



Contexte et objectifs du travail d'Aynaz Adl Zarrabi



La presbyacousie

- Définition : perte d'audition liée à l'âge (> 50 ans)
- Effets :
 - augmentation des seuils d'audibilité
 - distortion des signaux auditifs perçus (perte de sélectivité fréquentielle, recrutement de sonie)

Contexte

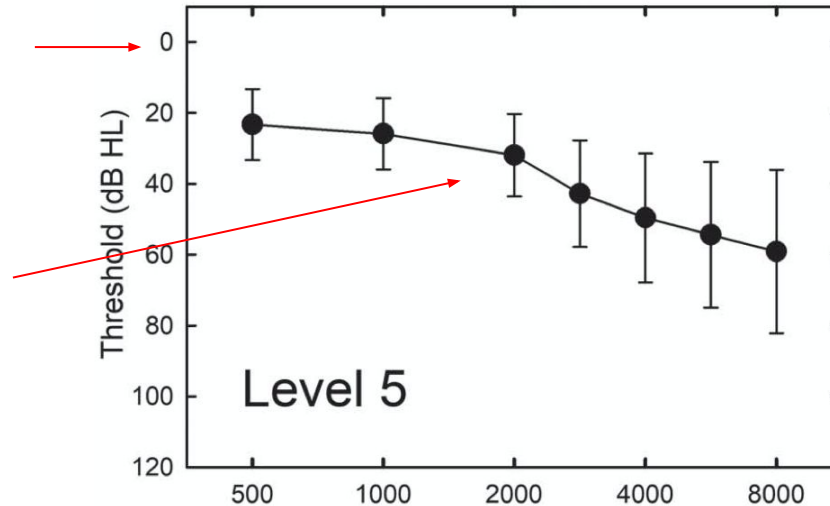


La presbyacousie

- Caractérisation de la perte auditive d'une personne en fonction de ses **seuils d'audibilité**

Oreille
« saine »

Seuils
mesurés



Audiogram moyen
pour une perte de niveau 5
(Humes et al 2021)



La presbyacousie

- Effets **néfastes sur la perception et la compréhension de la parole** (difficulté à reconnaître des mots, à comprendre des phrases)
- Ces effets sont mesurés par **des tests subjectifs de répétition de mots** (tests **d'intelligibilité**) : on demande au patient de répéter X mots, et on compte le **% de mots correctement identifiés**
- **Les tests subjectifs sont lourds (temps, fatigue du sujet), ils ne peuvent être conduits autant de fois que souhaité (ex. pour trouver le meilleur réglage d'une audioprothèse)**

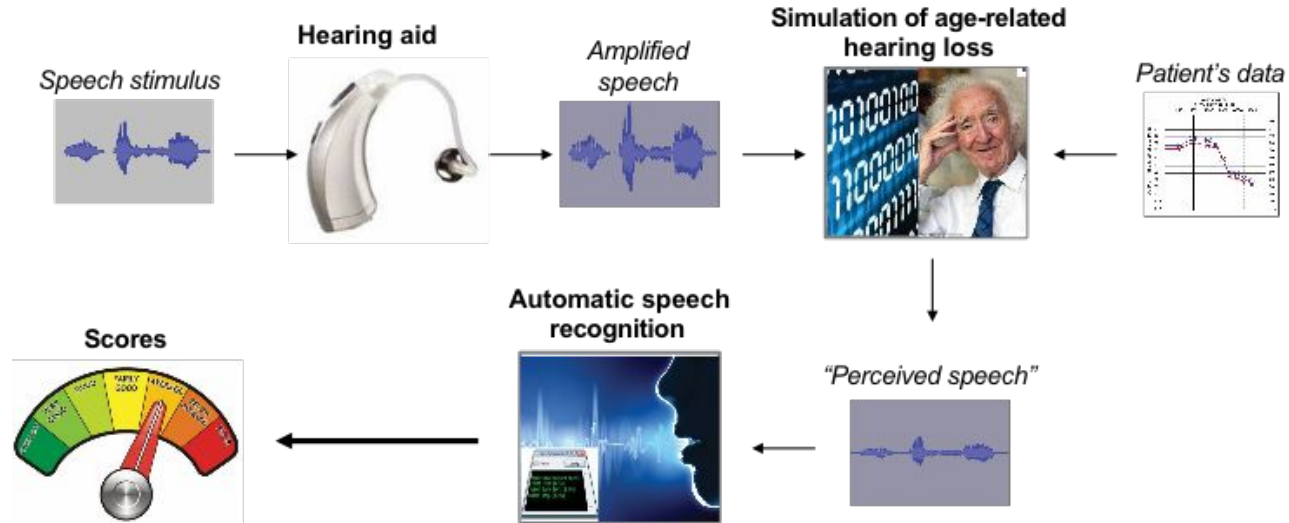


Projet CarlW (Archean Technologies & IRIT)

- **Idée** : utiliser **un simulateur de perte auditive** ET un **moteur de reconnaissance automatique de la parole** pour « remplacer le patient » dans les tests de répétition de mots

Contexte

Projet CarlW (Archean Technologies & IRIT)





Projet CarlW (Archean Technologies & IRIT)

- Nous avons fait plusieurs preuves de concept qui montrent :
 1. Qu'il y a une relation significative (corrélation positive) entre les scores automatiques obtenus via CarlW et les scores obtenus par de « vrais » patients lors de tests de répétition de mots
 2. Que CarlW peut être utilisé pour affiner les réglages audioprothétiques pour un patient donné (i.e., trouver un meilleur réglage qu'un réglage « standard »)

Objectifs à long terme



Projet CarlW (Archean Technologies & IRIT)

- Les preuves de concept effectuées jusqu'à présent ne concernent que les aspects **quantitatifs** des performances des patients en termes de perception de la parole (**VD = % de mots correctement identifiés**)
 - Question de recherche à long terme : **est-il possible de prédire aussi les erreurs de perception de la parole effectuées par les patients ?** (Ex. « Le bateau » répété « Le barreau »). Autrement dit, **passer du quantitatif au qualitatif**

Objectifs à long terme



Projet CarlW (Archean Technologies & IRIT)

- Pour effectuer cette preuve de concept, les étapes suivantes sont envisagées :
 1. Démontrer qu'il existe une relation significative entre les profils auditifs de patients (les audiogrammes) simulés et les erreurs de « perception » commises par le moteur de reconnaissance automatique de la parole
 2. Identifier des profils (audiogrammes) « typiques » pour des erreurs de perception particulières
 3. Simuler ces pertes (ces audiogrammes typiques) sur des stimuli de parole et les faire écouter à des sujets normo-entendants pour vérifier qu'ils commettent les erreurs de perception attendues

Objectifs d'Aynaz



Projet CarlW (Archean Technologies & IRIT)

- Pour effectuer cette preuve de concept, les étapes suivantes sont envisagées :
1. **Démontrer qu'il existe une relation significative entre les profils auditifs de patients (les audiogrammes) simulés et les erreurs de « perception » commises par le moteur de reconnaissance automatique de la parole**
 2. **Identifier des profils (audiogrammes) « typiques » pour des erreurs de perception particulières**
 3. Simuler ces pertes (ces audiogrammes typiques) sur des stimuli de parole et les faire écouter à des sujets normo-entendants pour vérifier qu'ils commettent les erreurs de perception attendues

Données



- 1000 audiogrammes (vecteurs de 12 seuils auditifs, de .125 à 10 kHz) ont été générés (attention, les valeurs des seuils ne sont pas totalement indépendantes les unes des autres au sein d'un même audiogrammes)
- Les 1000 audiogrammes ont été simulés sur 16 fichiers audio contenant les 16 consonnes du français entourées par la voyelle [a] (ex. [apa], [afa], etc.)
- Les 16 000 fichiers résultant de ce traitement ont été donnés à un moteur de reconnaissance automatique de la parole qui devait choisir entre les 16 possibilités ([afa], [apa], etc.)

Données



- Pour chaque audiogramme et chacun des stimuli attendus (ex. « apa »), nous avons comme information :
 - Le mot reconnu (ex. « afa »)
 - Le fait que le mot reconnu était correct ou non (0 ou 1, 0 dans notre exemple)
 - Le cas échéant, l'erreur phonétique (p → f dans notre exemple)
 - Le cas échéant, la ou les erreur(s) phonologique(s) (+continu dans notre exemple)



- **(Aparté : les traits phonologiques)**

- Les 16 consonnes du français s'opposent les unes aux autres en fonction de 8 traits phonologiques binaires (continu, nasal, voisé...).
- Certaines consonnes ne s'opposent que par un seul trait (p ↔ b : -voisé devient + voisé), et pour d'autres plusieurs traits sont différents. Donc pour une seule erreur phonétique (ex. R ↔ f) il peut y avoir plusieurs erreurs phonologiques

Données



- Pour chaque audiogramme et chacun des stimuli attendus (ex. « apa »), nous avons comme information :
 - Le mot reconnu (ex. « afa »)
 - Le fait que le mot reconnu était correct ou non (0 ou 1, 0 dans notre exemple)
 - Le cas échéant, l'erreur phonétique (p → f dans notre exemple)
 - Le cas échéant, la ou les erreur(s) phonologique(s) (+continu dans notre exemple)

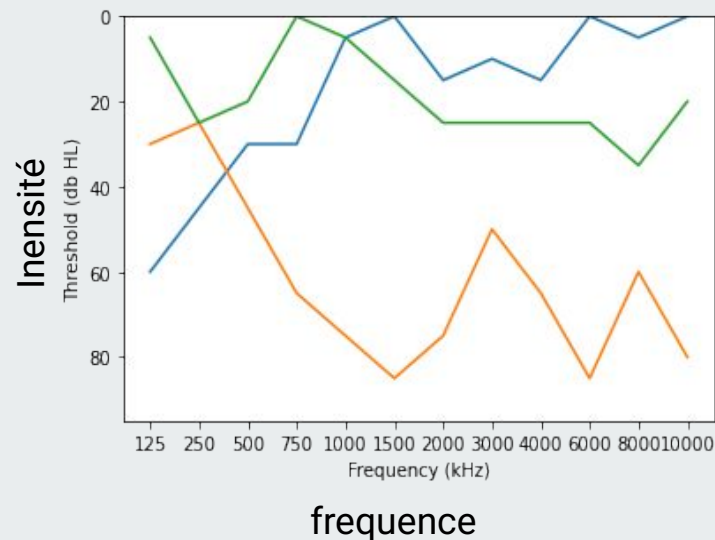
Données



- De plus, pour chaque audiogramme nous avons comme information le % de mots reconnus ($x / 16$). Ceci peut être important pour filtrer les résultats en ne conservant que les audiogrammes pour lesquels les performances du moteurs de reconnaissance étaient suffisantes (ex. au moins 50%). Sinon, la pertinence des erreurs phonétiques ou phonologiques observées peut être questionnée.

Résumé de travail sur le projet de CarlW

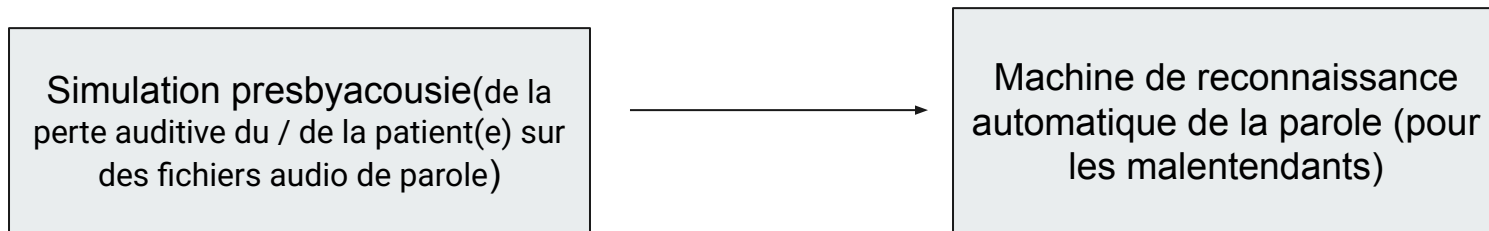
Aynaz Adl Zarrabi



Objectifs



1. La relation significative entre les audiogrammes et chaque erreur
2. Identifier l'audiogramme le plus typique pour chaque erreur (phonème cible et phonème reconnu) et chaque trait phonologique (ajouté enlevé)
3. Prendre ces audiogrammes pour mettre dans la machine de reconnaissance de la parole (qui va donner des audios de chaque mot et voir est-ce qu'on fait les mêmes erreurs de perception)



Données brutes

id_audiogramme	seuil_125	seuil_250	seuil_500	seuil_750	seuil_1000	seuil_1500	seuil_2000	seuil_3000	seuil_4000	seuil_6000	seuil_8000	seuil_10000	phoneme_cible	phoneme_reconnu	phoneme_correct	erreurs_phono	
1	1	0	0	10	0	20	25	35	55	40	20	0	15	t	R	0	[+sonant#+continuum+voise#-coronal#+haut#+arrie...
2	1	0	0	10	0	20	25	35	55	40	20	0	15	b	b	1	NaN
3	1	0	0	10	0	20	25	35	55	40	20	0	15	Z	z	0	[-haut#+anterior]
4	1	0	0	10	0	20	25	35	55	40	20	0	15	m	m	1	NaN
5	1	0	0	10	0	20	25	35	55	40	20	0	15	S	s	0	[-haut#+anterior]
...
15996	1000	90	90	90	85	65	65	50	65	55	30	30	15	p	f	0	[+continuum]
15997	1000	90	90	90	85	65	65	50	65	55	30	30	15	z	s	0	[-voise]
15998	1000	90	90	90	85	65	65	50	65	55	30	30	15	v	f	0	[-voise]
15999	1000	90	90	90	85	65	65	50	65	55	30	30	15	s	s	1	NaN
16000	1000	90	90	90	85	65	65	50	65	55	30	30	15	R	v	0	[-sonant#-haut#-arriere#+anterior]

16000 rows x 17 columns

Réalisé les différentes méthodes de preprocessing (one hot encoding, Caractéristiques des fonctionnalités) pour que les données devient analysable et lisible pour ML .

Données après les préparations



Audiogramme: numero des audiogrammes 1000
16 Logatoms(consonnes)
Frequence: 125 Hz -10000 Hz
Seuil d'audiometrique: 0-90 dBL
Audiogramme:moins d'intensité, plus de fréquence
Erreur fait: Z->b
Categorie: 16 traits phonologiques

L'idée de faire les audiogrammes moyennes de chaque ajout ou suppression de trait phonologique

frequence	+anterior	+arriere	+continu	+coronal	+haut	+nasal
0.125	17.88235294117647	15.68359375	22.587719298245613	12.125984251968504	16.41176470588235	13.076923076923077
0.25	22.84313725490196	17.08984375	29.05701754385965	16.22047244094488	18.15686274509804	14.935897435897436
0.5	26.254901960784316	16.66015625	33.70614035087719	22.322834645669293	18.31372549019608	16.794871794871796
0.75	32.25490196078432	18.6328125	39.10087719298246	29.37007874015748	20.058823529411764	20.384615384615383
1.0	36.21568627450981	19.765625	39.25438596491228	34.44881889763779	20.823529411764707	21.7948717948718
1.5	41.78431372549019	24.4140625	37.89473684210526	33.93700787401575	24.15686274509804	31.08974358974359
2.0	49.490196078431374	30.15625	39.89035087719298	31.5748031496063	28.07843137254902	40.0
3.0	52.90196078431372	29.8046875	40.43859649122807	28.89763779527559	26.529411764705884	44.23076923076923
4.0	54.45098039215687	31.66015625	41.97368421052632	27.55905511811024	28.647058823529413	48.14102564102564
6.0	51.588235294117645	30.8984375	40.76754385964912	24.33070866141732	28.58823529411765	44.80769230769231
8.0	49.372549019607845	33.18359375	39.42982456140351	26.496062992125985	31.19607843137255	46.7948717948718
10.0	48.74509803921568	33.61328125	38.70614035087719	29.566929133858267	31.686274509803923	47.11538461538461

Pour trouver les audiogrammes plus typiques de chaque erreurs on a regardé aussi des moyennes de fréquences, qu'il ne donnent pas les bonnes résultats.

Filtrage des données

```
df_pourcentage=pourcentage.loc[(pourcentage['pourcentage'] >50)& (pourcentage['phoneme_correct'] == '1') ]  
df_pourcentage
```

	id_audiogramme	phoneme_correct	pourcentage
4	10	1	58.333333
6	100	1	83.333333
10	102	1	91.666667
18	106	1	66.666667
20	107	1	58.333333
...
1694	90	1	66.666667
1712	91	1	58.333333
1764	94	1	91.666667
1785	95	1	75.000000
1857	99	1	66.666667

Création une base de données comme la base de données de Phonème


	id_audiogramme	0.125	0.25	0.5	0.75	1.0	1.5	2.0	3.0	4.0	6.0	8.0	10.0	mot
1	123	10.0	20.0	15.0	25.0	25.0	25.0	0.0	0.0	10.0	25.0	25.0	25.0	RS
2	26	0.0	10.0	20.0	30.0	30.0	20.0	0.0	5.0	10.0	30.0	15.0	25.0	RS
3	275	25.0	10.0	25.0	30.0	35.0	25.0	0.0	15.0	10.0	20.0	40.0	55.0	RS
4	281	25.0	15.0	30.0	20.0	30.0	30.0	5.0	0.0	10.0	30.0	15.0	10.0	RS
5	600	5.0	15.0	10.0	15.0	35.0	20.0	5.0	0.0	20.0	15.0	10.0	25.0	RS
...
645	99	10.0	0.0	15.0	0.0	20.0	5.0	10.0	30.0	25.0	0.0	15.0	20.0	nm
646	148	10.0	30.0	40.0	50.0	30.0	30.0	20.0	35.0	35.0	50.0	30.0	45.0	nv
647	338	30.0	20.0	40.0	35.0	35.0	10.0	10.0	25.0	0.0	5.0	10.0	15.0	nz
648	626	5.0	25.0	20.0	25.0	25.0	5.0	15.0	25.0	5.0	0.0	10.0	10.0	nz
649	388	35.0	15.0	30.0	45.0	40.0	50.0	25.0	45.0	60.0	70.0	75.0	65.0	pf

Jeu de données déséquilibrés

(taille de chaque classe d'erreur dans l'ensemble d'audiogrammes)

bp	72	db	7
dt	70	bd	6
nm	60	ln	6
fp	51	ng	5
kp	30	RS	5
dp	29	Sf	5
mb	26	dz	4
mn	25	bz	4
ml	20	nd	4
bv	19	mz	4
lm	18	fS	3
fR	18	Sk	3
dg	15	bg	3
Rf	13	dR	3
nb	13	dv	3
gk	12	Rp	2
bf	11	mf	2
dk	10	dm	2
Rs	10	mv	2
bR	9	nz	2
fk	9		
df	8		
ZS	8		
fs	7		

Oversampling
(knn)



id_audiogramme	
mot	
RS	72
Rf	72
Rp	72
Rs	72
Sf	72
Sk	72
ZS	72
bR	72
bd	72
bf	72
bg	72
bp	72
bv	72
bz	72

Les variables explicative et à expliquer

Variable à expliquer,Y

	0.125	0.25	0.5	0.75	1.0	1.5	2.0	3.0	4.0	6.0	8.0	10.0
1	10	20	15	25	25	25	0	0	10	25	25	25
2	0	10	20	30	30	20	0	5	10	30	15	25
3	25	10	25	30	35	25	0	15	10	20	40	55
4	25	15	30	20	30	30	5	0	10	30	15	10
5	5	15	10	15	35	20	5	0	20	15	10	25
6	10	15	30	20	30	40	15	25	40	55	65	45
7	10	15	30	40	35	30	30	50	45	60	60	60
8	10	5	25	30	35	50	35	30	45	55	50	70
9	20	10	20	10	25	20	0	15	35	15	10	10
10	0	10	20	35	45	50	30	40	45	25	15	10
11	30	15	20	20	40	35	15	30	45	40	60	55
12	30	15	25	20	35	40	20	25	35	25	10	0
13	35	10	30	25	40	40	15	25	40	20	10	15
14	35	15	30	45	40	50	25	45	60	70	75	65
15	35	20	20	35	55	30	20	30	35	20	30	20
16	5	20	30	35	55	55	40	30	45	35	55	45
17	0	20	40	35	50	30	50	40	30	20	35	30

Variable explicative,X

id_audiogramme	mot
9	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd
95	bd

Variable à expliquer → Y → les audiogrammes (fonctionnelles)
 Variable explicative → X → les groupes d'erreur

Code R pour construire un fdata



```
seuil<-as.matrix(df) #absorp
```

```
row.names(seuil) = c(1:163)
```

```
dim(seuil) #453 12
```

```
frequence<- c(0.125,0.250,0.500,0.750,1.000,1.500,2.000,3.000,4.000,6.000,8.000,10.000) #tt
```

```
rangfreq<-c(0.125,10.000)
```

```
rangfreq
```

```
main = c("audiogram curves")
```

```
xlab = c("frequence")
```

```
ylab = c("seuil")
```

```
names=list(main='audiogram curves',xlab='frequence',ylab='seuil')
```

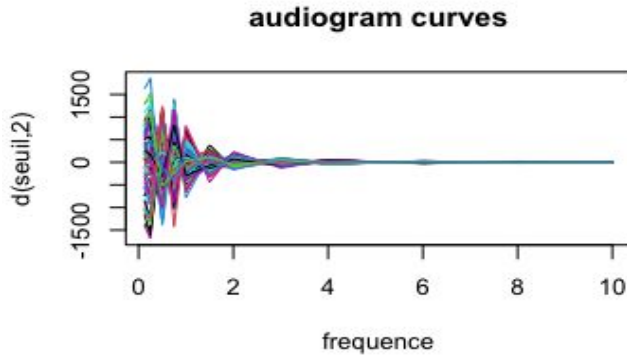
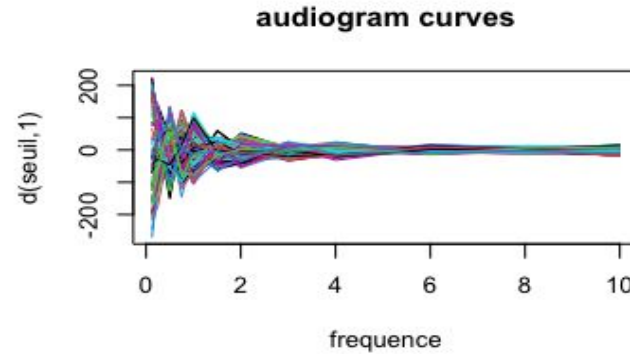
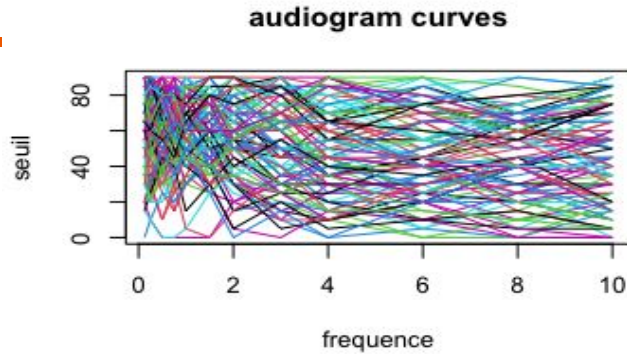
```
datacarlw=fdata(seuil, argvals = frequence,  
rangeval = rangfreq, names = names, fdata2d =  
TRUE)
```

```
absorp1<-datacarlw
```

```
tt = absorp1[["argvals"]]
```

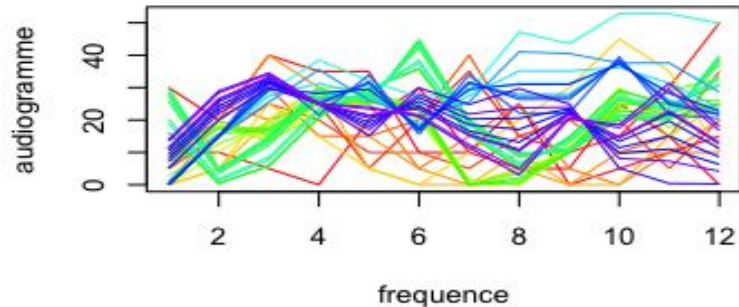
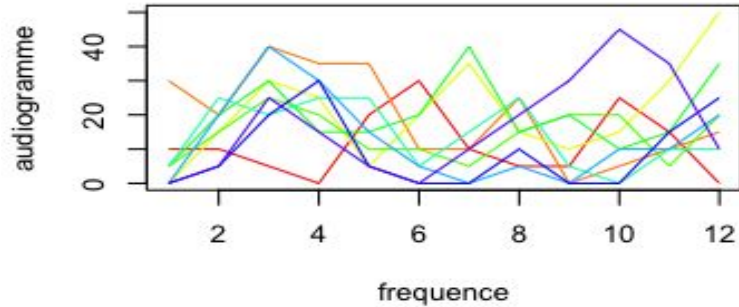
```
X = absorp1[ind, ]
```

Plots avec fdata.deriv pour toutes les erreurs



```
X.d1 = fdata.deriv(X, nbasis = 50, nderiv = 1)
X.d2 = fdata.deriv(X, nbasis = 50, nderiv = 2)
plot(X)
plot(X.d1)
plot(X.d2)
```

Plot fds pour une erreur (b -->R)



```
df_org <- read.csv("pivot.csv",check.names = FALSE)
df_org=df_org[,-1]
x=c(1:12)
bR=df_over[df_org$mot == "bR",]
bR=bR[,-13]
bR=transpose(bR)
colnames(bR)=c(1:45)
bR_fds=fds(x,bR,xname =
"frequence",yname="audiogramme")
plot(bR_fds)
```

Visualisation pour chaque trait (voise +) avec l'aide de fda et sampling

frequence	103	104	105	111	117	12	122	125	128	131	132	133	140	143	144	145
0.125	10	10	10	10	10	0	10	10	10	10	10	10	10	10	10	10
0.250	10	10	10	15	15	0	20	20	20	20	20	20	25	30	30	30
0.500	20	20	20	20	35	5	15	20	25	30	30	40	45	15	15	30
0.750	10	20	5	10	45	0	15	20	0	5	5	35	45	0	25	15
1.000	0	5	0	5	50	10	5	5	5	15	5	20	65	10	25	5
1.500	10	15	20	0	35	10	15	10	15	35	10	30	50	5	20	10
2.000	5	10	35	0	40	30	30	10	15	55	15	30	60	10	35	30
3.000	25	30	45	10	40	25	5	15	5	30	35	5	55	25	15	30
4.000	35	45	45	25	35	10	0	0	10	50	45	0	65	5	0	10
6.000	30	60	55	25	10	10	15	10	0	50	25	20	85	5	15	30
8.000	35	60	30	5	10	0	20	25	0	60	0	25	60	15	10	45
10.000	20	60	15	20	30	10	5	35	10	55	20	25	40	10	20	20

