-1}(t)\xi_j(t)\,dt = 0$$

ACP classique sur les données de criminologie

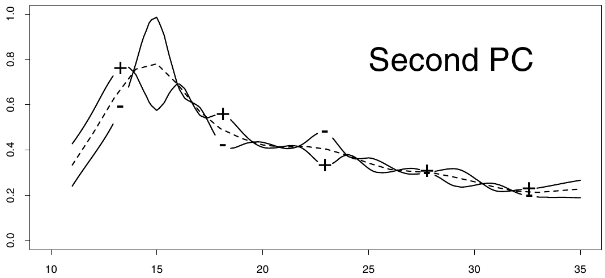
Méthode on applique l'ACP sur les données brutes et on interpole le résultat.

Composante 1 met en évidence les personnes âgées de plus de 15 ans

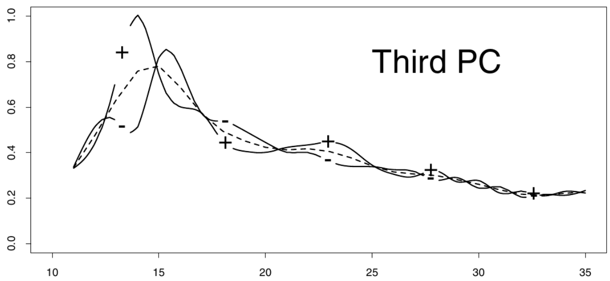


On représente la moyenne et l'effet de l'axe principal multiplié par un coefficient « adapté »

Composante 2 très difficile à interpréter.



Composante 3 très difficile à interpréter.



ACP fonctionnelle lissée

Idée pour éviter les problèmes, on ajoute une facteur de lissage dans l'ACP

1re composante principale on veut contrôler la rugosité de ξ : on modifie donc la contrainte pour arriver au problème

$$(\xi_1, \mathbf{z}_1) = \arg\max_{\xi} \operatorname{var} \mathbf{z}, \text{ avec } \int \xi_1(t)^2 dt + \lambda \int \xi_1''(t)^2 dt = 1$$

— $\mathbf{z}_1 = (z_{i1})_{1\le i\le n}$ est la première composante principale (*score* en anglais)
— $\xi_1(t)$ est le premier facteur principal (*weight* ou *loading function* en anglais)
— λ est la pénalité de rugosité

Autres composantes pour la k^{e} composante principale, on a les conditions

$$\int \xi_k(t)^2 dt + \lambda \int \xi_k''(t)^2 dt = 1$$

$$\int \xi_k(t)\xi_\ell(t)\, dt + \lambda \int \xi_k''(t)\xi_\ell''(t)\, dt = 0, \text{ si } k \neq \ell$$

L'ACP fonctionnelle en pratique

Les parts de variation ce sont les variances des composantes principales $\mathbf{z}_k$, en ordre décroissant

$$\operatorname{var} \mathbf{z}_1 \geq \operatorname{var} \mathbf{z}_2 \geq \cdots \geq \operatorname{var} \mathbf{z}_p \geq 0$$

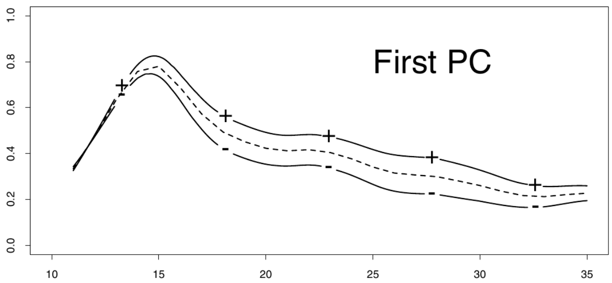
Les facteurs principaux on peut les représenter comme pour les données de criminologie, comme un effet sur la moyenne.

Sélection de variables
— on garde les variables de 1 à q, pas des variables arbitraires
— on peut se servir de la courbe des valeurs propres (coude)
— de toute façon, seuls les axes interprétables sont intéressants.

ACP lissée sur les données de criminologie

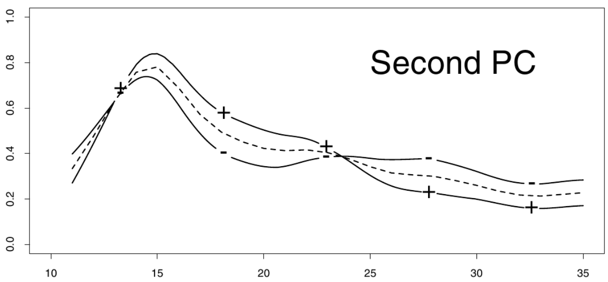
Méthode on applique l'ACP lissée sur les données.

Composante 1 met en évidence les délinquants qui commencent à la fin de l'adolescence

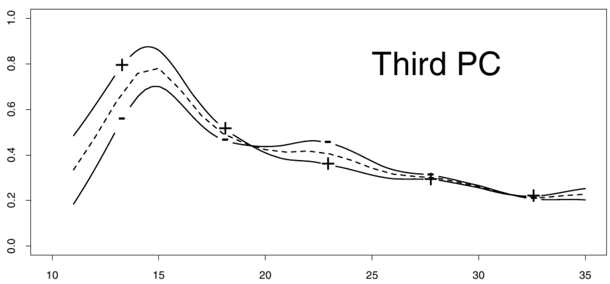


On représente la moyenne et l'effet de l'axe principal multiplié par un coefficient « adapté »

Composante 2 représente la désistance à long terme : la délinquance s'arrête avant 25 ans et le comportement devient meilleur que la moyenne



Composante 3 représente la délinquance juvénile : commence très jeune, s'améliore vers 20 ans, puis revient au niveau normal



Partie IV. Méthodes mathématiques

Approche matricielle

Données elles sont décrites avec les coefficients $\mathbf{Y} = (y_{ij})$ et une base de fonctions $\delta_j(t)$,

$$y_i(t) = \sum_{j=1}^{m} y_{ij}\delta_j(t), \quad \text{et } \bar{y}_j = \frac{1}{n}\sum_{i=1}^{n} y_{ij}$$

La matrice $\mathbf{V}$ de variance-covariance de $\mathbf{Y}$ s'écrit

$$V_{j\ell} = \frac{1}{n-1}\sum_{i=1}^{n}(y_{ij} - \bar{y}_j)(y_{i\ell} - \bar{y}_\ell)$$

Calcul des composantes on pose

$$\xi(t) = \sum_{k=1}^{p} \xi_k\beta_k(t), \qquad L_{jk} = \int \delta_j(t)\beta_k(t)\,dt$$

et on peut alors écrire

$$\mathbf{z} = \int \xi(t)\mathbf{y}(t)\,dt = \mathbf{Y}\mathbf{L}'\boldsymbol{\xi}, \qquad \operatorname{var}\mathbf{z} = \boldsymbol{\xi}'\mathbf{L}\mathbf{V}\mathbf{L}'\boldsymbol{\xi}$$

Maximisation sous contrainte

Contrainte la contrainte sur ξ_1 s'écrit

$$\boldsymbol{\xi}'(\mathbf{J} + \lambda\mathbf{K})\boldsymbol{\xi} = 1$$

$$J_{jk} = \int \beta_j(t)\beta_k(t)\,dt \qquad K_{jk} = \int \beta_j''(t)\beta_k''(t)\,dt$$

Lagrangien on définit la *métrique* $\mathbf{M} = \mathbf{J} + \lambda\mathbf{K}$ et on maximise la grandeur

$$\mathcal{L}(\boldsymbol{\xi}, \theta) = \boldsymbol{\xi}'\mathbf{L}\mathbf{V}\mathbf{L}'\boldsymbol{\xi} - \theta\left[\boldsymbol{\xi}'\mathbf{M}\boldsymbol{\xi} - 1\right]$$

Résolution les équations sont

$$\mathbf{L}\mathbf{V}\mathbf{L}'\boldsymbol{\xi} - \theta\mathbf{M}\boldsymbol{\xi} = 0 \qquad \boldsymbol{\xi}'\mathbf{M}\boldsymbol{\xi} = 1$$

et finalement $\boldsymbol{\xi}$ est un vecteur propre de la matrice $\mathbf{M}^{-1}\mathbf{L}\mathbf{V}\mathbf{L}'$

$$\mathbf{M}^{-1}\mathbf{L}\mathbf{V}\mathbf{L}'\boldsymbol{\xi} = \theta\boldsymbol{\xi} \qquad \boldsymbol{\xi}'\mathbf{M}\boldsymbol{\xi} = 1$$

Problème on ne sait rien du spectre de $\mathbf{M}^{-1}\mathbf{L}\mathbf{V}\mathbf{L}'$

Diagonalisation

Propriété Soit $\mathbf{M}$ une matrice réelle symétrique définie positive. une matrice $\mathbf{S}$ est $\mathbf{M}$-symétrique si $\mathbf{S}'\mathbf{M} = \mathbf{M}\mathbf{S}$. Dans ce cas

- $\mathbf{S}$ est une matrice diagonalisable et ses vecteurs/valeurs propres sont réels.
- Elle possède une base de vecteurs propres $\mathbf{v}_1, \ldots, \mathbf{v}_p$ qui est $\mathbf{M}$-orthonormale

$$\mathbf{v}_k'\mathbf{M}\mathbf{v}_\ell = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{sinon} \end{cases}$$

Application la matrice $\mathbf{M}^{-1}\mathbf{L}\mathbf{V}\mathbf{L}'$ est $\mathbf{M}$-symétrique :

$$\left[\mathbf{M}^{-1}\mathbf{L}\mathbf{V}\mathbf{L}'\right]'\mathbf{M} = \mathbf{L}\mathbf{V}\mathbf{L}' = \mathbf{M}\left[\mathbf{M}^{-1}\mathbf{L}\mathbf{V}\mathbf{L}'\right]$$

Elle possède donc une base de vecteurs propres $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_p$ tels que

$$\boldsymbol{\xi}_k'\mathbf{M}\boldsymbol{\xi}_\ell = \begin{cases} 1 & \text{si } k = \ell \\ 0 & \text{sinon} \end{cases}$$

Ces vecteurs vérifient donc les contraintes demandées dans la définition de l'ACP.

L'ACP fonctionnelle lissée

Facteur principal c'est le vecteur propre $\boldsymbol{\xi}_k$ associé à θ_k

Composantes principale c'est $\mathbf{z}_k = \mathbf{Y}\mathbf{L}'\boldsymbol{\xi}_k$, dont la variance est

$$\operatorname{var}\mathbf{z}_k = \boldsymbol{\xi}_k'\mathbf{L}\mathbf{V}\mathbf{L}'\boldsymbol{\xi}_k = \boldsymbol{\xi}_k'\mathbf{M}\mathbf{M}^{-1}\mathbf{L}\mathbf{V}\mathbf{L}'\boldsymbol{\xi}_k = \theta_k\boldsymbol{\xi}_k'\mathbf{M}\boldsymbol{\xi}_k = \theta_k$$

Les composantes principales sont décorrélées

$$\operatorname{cov}(\mathbf{z}_k, \mathbf{z}_\ell) = \boldsymbol{\xi}_k'\mathbf{L}\mathbf{V}\mathbf{L}'\boldsymbol{\xi}_\ell = \theta_\ell'\boldsymbol{\xi}_k'\mathbf{M}\boldsymbol{\xi}_\ell = 0 \quad \text{si } k \neq \ell$$

Valeurs propres on classe les valeurs propres par ordre décroissant

$$\theta_1 \geq \theta_2 \geq \ldots \geq \theta_p \geq 0$$

Elles sont positives parce que ce sont des variances

Lien avec l'ACP classique L'ACP fonctionnelle est une sorte d'ACP centrée des coefficients $\mathbf{Y}$ sous la métrique $\mathbf{M}$

Partie V. Utilisation du paquet « fda »

Lissage de densité

Fonction calcule une densité de la forme $C^{-1} \exp W(t)$

```
denslist <- density.fd(x, WfdParobj,
    conv=0.0001, iterlim=20, dbglev=1...)
```

Paramètres utiles
- `x` : ensemble d'observations. Soit un vecteur de valeurs (avec un poids uniforme), soit une matrice à 2 colonnes (valeurs, poids)
- `WfdParObj` : un objet retourné par `fdPar()`

Paramètres peut-être utiles si la convergence est difficile
- `conv` : précision pour la convergence de l'algorithme
- `iterlim` : nombre max d'itérations
- `dbglev` : niveau d'information (0 : rien, 1 : normal, 2 : maximum)

Valeur retournée une liste contenant
- `Wfdobj` : un objet de données fonctionnelles décrivant la fonction $W(x)$
- `C` : la constante de normalisation

Utilisation : `with(denslist, exp(eval.fd(evalarg, Wfdobj)) / C)`

Lissage de fonction monotone

Fonction calcule une fonction de la forme $\beta_0 + \beta_1 \int \exp W(u)\, du$

```
monfdobj <- smooth.monotone(argvals, y, WfdParobj,
    conv=.0001, iterlim=50, ...)
```

Paramètres
- `argvals` : les abscisses t_j des points mesurés
- `y` : les données correspondantes (en colonnes si plusieurs individus)
- `WfdParobj` : un objet produit par `fdPar()`
- `conv`, `interlim` : voir `density.fd`

Valeur retournée un object de classe `monfd`, qui contient
- `Wfdobj` : un objet de données fonctionnelles décrivant la fonction $W(x)$
- `beta` : les coefficients de régression b_0 et b_1 pour chaque courbe lissée
- `yhatfd` : donnée fonctionnelle pour la courbe monotone lissée
- `argvals`, `y` : les paramètres d'entrée

Utilisation la fonction suivante évalue une fonction à une série de points `newdata`

```
predict(object, newdata=NULL, ...)
```

On peut utiliser `eval.monfd`, mais il ne prend pas en compte `monfdobj$beta`

Lissage de fonction positive

Fonction calcule une fonction de la forme $\exp W(t)$

```
posfdobj <- smooth.pos(argvals, y, WfdParobj,
    conv=.0001, iterlim=50, ...)
```

Paramètres
- `argvals` : les abscisses t_j des points mesurés
- `y` : les données correspondantes. Si on a plusieurs individus, alors ils sont en colonne (chaque ligne correspond à un t_j)
- `WfdParobj` : un objet produit par `fdPar()`
- `conv`, `interlim` : voir `density.fd`

Valeur retournée un object de classe `posfd`, qui contient
- `Wfdobj` : un objet de données fonctionnelles décrivant la fonction $W(t)$
- `argvals`, `y` : les paramètres d'entrée

Utilisation Typiquement on utilise `eval.posfd` ou `predict`

```
with(posfdobj, eval.posfd(evalarg, Wfdobj))
predict(posfdobj, newdata=evalarg)
```

ACP lissée

Fonction

```
pca.fd(fdobj, nharm = 2, harmfdPar=fdPar(fdobj),
       centerfns = TRUE)
```

Paramètres
- `fdobj` : données fonctionnelles décrivant les individus
- `nharm` : le nombre d'axes principaux à calculer
- `harmdfPar` : un objet retourné par `fdPar()`
- `centerfns` : indique si les données doivent être centrées

Valeur un objet de classe `pca.fd` qui contient
- `harmonics` : données fonctionnelles représentant les fonctions propres
- `values` : toutes les valeurs propres
- `scores` : matrices des scores ($\mathbf{z}$)
- `varprop` : proportion de variance expliquée par chaque valeur propre
- `meanfd` : données fonctionnelles représentant la moyenne

Représentation des fonctions propres

Fonction sert à représenter des fonction propres de l'ACP. Chacune est représentée en terme de variation autour de la moyenne.

```
plot(x, nx = 128, pointplot = TRUE, harm = 0,
          expand = 0, cycle = FALSE, ...)
```

Paramètres voir la documentation de plot pour plus de détails (`titles`, `xlab`, `ylab`, ...)
- `x` : le résultat de `pca.fd`
- `nx` : nombre de points à tracer en abscisse
- `pointplot` : si `TRUE`, utiliser des $+/-$ pour dénoter les variations, sinon des courbes

— `harm` : les harmoniques à afficher (un plot par harmonique), toutes affichées par défaut
— `expand` : la distance entre les $+/-$ et la moyenne (défaut : 2 écarts types)
— `cycle` : si `length(harm) == 2`, représente les harmoniques par un cycle