TP : station de comptage de trafic (jour 2)

1 Introduction

On réutilise le jeu de données `S21.csv` provenant d'une station de comptage de trafic routier (boucle magnétique), fournie par la société SISTeMA ITS. On cherche aujourd'hui à avancer dans deux directions :

— calcul de l'ACP lissée

— gestion des valeurs manquantes

Comme précédemment, on donne ci-dessous une liste de questions pour guider le travail. On demandera de répondre de manière brève mais aussi précise que possible en donnant :

— le code utilisé,

— le résultat (données, graphiques),

— éventuellement un commentaire et/ou des explications sur ce que vous avez fait.

On partira si besoin des variables suivantes, utilisées dans la correction du TP1 (ou alors, tout autre code équivalent)

```
> # dat0 contient les donnees brutes
> dat0=as.matrix(read.table("S21.txt", head=T))
> heures=c(0:239)/10
> # Vecteur des donnees manquantes par ligne
> na_count = rowSums(is.na(dat0))
> # Quelques courbes choisies arbitrairement
> courbes=c(1, 100, 200, 300, 400)
> # Journees sans donnee manquante
> dat = dat0[na_count == 0,]
> # Si on veut reduire les grandes valeurs
> #dat=sqrt(dat)
> require(fda)
> # Representation fonctionnelle des donnees de trafic comme au TP1
> # On lisse toutes les courbes
> nderiv = 2
> norder = nderiv + 2
> fdnames=list("heure", "jour", "debit")
> basisobj = create.bspline.basis(breaks=heures, norder=norder)
> lambda=0.1
> fdParobj = fdPar(fdobj=basisobj, Lfdobj=nderiv, lambda=lambda)
> dat.fd = smooth.basis(argvals=heures, y=t(dat), fdParobj, fdnames=fdnames)$fd
```

2 ACP lissée

On cherche à faire une ACP lissée des données de trafic (uniquement les journées où toutes les données sont disponibles).

Question 1: *Faire une ACP fonctionnelle des débits. Représenter la courbe de décroissance des valeurs propres : combien d'axes semblent pouvoir être intéressants ?*

Question 2: *Représenter les 4 premiers axes principaux. Interpréter les axes. Combien faut-il en conserver ? Quelle proportion de l'inertie représentent-ils ?*

Question 3: *Est-ce que l'ACP fonctionnelle sur la racine carrée des données est plus intéressante ? Quelle différence y a-t-il ?*

Question 4: *On se donne un tableau contenant les jours de la semaine (0=dimanche, ... 6=samedi) calculé comme suit :*

```
> jours=as.numeric(format(as.Date(rownames(dat), format="%d/%m/%Y"),"%w"))
```

Réalisez pour les axes $(1, 2)$ *d'une part, puis* $(1, 3)$ *d'autre part, des graphique où chaque jour est représenté par une couleur qui dépend du jour de la semaine. Que peut-on déduire à propos du lien entre les axes et les jours de la semaine ? Que peut-on déduire en prenant en compte l'interprétation des axes ?*

3 Gestion des valeurs manquantes

On voudrait analyser aussi les journées dont le nombre de données manquantes est inférieur à 10 (mais non nul)

Question 5: *faire une sélection de ces données. Combien y en a-t-il ?*

Question 6: *La méthode à utiliser est de traiter les journées une par une en enlevant les données manquantes et les temps correspondants. Écrire le code permettant de lisser une journée avec des données manquantes.*

Question 7: *Écrire le code permettant de lisser un ensemble deDernière question journées et de mettre le résultat dans un objet fd unique. Pour cela on utilisera la fonction* `fd()` *pour créer à la main un objet fonctionnel.*

Question 8: *ajouter ce nouvel objet fonctionnel à celui qu'on avait pour les éléments sans valeurs manquantes. On tracera par exemple les courbes des moyennes des différentes données (*$\texttt{NA} = 0$, $0 < \texttt{NA} \leq 9$, $\texttt{NA} \geq 9$*) pour montrer que les résultats sont cohérents, même s'il y a des différences Expliquer pourquoi on peut maintenant faire toutes les analyses (ACP...) sur cette nouvelle donnée*