Analyse fonctionnelle de données
Jour 1 : les données

Jean-Marc Lasgouttes, Inria Paris

jean-marc.lasgouttes@inria.fr

Partie I. Introduction au cours

Organisation du cours

Trois jours

1. description et préparation des données, calculs de moyennes, bases de fonctions
2. encore des transformations de données, ACP lissée
3. recalage temporel, recalage en amplitude

Matin Cours (3 heures)
- démonstration des méthodes sur des exemples
- description des outils mathématiques nécessaires

Après midi TP (4 heures)
- description des fonctions du paquet « fda » de R nécessaires
- application des méthodes sur un jeu de données
- rédaction d'un rapport rapide décrivant votre approche et vos résultats.

Qu'est-ce que l'analyse fonctionnelle de données ?

Données fonctionnelles Il s'agit de données dans lesquelles chaque individu peut être assimilé à une courbe

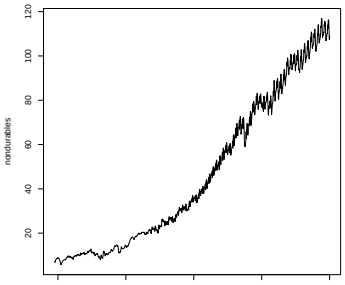
Plusieurs étapes
- préparation des données pour utiliser des modèles linéaires
- stockage sous forme de fonctions
- recalage en phase et amplitude
- éventuellement différentiation pour faire apparaître les phénomènes
- application d'une ACP lissée

Références
- ce cours suit « Applied Functional Analysis » de Ramsay et Silverman, 2002 (Springer)
- des développements plus mathématiques sont dans « Functionnal Data Analysis » des mêmes auteurs (2^e édition, 2005, Springer)

Partie II. Les données fonctionnelles par l'exemple

Production de bien consommables

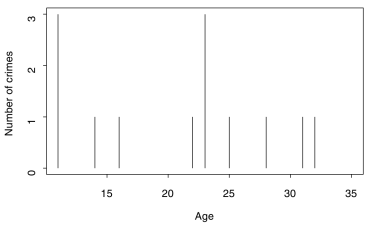
Nondurable goods index Mesure la production de biens consommables (nourriture, tabac, vêtements, essence...) mois par mois de janvier 1919 à janvier 2000.



Propriétés
- périodicité annuelle : augmentation en fin d'année (cadeaux de Noël), à la rentrée (vêtements), baisse en été (ouvriers en vacances)
- donc 80 individus à 12 variables chacun
- forme pas vraiment linéaire (augmentation en pourcentage)
- évolution difficile à comprendre

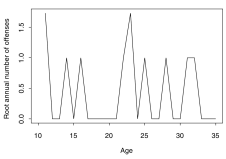
Criminologie

Données nombre d'arrestations par an entre les âges de 11 et 35 ans pour 413 hommes américains. On veut comprendre leur « carrière » criminelle.



Données enregistrées pour un individu particulier
(arrêté 3 fois à 11 ans, 1 fois à 15 ans, etc.).

Propriétés
- peu de données
- données très bruitées : activité ≠ arrestation
- pas vraiment une courbe



Partie III. Préparation des données

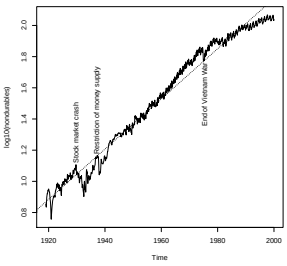
Biens consommables : transformation

Action on considère le logarithme $\log_{10} I$ de l'index et on lisse la courbe

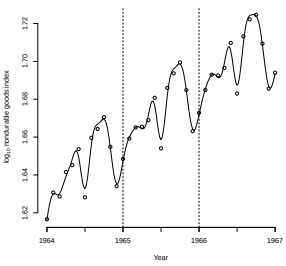
Pourquoi ?

— en général, il y a une progression géométrique (x% par an)

— chaque année devient un individu avec des propriétés comparables



$\log_{10} I$ *avec dates annotées*



Courbe lissée pour 1964-67

Criminologie : transformation des données

Action On prend la racine carrée du nombre d'arrestations

Pourquoi ?

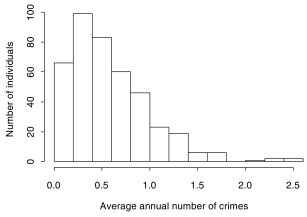
— beaucoup de valeurs très grandes, risque de surreprésentation

— la racine carrée est un *stabilisateur de variance* pour le processus de Poisson :

  — Si X est Poisson de paramètre λ, alors $\mathbb{E}X = \operatorname{var} X = \lambda$

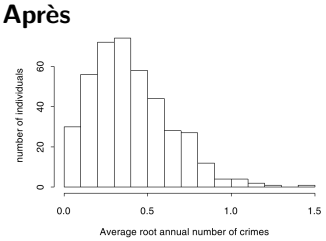
  — Par contre, $\sqrt{X}$ est approximativement normale de moyenne $\sqrt{\lambda}$ et de variance 1

Avant



Distribution du nombre d'arrestations par an

Après



Distribution de la racine carrée du nombre d'arrestations par an

Criminologie : création d'une courbe

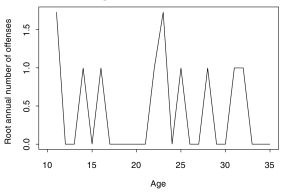
Action on relie les points par des lignes droites

Pourquoi ?

— maintenant, chaque délinquant est représenté par une courbe

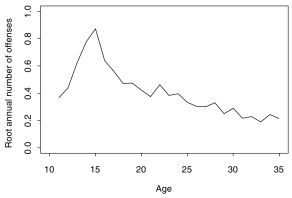
— on ne lisse pas la courbe pour ne pas perdre d'information

Un délinquant



Interpolation linéaire des racines carrées des nombre d'arrestation d'un sujet

Courbe moyenne



Moyenne empirique des données fonctionnelles de criminologie

Observation La moyenne empirique des 413 personnes est très bruitée

Partie IV. Des données vers les fonctions

Lissage de courbe

Pourquoi ?

— on veut la variation des données, il faut donc pouvoir les dériver

— les calculs empiriques de dérivée sont très bruités

Données points de mesure $t_1, \dots, t_m$ (peuvent dépendre de l'individu) et série de valeurs $y_1, \dots, y_m$ prises aux points t_i

Sortie fonction h « lisse » (donc avec h'' pas trop grand) qui approxime les valeurs

Méthode on cherche la fonction h_λ qui minimise la fonctionnelle pénalisée

$$E_\lambda(h) = \sum_{i=1}^{m} [y_j - h(t_j)]^2 + \lambda \int [h''(t)]^2 dt$$

Paramètre à régler λ agit sur la courbure de h_λ (c'est la pénalisation de rugosité)

— trop petit : la courbe est proche des points, mais elle n'est pas lisse

— trop grand : on obtient une droite (dérivée seconde nulle)

La question des instants de mesure

Cas simple Toutes les courbes sont mesurées à des points communs t_i.

$\Longrightarrow$ calcul très simple, tout pourra être fait d'un coup

Valeurs manquantes Des points manquent pour certains t_i.

$\Longrightarrow$ on doit lisser les courbes un à une, on n'utilise que les t_i utiles

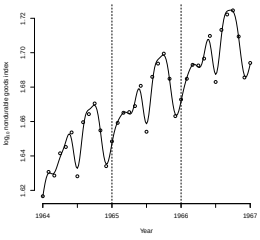
Mesures non syncronisées Chaque individu a son propre jeu de t_i

$\Longrightarrow$ on lisse les courbes une à une avec ses t_i.

Résultat Dans tous les cas, on a un jeu de courbes qui sont définies sur la même base !

Exemple : biens consommables

Trois années



Courbe lissée pour 1964-67 ($\lambda = 10^{-9.5}$)

Quelles données ?

— on s'intéresse au taux d'augmentation de la consommation plutôt que sa valeur
— donc on regarde aussi sa dérivée (accélération)

Le critère on veut que l'accélération de la courbe soit lisse. On prend donc comme fonction de coût

$$E_\lambda(h) = \sum_{i=1}^{m} [y_i - h(t_i)]^2 + \lambda \int [h^{(4)}(t)]^2 dt$$

Choix de λ : complètement subjectif
— assez petit pour bien suivre les points
— assez grand pour que les courbes de vitesse/accélération soient bien lisses

On doit regarder une par une toutes les courbes

Exemple : lissage de la moyenne

Données Chaque individu est représenté par une fonction $Y_i(t)$, $i = 1, \ldots, n$, lisse ou pas. La moyenne empirique est

$$\overline{Y}(t) = \frac{1}{n} \sum_{i=1}^{n} Y_i(t)$$

Problème dans certains cas, la moyenne est bruitée, on cherche une moyenne g lisse

Méthode on minimise encore un critère de la forme

$$E_\lambda(g) = \int [g(t) - \overline{Y}(t)]^2 dt + \lambda \int [g''(t)]^2 dt$$

Choix de λ par validation croisée

Problème Normalement on doit évaluer la qualité de la courbe sur des données de test. C'est impossible quand on a peu de données.

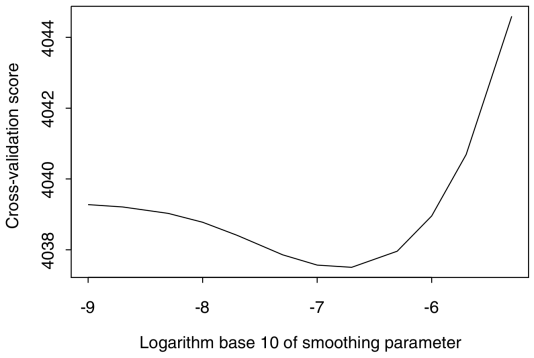
Idée on sait que la moyenne est le meilleur prédicteur d'une variable au sens L^2

$$\arg\min_{\mu} \mathbb{E}[X - \mu]^2 = \mathbb{E}X$$

Méthode on enlève l'individu i et on calcule la moyenne lisse $g_\lambda^{-i}(t)$. On cherche alors

$$\lambda_0 = \arg\min_{\lambda} \sum_{i=1}^{n} \int [g_\lambda^{-i}(t) - Y_i(t)]^2 dt$$

C'est la valeur de λ qui donne la moyenne la plus fidèle



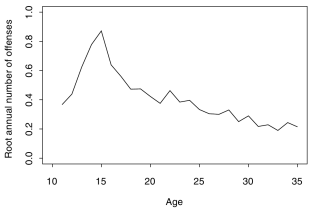
En pratique on ajuste subjectivement (encore !)

Exemple : criminologie

Observation

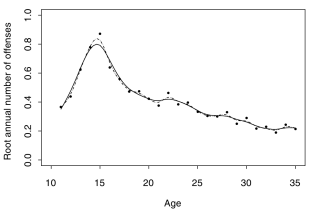
— La moyenne empirique présente des artefacts (moins d'arrestations à 29 ans qu'à 28 ou 30).
— la moyenne optimale par validation croisée ($\lambda = 2 \times 10^{-7}$) est encore « sinueuse »
— on choisit $\lambda = 10^{-6}$ de manière subjective

Moyenne empirique



Moyenne empirique des données fonctionnelles de criminologie

Moyenne lissée



Moyenne lissée des données fonctionnelles de criminologie (pointillé : $\lambda = 2 \times 10^{-7}$; ligne pleine : $\lambda = 10^{-6}$)

Partie V. Outils mathématiques

Représentation fonctionnelle

Pourquoi ? On pourrait conserver les données sous forme de points, mais...
— le vrai objet est la fonction
— on peut avoir des données évaluées à des points différents ou des données manquantes
— on est souvent intéressé par les dérivées des fonctions

Comment ?
— on choisit une base de fonctions formée de p éléments $\beta_1(t), \ldots, \beta_p(t)$
— à un ensemble de coefficients $\xi_1, \ldots, \xi_p$ on associe la fonction

$$y(t) = \sum_{k=1}^{p} \xi_k \beta_k(t)$$

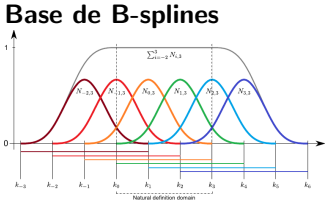
La base choix classiques
— base polygonale (criminologie)
— base de B-splines (biens consommables)
— base de Fourier (fonctions périodiques), ondelettes

Les B-splines

Propriétés

- une B-spline d'ordre m (degré $n = m-1$) est non-nulle sur m intervalles au plus
- la fonction et ses dérivées d'ordre $1, \dots, m-2$ sont continues
- flexibles, numériquement stables
- les calculs sur les B-splines sont très rapides (de complexité $O(p)$)

Base de B-splines



Une base de B-splines cubiques uniformes

Nombre de splines en général supérieur au nombre de points, typiquement

$$p = \text{nb points} - 2 + m$$

Ordre il doit être assez grand pour permettre de dériver la fonction autant que nécessaire

Données on se donne un ensemble croissant de nœuds $t_0, \dots, t_m$

Définition chaque B-spline est formée de plusieurs morceaux de degré n, définis par récurrence sur le degré inférieur :

$$b_{j,0}(t) = \begin{cases} 1 & \text{si} \quad t_j \leqslant t < t_{j+1} \\ 0 & \text{sinon} \end{cases}$$

$$b_{j,n}(t) = \frac{t - t_j}{t_{j+n} - t_j} b_{j,n-1}(t) + \frac{t_{j+n+1} - t}{t_{j+n+1} - t_{j+1}} b_{j+1,n-1}(t)$$

Exemples l'application de la récurrence avec des nœuds placés en 0, 1, 2 et 3 donne les morceaux de la B-spline uniforme de degré 2

$$\begin{aligned} b_{1,2}(t) &= t^2/2 & 0 \leq t \leq 1 \\ b_{2,2}(t) &= (-2t^2 + 6t - 3)/2 & 1 \leq t \leq 2 \\ b_{3,2}(t) &= (3-t)^2/2 & 2 \leq t \leq 3 \end{aligned}$$

Calcul des coefficients sur la base

Cadre général on a les valeurs $y_1, \dots, y_q$ aux points de mesure $t_1, \dots, t_q$ et un ensemble de fonctions de base $\beta_k(t)$.

On note $\mathbf{B}$ la matrice de taille $q \times p$ définie par $B_{jk} = \beta_k(t_j)$. Alors

$$y_j = \sum_{k=1}^{p} \xi_k \beta_k(t_j) = \sum_{k=1}^{p} \xi_k B_{jk}, \text{ c'est-à-dire } \mathbf{y} = \mathbf{B}\boldsymbol{\xi}$$

Régression moindre carrés ($p \leq q$) méthode classique de moindre carrés : on minimise

$$(\mathbf{B}\boldsymbol{\xi} - \mathbf{y})'(\mathbf{B}\boldsymbol{\xi} - \mathbf{y}) = \boldsymbol{\xi}'\mathbf{B}'\mathbf{B}\boldsymbol{\xi} - 2\boldsymbol{\xi}'\mathbf{B}'\mathbf{y} + \mathbf{y}'\mathbf{y}$$

En dérivant par rapport à $\boldsymbol{\xi}$, on obtient l'équation

$$2\mathbf{B}'\mathbf{B}\boldsymbol{\xi} - 2\mathbf{B}'\mathbf{y} = 0$$

soit

$$\boldsymbol{\xi} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{y}$$

Régression avec lissage

- si $p > q$, il y a un continuum de solutions qui décrivent $\mathbf{y}$ ($\mathbf{B}'\mathbf{B}$ non inversible)
- même si $p \leq q$, on peut vouloir appliquer une pénalité de rugosité

$$\lambda \int y''(t)^2 dt = \lambda \boldsymbol{\xi}'\mathbf{K}\boldsymbol{\xi}, \quad \text{où } K_{k\ell} = \int \beta_k''(t)\beta_\ell''(t)\, dt$$

Minimisation on minimise maintenant

$$(\mathbf{B}\boldsymbol{\xi} - \mathbf{y})'(\mathbf{B}\boldsymbol{\xi} - \mathbf{y}) + \lambda \boldsymbol{\xi}'\mathbf{K}\boldsymbol{\xi}$$

Le minimum est atteint pour

$$\boldsymbol{\xi} = (\mathbf{B}'\mathbf{B} + \lambda\mathbf{K})^{-1}\mathbf{B}'\mathbf{y}$$

Cas des B-splines Les matrices $\mathbf{B}'\mathbf{B}$ et $\mathbf{K}$ ont une structure en bande, et les calculs matriciels sont rapides.

Cas de plusieurs courbes Une même inversion de matrice permet de calculer tous les coefficients

Calcul de moyenne lissée

Données on définit sur une base, lisse ou pas, $\delta_1(t), \dots, \delta_m(t)$ un ensemble de fonctions $Y_i(t) = \sum_{j=1}^{q} a_{ij}\delta_j(t)$. Leur moyenne est

$$\overline{Y}(t) = \frac{1}{n}\sum_{i=1}^{n} Y_i(t) = \sum_{j=1}^{q} \bar{a}_j \delta_j(t), \quad \text{avec } \bar{a}_j = \frac{1}{n}\sum_{i=1}^{n} a_{ij}$$

Problème on se donne une autre base $\beta_1, \dots, \beta_p(t)$ qui est lisse et on cherche une moyenne lissée $g(t) = \sum_{k=1}^{p} \xi_k \beta_k(t_j)$

Méthode des moindres carrés on minimise

$$\int [g(t) - \overline{Y}(t)]^2\, dt + \lambda \int g''(t)^2 dt = \int \overline{Y}^2(t) + \boldsymbol{\xi}'\mathbf{J}\boldsymbol{\xi} + \lambda\boldsymbol{\xi}'\mathbf{K}\boldsymbol{\xi} - 2\boldsymbol{\xi}'\mathbf{L}\bar{\mathbf{a}}$$

où $J_{jk} = \int \beta_j(t)\beta_k(t)\, dt$ et $L_{jk} = \int \delta_j(t)\beta_k(t)\, dt$. On obtient encore

$$\boldsymbol{\xi} = (\mathbf{J} + \lambda\mathbf{K})^{-1}\mathbf{L}'\bar{\mathbf{a}}$$

Partie VI. Utilisation du paquet « fda » de R

Le paquet « fda »

Origine ces fonctions ont été écrites principalement par J. O. Ramsay pour implémenter les méthodes décrites dans le livre « Functional Data Analysis ». Elles sont décrites dans

> Ramsay, J. O., Hooker, Giles, and Graves, Spencer, *Functional Data Analysis with R and Matlab*, Springer, 2009.

Fonctions elles permettent notamment de faire des approximation de fonctions par des bases finies (B-splines, Fourier...), avec des lissages sur les dérivées

Installation le paquet n'est pas disponible par défaut dans R, mais il est sur CRAN. Il suffit donc de faire

```
> install.packages("fda")
```

puis avant chaque utilisation

```
> require(fda)
```

Préambule : définition de fonction R

But La définition des fonctions, comme donnée par R dans la commande `?mafonction`, définit les paramètres obligatoires et optionnels

Exemple On considère la définition suivante

```
result <- mafonction(x, y, a=2, b=NULL)
```

Paramètres obligatoires `x` et `y`

Paramètres optionnels `a` et `b`

- si on ne donne pas `a` explicitement, sa valeur sera `2`.
- de même, b sera égal à `NULL` si il est omis.

Ce n'est pas la peine de passer ces paramètres si ils ont la valeur par défaut

Exemples

- `mafonc(1,2,3,4)` : `x=1, y=2, a=3, b=4`
- `mafonc(1,2,3)` : `x=1, y=2, a=3, b=NULL`
- `mafonc(y=2,x=99,b=3)` : `x=99, y=2, a=2, b=3`

Résultat de la fonction Il est retourné dans une variable (pas de passage par variable)

Données fonctionnelles

Fonction cette fonction n'est jamais appelée directement, mais elle crée un objet de type `fd` (donnée fonctionnelle)

```
fdobj <- fd(coef, basisobj, fdnames=NULL)
```

Paramètres ce sont les composants d'une donnée fonctionnelle

- `coef` : soit un vecteur pour une fonction, sont une matrice pour plusieurs fonctions (en colonne), soit une matrice à 3 dimensions pour des données multivariées
- `basisobj` : une base retournée par `create.*.basis()`
- `fdnames` : une liste de trois noms pour décrire : les points de mesure, les différents individus et les valeurs mesurées

Évaluation la fonction suivante évalue une fonction à une série de points

```
eval.fd(evalarg, fdobj, Lfdobj=0)
```

- `evalarg` : un vecteur ou une matrice de points où évaluer la fonction
- `fdobj` : un objet de type `fd`
- `Lfdobj` : un degré de dérivation (ou un opérateur différentiel)

Sélection l'expression `fdobj[i]` renvoie la `i`-ème donnée fonctionnelle de `fdobj`

Création d'une base de splines

Fonction pour créer une base de B-splines

```
basisobj <- create.bspline.basis(rangeval=NULL,
    nbasis=NULL, norder=4, breaks=NULL)
```

Paramètres utiles

- `breaks` : les points où les fonctions sont évaluées
- `norder` (défaut 4) : l'ordre des splines, c'est-à-dire un de plus que le degré

Paramètres secondaires

- `nbasis` : le nombre de splines dans la base, par défaut `length(breaks) + norder - 2`
- `rangeval` : l'intervalle de définition des splines, par défaut `range(breaks)`

Autres Paramètres voir la documentation en ligne

Paramètres de lissage

Fonction pour définir les paramètres de lissage passés à `toutes` les fonctions de lissage

```
fdparobj <- fdPar(
    fdobj=NULL, Lfdobj=NULL, lambda=0)
```

Paramètres utiles

- `fdobj` : on passe en général la base de spline qu'on vient de créer
- `Lfdobj` : on peut juste passer le degré de dérivation souhaité pour le lissage
  Si on ne donne pas `Lfdobj`, la valeur par défaut est `norder(fdobj) - 2`
- `lambda` : c'est le λ décrit dans le cours

Lissage des données

Fonction pour le lissage de données

```
smoothfun <- smooth.basis(argvals=1:n, y,
     fdParobj, fdnames=NULL)
```

Paramètres utiles

- `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)
- `fdParobj` : un objet produit par `fdPar()`

Résultat une liste avec les éléments suivants (entre autres)
- `fd` : les données fonctionnelles
- `gcv` : critère de validation croisée (utile pour choisir λ)

Représentation graphique

Fonction pour tracer une donnée fonctionnelle

```
plot(x, Lfdobj=0, href=TRUE, ...)
```

Paramètres voir la documentation pour plus de détails (`titles`, `xlab`, `ylab`, ...)
- `x` : données fonctionnelles à tracer
- `Lfdobj` : un degré de dérivation (ou un opérateur différentiel)
- `href` : si `TRUE`, tracer une ligne horizontale de référence en 0

Fonction pour tracer une courbe comparative des données brutes et lissées

```
plotfit(y, argvals, fdobj, residual=FALSE, ...)
```

Paramètres voir la documentation pour plus de détails (`titles`, `xlab`, `ylab`, ...)
- `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)
- `fdobj` : données fonctionnelles
- `residual` : si `TRUE`, les résidus sont représentés plutôt que les courbes

Remarque Souvent illisible, car les données brutes sont indiquées par ces cercles. Alternative : utiliser `plot()` sur les données lissées et `lines()` sur les données brutes.