Projet Traitement Signaux et Images

Paul Le Bretons

1. Introduction et contexte

Ce projet a pour objectif d'analyser l'évolution des prénoms les plus donnés à la naissance, en France entre 1980 et 2023.
À partir des données publiques de l'INSEE (https://catalogue-donnees.insee.fr/fr/catalogue/recherche/DS_PRENOMS), ce sont les 20 prénoms les plus populaires (à partir du top 10 masculins et du top 10 féminins) qui ont été sélectionné pour examiner les tendances qui caractérisent la popularité de ces prénoms au fil des années.
En utilisant des techniques d'analyse de données et de modélisation fonctionnelle, je vais essayer de chercher et identifier les tendances, les variations et les éventuelles périodes significatives dans les fréquences d'attribution.

2. Préparation des données

```
data <- read.csv("DS_PRENOMS_data.csv",
                 header = TRUE,
                 sep = ";",
                 stringsAsFactors = FALSE) |>
  filter(!is.na(YOB) & grepl("^[0-9]+$", YOB)) |>
  mutate(
    OBS_VALUE = as.integer(gsub(",", "", OBS_VALUE)),
    YOB = as.integer(YOB)
  ) |>
  filter(GEO_OBJECT == "FRANCE",
         PRENOM != "PRENOMS_RARES",
         YOB >= 1980)


top10_masculins <- data |>
  filter(SEX == 1) |>
  group_by(PRENOM) |>
  summarise(total_occurrences = sum(OBS_VALUE, na.rm = TRUE),
            .groups = "drop") |>
  arrange(desc(total_occurrences)) |>
  slice_max(total_occurrences, n = 10) |>
  ungroup()


top10_feminins <- data |>
  filter(SEX == 2) |>
  group_by(PRENOM) |>
```

```
  summarise(total_occurrences = sum(OBS_VALUE, na.rm = TRUE),
            .groups = "drop") |>
  arrange(desc(total_occurrences)) |>
  slice_max(total_occurrences, n = 10) |>
  ungroup()


data <- data |>
  filter((PRENOM %in% top10_masculins$PRENOM & SEX == 1) |
           (PRENOM %in% top10_feminins$PRENOM & SEX == 2)) |>
  mutate(PRENOM = gsub("_\\d+$", "", PRENOM)) |>
  dplyr::select(YOB, SEX, PRENOM, OBS_VALUE)

write.csv(data, file = 'data_clean.csv', row.names = FALSE)
```

Pour la préparation des données, j'ai décidé de me limiter aux enregistrements concernant la France et aux prénoms relevés de 1980 à fin 2023.
Le choix de prendre les données à partir de 1980 permet de se concentrer sur les tendances récentes de la popularité des prénoms, en excluant les prénoms qui ont peut-être eu une popularité passée et qui ne sont plus du tout représentatifs des tendances actuelles.

```
data = read.csv("data_clean.csv",
                header = TRUE,
                sep = ",",
                stringsAsFactors = FALSE)

data$SEX <- factor(data$SEX)

head(data)
```

```
##    YOB SEX    PRENOM OBS_VALUE
## 1 2014   1 ALEXANDRE      1818
## 2 1984   1 ALEXANDRE      6073
## 3 2012   1 ALEXANDRE      2105
## 4 1982   1 ALEXANDRE      7197
## 5 1981   1 ALEXANDRE      7027
## 6 2020   1 ALEXANDRE      1040
```

```
summary(data)
```

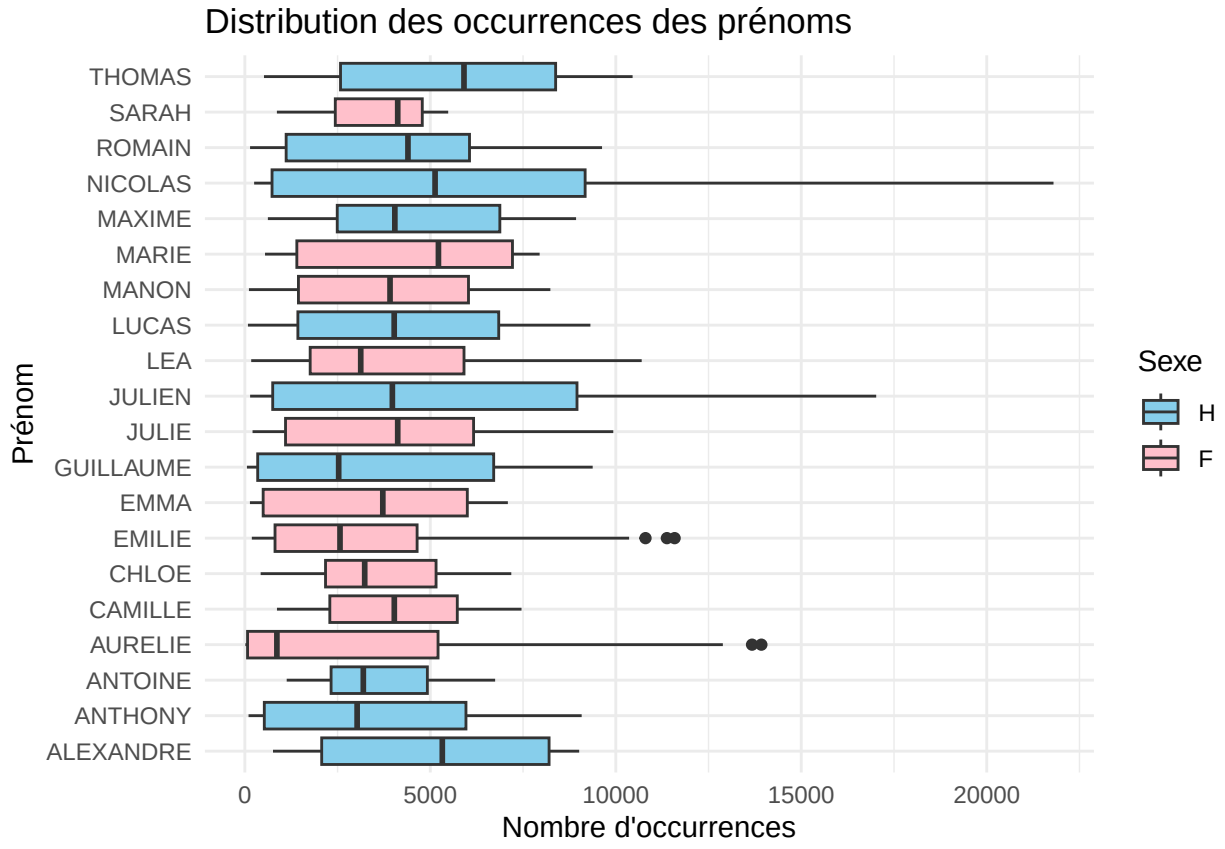
```
##       YOB       SEX        PRENOM            OBS_VALUE
##  Min.   :1980   1:440   Length:880         Min.   :   14
##  1st Qu.:1991   2:440   Class :character   1st Qu.: 1334
##  Median :2002           Mode  :character   Median : 3680
##  Mean   :2002                              Mean   : 4196
##  3rd Qu.:2012                              3rd Qu.: 6448
##  Max.   :2023                              Max.   :21800
```

On obtient un dataframe avec les informations sur les prénoms les plus donnés en France, de 1980 à 2023. Il se compose des colonnes suivantes :
- YOB : année de naissance, indiquant l'année pour laquelle les données de prénom sont comptabilisées. - SEX : facteur représentant le sexe du prénom (1 pour les prénoms masculins, 2 pour les féminins). - PRENOM : chaîne de caractères indiquant le prénom. - OBS_value : entier représentant le nombre de fois où ce prénom a été donné, pour l'année correspondante.

3. Analyse descriptive

3.1 Boxplots des occurrences des prénoms

```
ggplot(data, aes(x = PRENOM,
                 y = OBS_VALUE,
                 fill = factor(SEX))) +
  geom_boxplot() +
  labs(title = "Distribution des occurrences des prénoms",
       x = "Prénom",
       y = "Nombre d'occurrences",
       fill = "Sexe") +
  scale_fill_manual(values = c("skyblue", "pink"),
                    labels = c("H", "F")) +
  theme_minimal() +
  coord_flip()
```



Ces boxplots nous révèlent que :

- Les médianes suggèrent une certaine homogénéité dans la popularité de base de ces prénoms les plus donnés, indiquant un “seuil” de popularité relativement stable pour les prénoms très utilisés.
- Nicolas et Julien montrent les plus fortes variations de popularité sur la période, ce qui suggère une forte sensibilité aux effets de mode.
- Sarah, Antoine et Chloé présentent des distributions plus compactes, indiquant une popularité plus stable dans le temps, moins soumise aux effets de tendance.

- Aurélie et Émilie se distinguent par des points extrêmes élevés, suggérant des pics de popularité exceptionnels à certaines périodes.

3.2 Top 5 des prénoms les plus donnés

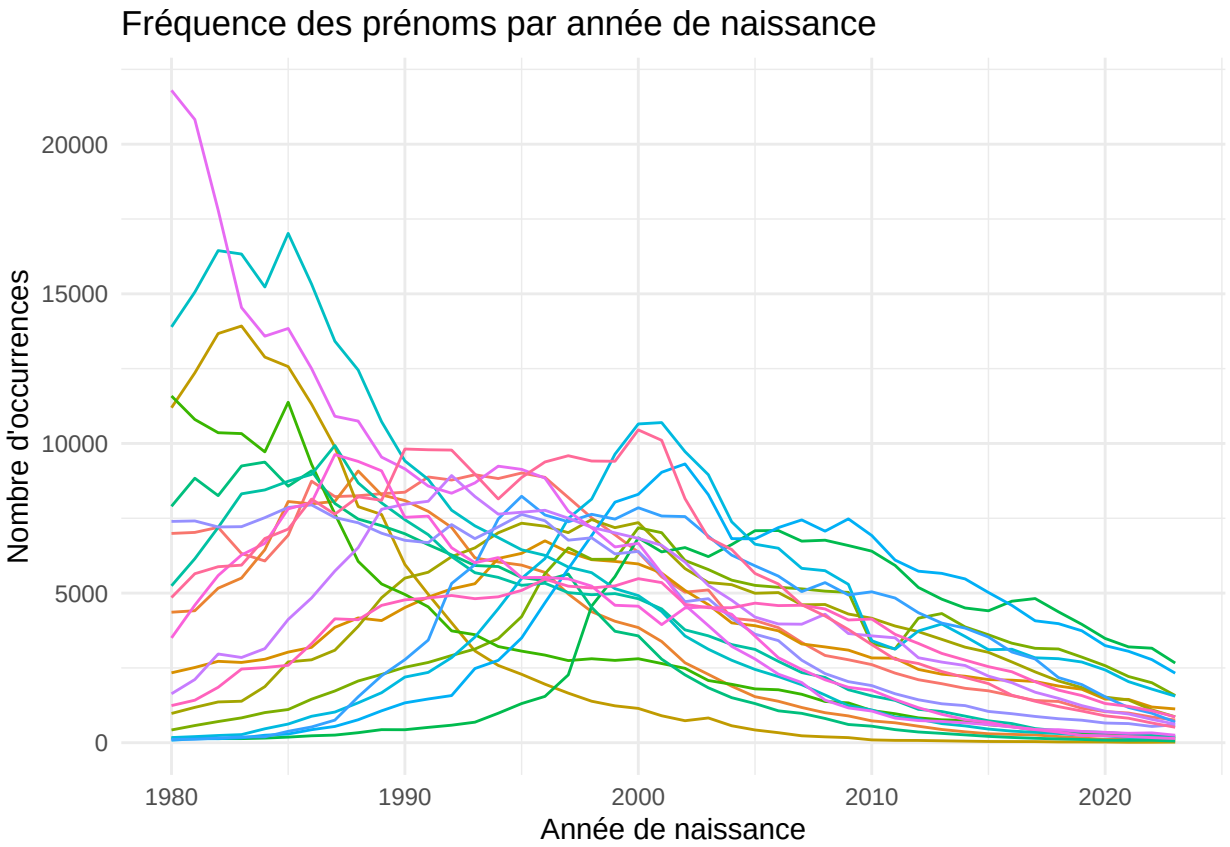
```
data |>
  group_by(PRENOM) |>
  summarise(total_occurrences = sum(OBS_VALUE, na.rm = TRUE)) |>
  arrange(desc(total_occurrences)) |>
  slice_head(n = 5)
```

```
## # A tibble: 5 x 2
##   PRENOM    total_occurrences
##   <chr>                 <int>
## 1 NICOLAS              270876
## 2 THOMAS               250521
## 3 JULIEN               249985
## 4 ALEXANDRE            223330
## 5 MARIE                198438
```

Parmi les 20 prénoms les plus donnés en France entre 1980 et 2023, Nicolas arrive en tête avec plus de 270 000 naissances, suivi de près par Thomas et Julien, montrant une préférence marquée pour ces prénoms masculins traditionnels français.

2.3 Courbes de fréquences

```
ggplot(data, aes(x = YOB,
                 y = OBS_VALUE,
                 color = PRENOM,
                 group = PRENOM)) +
  geom_line() +
  labs(title = "Fréquence des prénoms par année de naissance",
       x = "Année de naissance",
       y = "Nombre d'occurrences") +
  theme_minimal() +
  theme(legend.position = "none")
```



Ce graphique montre plusieurs choses :

- Une tendance générale à la baisse : la plupart de ces prénoms montrent une diminution prgressive de leur fréquence sur la période étudiée.
- Des pics de popularité distincts : certains prénoms ont connu des pics marqués de popularité, notamment au début des années 1980 où certains atteignaient plus de 10 000 attributions par an, puis dans les années 2000 pour d'autres prénoms.
- Différents profils d'évolution : certains prénoms montrent un déclin brutal après avoir atteint leur pic de popularité alors que d'autres suivent une diminution plus progressive. Quelques prénoms présentent quant à eux une longue période de stabilité.
- Une convergence vers le bas : à partir de 2010, on observe une convergence des courbes vers des valeurs plus faibles.

En résumé, ce graphique illustre clairement une tendance à la diversification des prénoms en France, avec une diminution générale de la concentration sur les prénoms les plus populaires au profit d'une plus grande variété de choix.

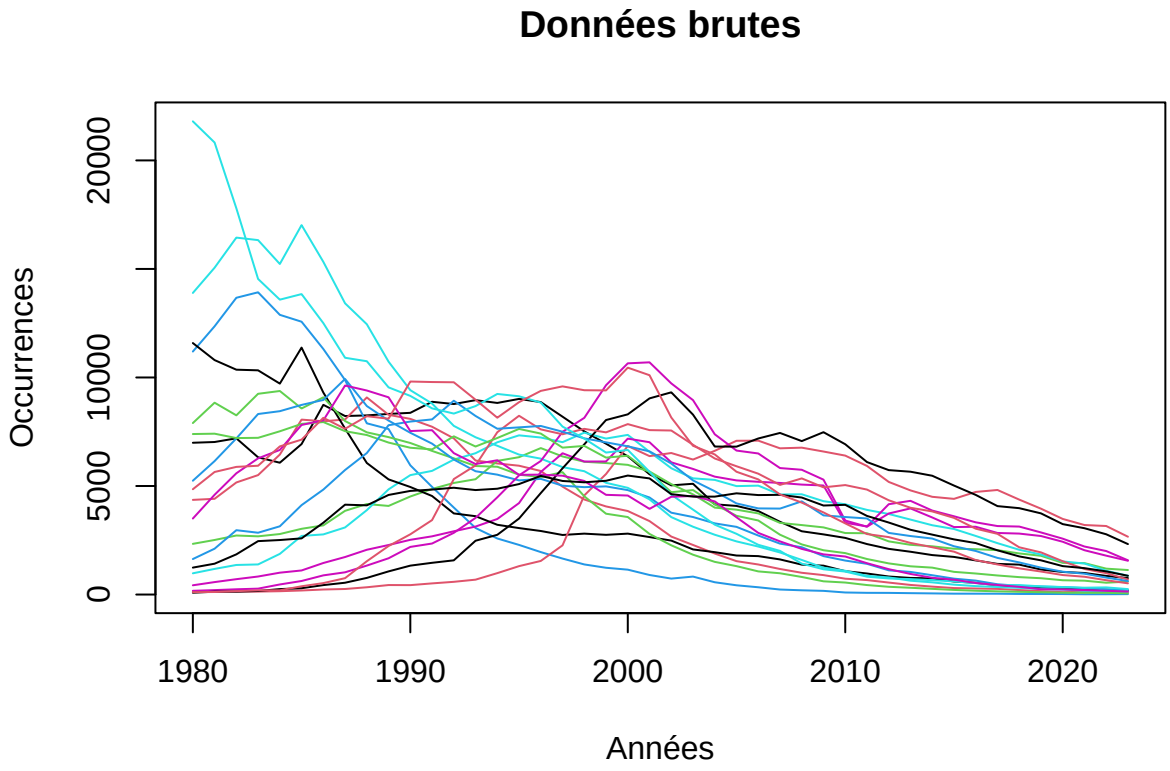
4. Lissage des Données et Création d'Objets Fonctionnels

Pour analyser les tendances à long terme dans la popularité des prénoms, je vais utiliser deux méthodes de lissage : les bases d'ondelettes et les bases de splines. Ces méthodes sont les mieux adaptées à mes données car elles vont permettre de capturer les variations naturelles dans le temps et vont offrir une flexibilité pour modéliser les changements progressifs et les tendances.

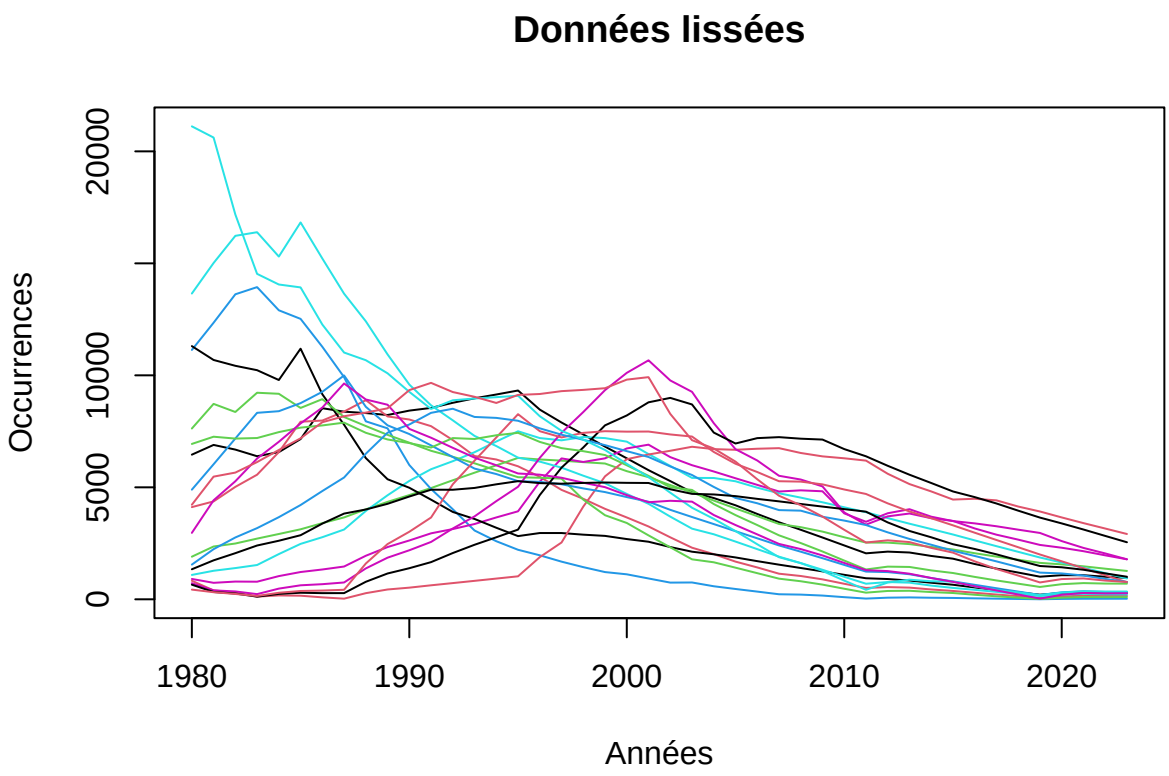
Les bases de Fourier ne seront pas explorées, car les données ne présentent pas de périodicité ou de cycles réguliers.

4.1 Avec les Bases d'Ondelettes

```
occur_by_year_matrix <- data |>
  group_by(YOB, PRENOM) |>
  summarise(occur = sum(OBS_VALUE, na.rm = TRUE),
            .groups = 'drop') |>
  spread(key = PRENOM,
         value = occur,
         fill = 0) |>
  dplyr::select(-YOB) |>
  as.matrix()

annees <- seq(1980, 2023, by = 1)

wavelet_smoothing <- function(data,
                              filter.number = 2,
                              family = "DaubExPhase") {
  n <- length(data)
  next_power <- 2^ceiling(log2(n))
  padded_data <- c(data,
                   rep(data[n],
                       next_power - n))

  wt <- wd(padded_data,
           filter.number = filter.number,
           family = family)

  wt_threshold <- threshold(wt,
                            policy = "universal",
                            value = 0.5)

  smoothed <- wr(wt_threshold)

  return(smoothed[1:n])
}

wavelets_matrix <- matrix(0,
                          nrow = length(annees),
                          ncol = ncol(occur_by_year_matrix))
for(i in 1:ncol(occur_by_year_matrix)) {
  wavelets_matrix[, i] <- wavelet_smoothing(occur_by_year_matrix[, i])
}

matplot(annees, occur_by_year_matrix,
        type = "l",
        main = "Données brutes",
        xlab = "Années",
        ylab = "Occurrences",
        lty=1)
```




```
matplot(annees, wavelets_matrix,
        type = "l",
        main = "Données lissées",
        xlab = "Années",
        ylab = "Occurrences",
        lty=1)
```



Les données lissées avec les bases d'ondelettes semblent suivre de très près les données brutes, ce qui pourrait indiquer que le lissage ne réduit pas suffisamment le bruit.

C'est pourquoi, je vais essayer d'utiliser les bases de splines qui pourraient être plus adaptées. En effet, elles sont particulièrement efficaces pour les données temporelles présentant des variations continues et progressives. Elles offrent aussi une flexibilité qui permet de modéliser les changements graduels dans les tendances.

4.2 Avec les Bases de Splines

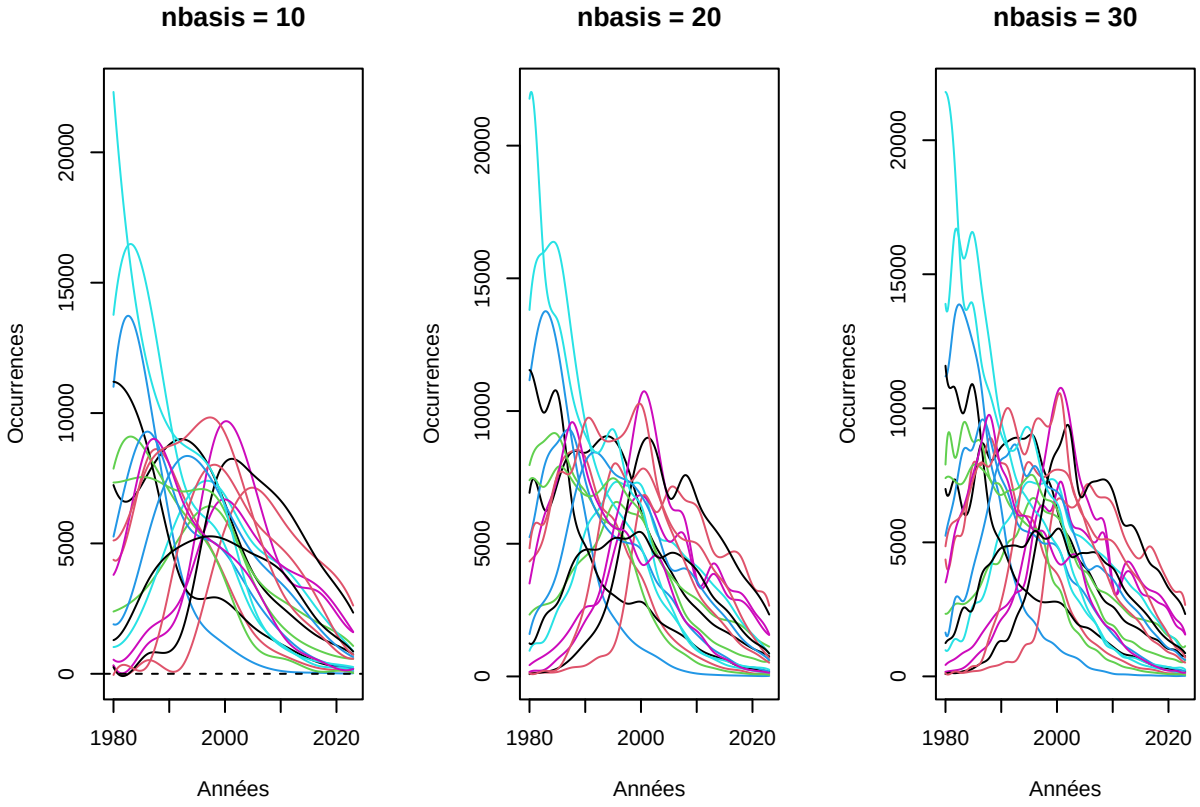
Je vais appliquer un lissage par moindres carrés non pénalisés, car les données suivent une tendance générale similaire, permettant ainsi de préserver les fluctuations naturelles et les variations locales.

Je vais tester différentes combinaisons de paramètres nbasis et lambda, afin de choisir le meilleur compromis pour le lissage.

4.2.1 Choix du nombre de bases optimal

```
annees <- seq(1980, 2023, by = 1)

occur_by_year_matrix <- data |>
  group_by(YOB, PRENOM) |>
  summarise(occur = sum(OBS_VALUE, na.rm = TRUE),
            .groups = 'drop') |>
  spread(key = PRENOM,
```

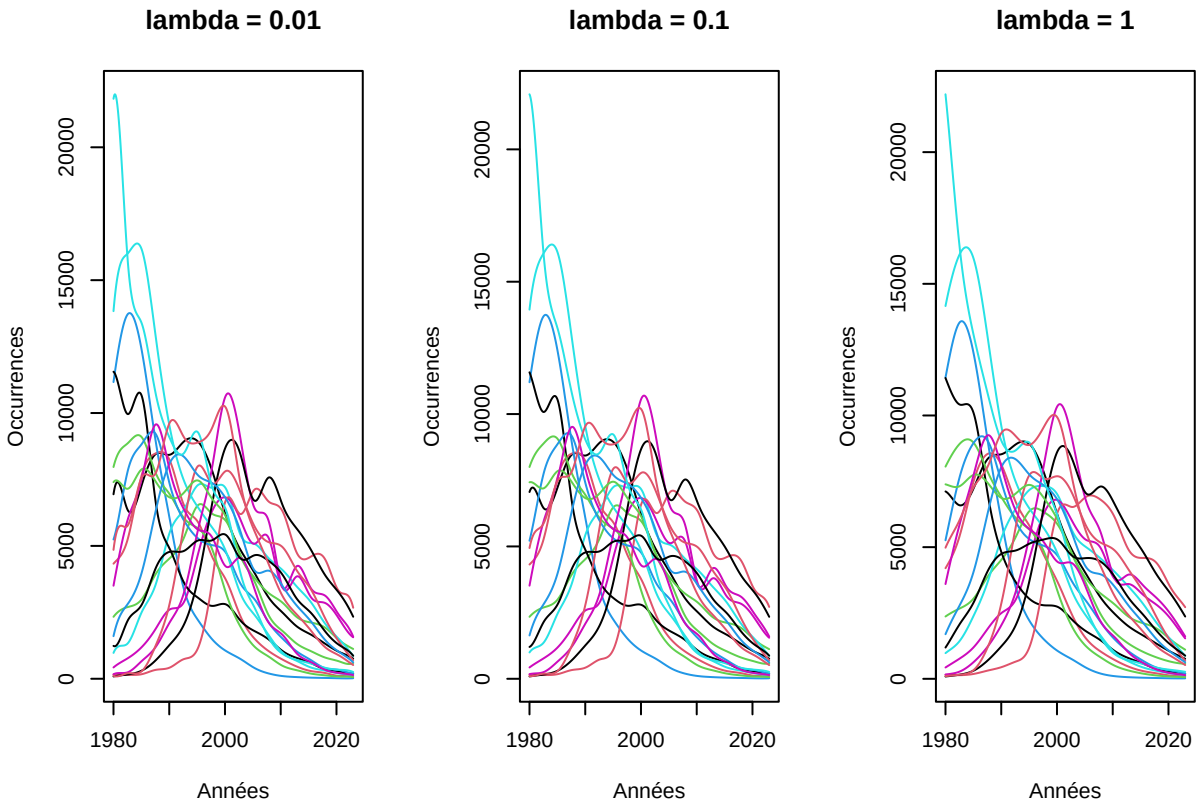
```
          value = occur,
          fill = 0) |>
  dplyr::select(-YOB) |>
  as.matrix()

nbasis_values <- c(10, 20, 30)

par(mfrow = c(1, length(nbasis_values)))

for (nbasis in nbasis_values) {
  basis_spline <- create.bspline.basis(rangeval = c(min(annees),
                                                    max(annees)),
                                       nbasis = nbasis)

  fdPar_spline <- fdPar(basis_spline,
                        Lfdobj = 2)
  fd_spline <- smooth.basis(argvals = annees,
                            y = occur_by_year_matrix,
                            fdParobj = fdPar_spline)

  plot(fd_spline$fd, main = paste("nbasis =", nbasis),
       xlab = "Années",
       ylab = "Occurrences",
       lty = 1)
}
```



Au vu des trois graphiques, nbasis = 20 semble offrir le meilleur compromis entre lissage et fidélité aux données, évitant à la fois le sur-lissage observé avec nbasis = 10 et la complexité excessive de nbasis = 30.

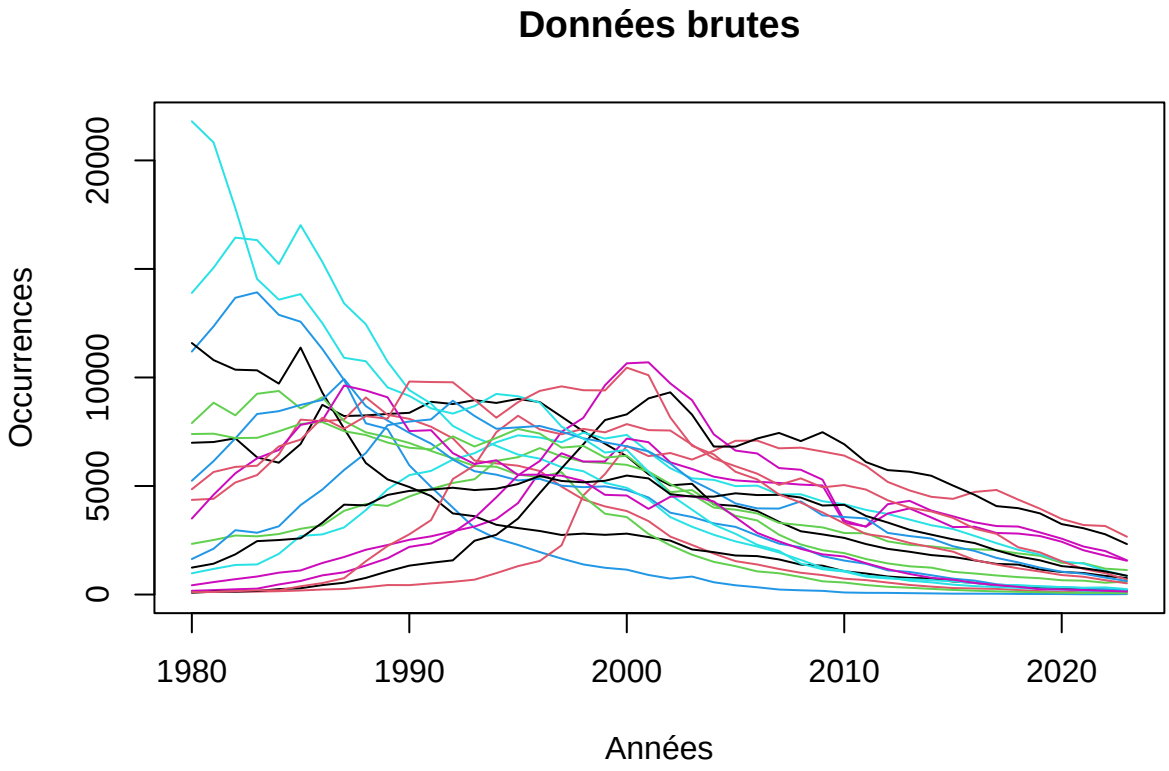
4.2.2 Choix du lambda optimal

```
lambda_values <- c(0.01, 0.1, 1)

par(mfrow = c(1, length(lambda_values)))

for (lambda in lambda_values) {
  basis_spline <- create.bspline.basis(rangeval = c(min(annees),
                                                    max(annees)),
                                       nbasis = 20)

  fdPar_spline <- fdPar(basis_spline,
                        Lfdobj = 2,
                        lambda = lambda)
  fd_spline <- smooth.basis(argvals = annees,
                            y = occur_by_year_matrix,
                            fdParobj = fdPar_spline)

  plot(fd_spline$fd, main = paste("lambda =", round(lambda, 2)),
       xlab = "Années",
       ylab = "Occurrences",
       lty = 1)
}
```

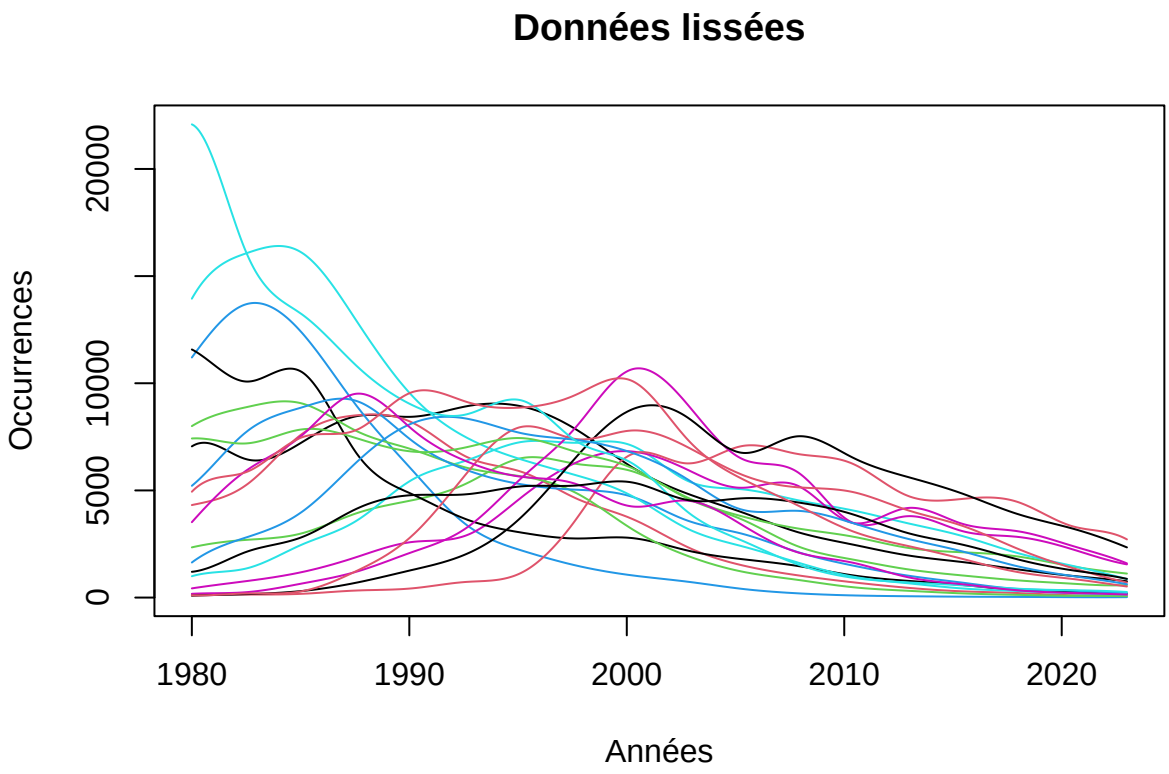


Après comparaison des trois valeurs de lambda, il n'y a pas de différence significative. Cependant, le paramètre lambda = 0.1 apparaît comme le plus approprié car c'est celui qui offre un équilibre optimal entre la régularisation et la préservation des caractéristiques importantes des courbes.

```
nbasis <- 20
lambda <- 0.1

basis_spline <- create.bspline.basis(rangeval = c(min(annees),
                                                  max(annees)),
                                     nbasis = nbasis)

fdPar_spline <- fdPar(basis_spline,
                      Lfdobj = 2,
                      lambda = lambda)
fd_spline <- smooth.basis(argvals = annees,
                          y = occur_by_year_matrix,
                          fdParobj = fdPar_spline)

matplot(annees, occur_by_year_matrix,
        type = "l",
        main = "Données brutes",
        xlab = "Années",
        ylab = "Occurrences",
        lty = 1)
```




```
plot(fd_spline$fd,
     main = "Données lissées",
     xlab = "Années",
     ylab = "Occurrences",
     lty = 1)
```



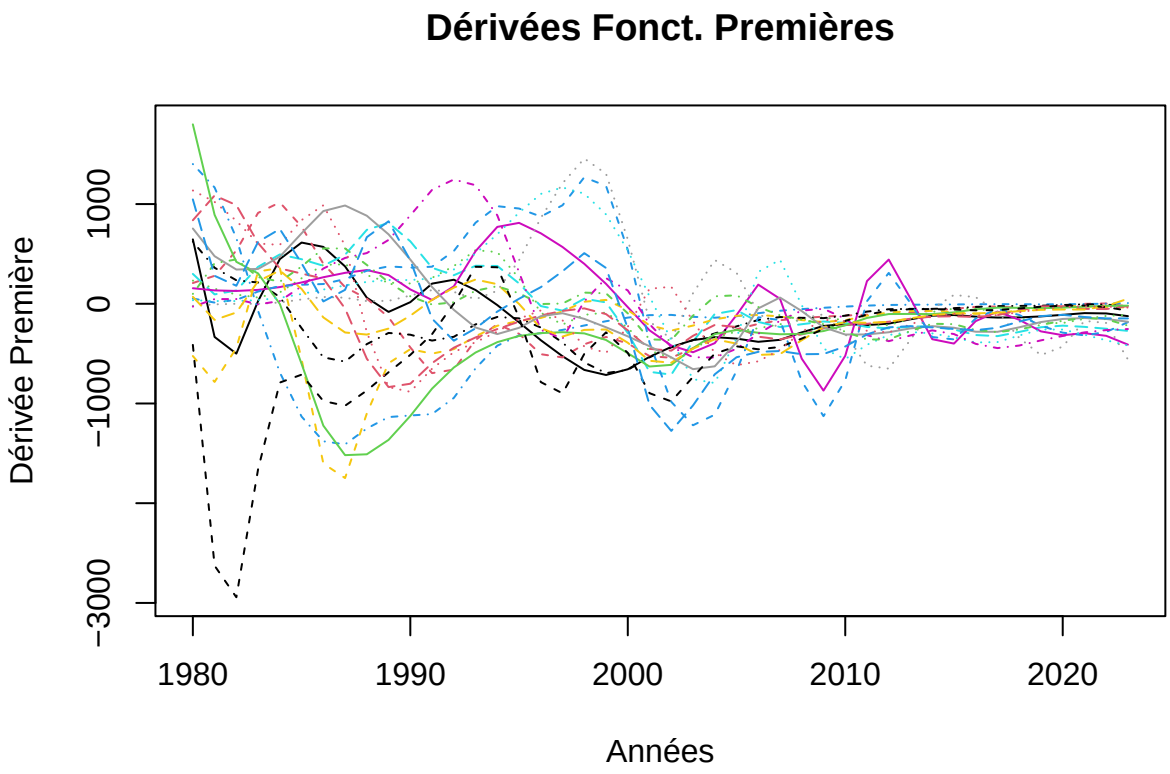

```
## [1] "done"
```

Le lissage par bases de splines avec nbasis = 20 et lambda = 0.1 offre un compromis optimal pour notre analyse, permettant de conserver les tendances significatives tout en réduisant efficacement le bruit dans les données.
Je vais donc utiliser ces données fonctionnelles lissées, qui facilitent l'identification des changements majeurs dans la popularité des prénoms, pour la suite de l'étude.

5. Statistique Exploratoire

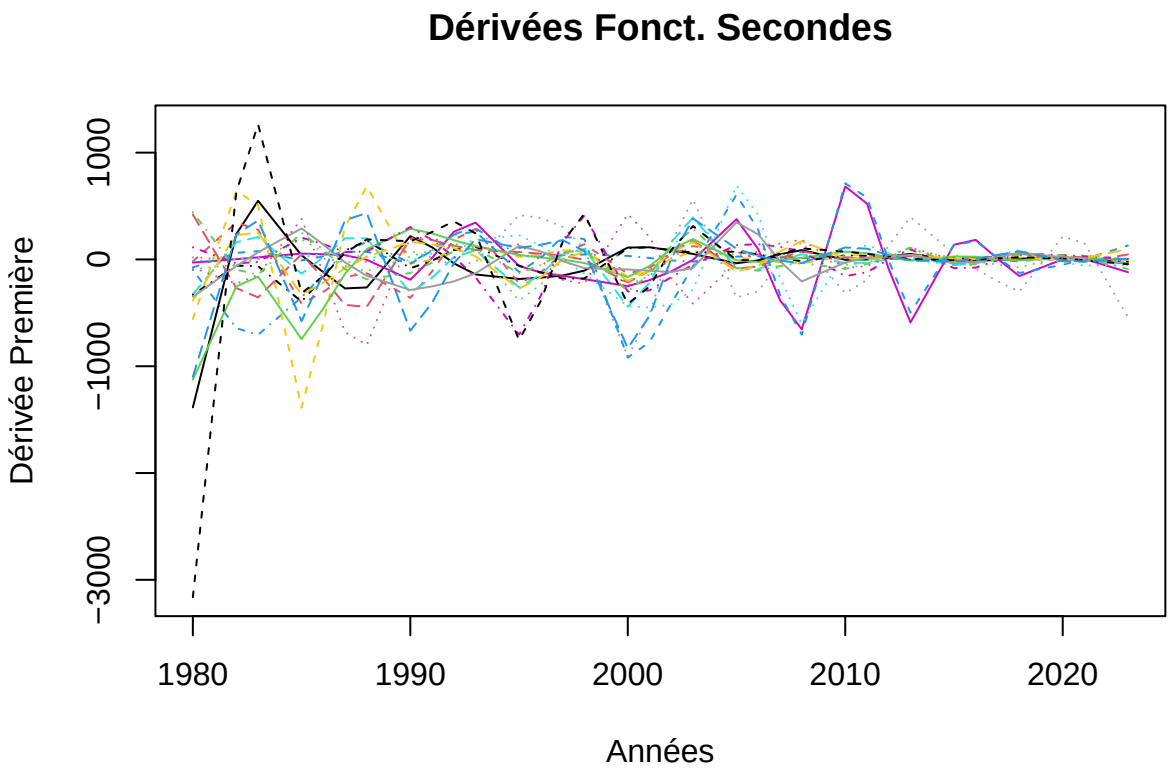
5.1 Dérivée première et seconde des données fonctionnelles

```
fd_deriv1 <- eval.fd(annees,
                     fd_spline$fd,
                     Lfdobj = 1)

matplot(annees, fd_deriv1,
        type = "l",
        col = 1:ncol(fd_deriv1),
        xlab = "Années",
        ylab = "Dérivée Première",
        main = "Dérivées Fonct. Premières")
```



Les dérivées premières montrent l'évolution des tendances générales, on peut y voir des creux et des pics de popularité significatifs au fil des années pour certains prénoms. La convergence progressive des courbes vers la fin de la période étudiée indique une stabilisation relative de la popularité de ces prénoms.

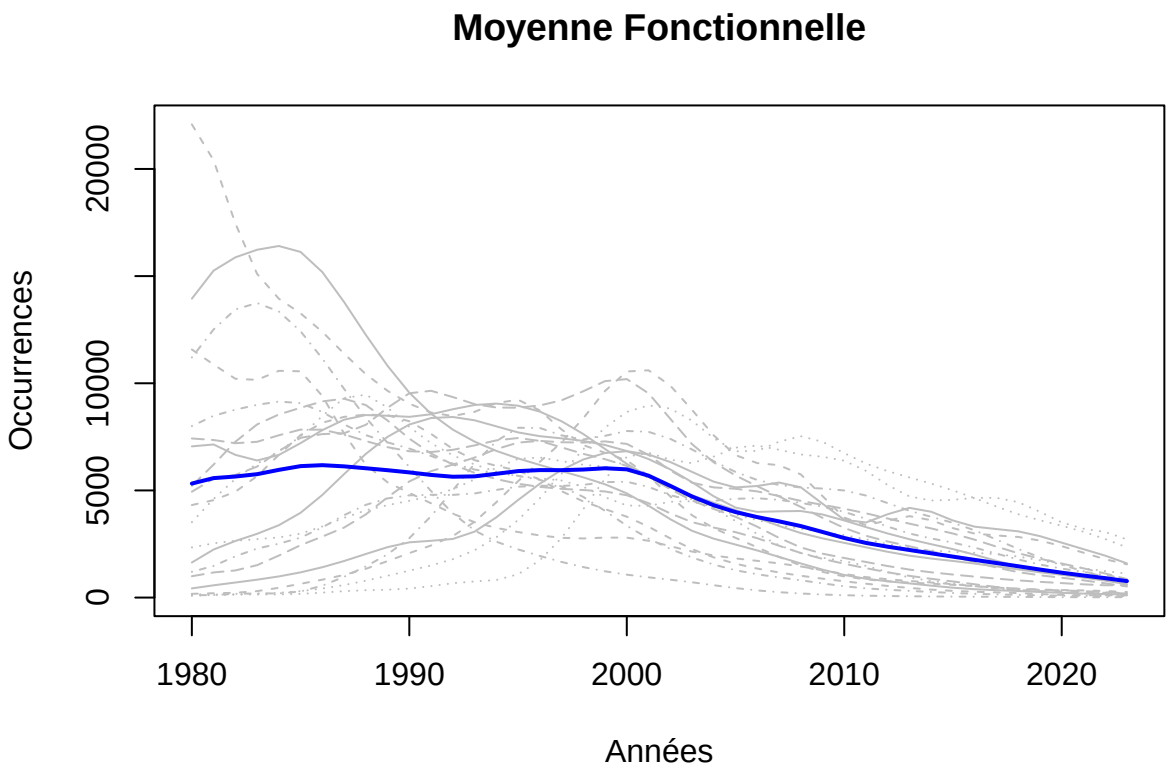
```
fd_deriv2 <- eval.fd(annees,
                     fd_spline$fd,
                     Lfdobj = 2)

matplot(annees, fd_deriv2,
        type = "l", col = 1:ncol(fd_deriv2),
        xlab = "Années",
        ylab = "Dérivée Première",
        main = "Dérivées Fonct. Secondes")
```



Les dérivées secondes montre les variations de la vitesse d'évolution, on peut y voir de nombreux changement de signe, ce qui reflète des alternances fréquentes entre phases de hausse et de baisse de popularité. Les courbes instables suggèrent que la popularité de ces prénoms est sujette à des effets de modes.

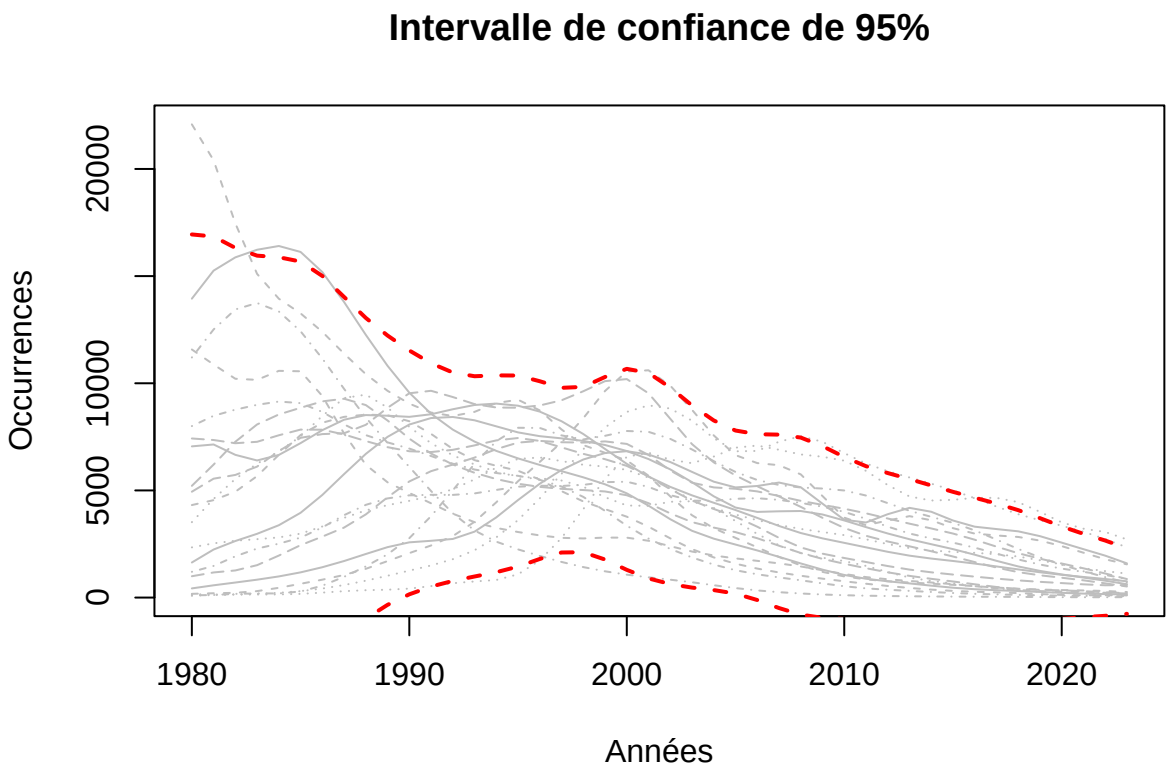
5.2 Moyenne et Variance Fonctionnelles

```
fd_values <- eval.fd(annees, fd_spline$fd)

matplot(annees,
        fd_values,
        type="l",
        col="gray",
        xlab="Années",
        ylab="Occurrences",
        main="Moyenne Fonctionnelle")

mean_fd <- mean.fd(fd_spline$fd)


mean_fd_values <- eval.fd(annees, mean_fd)

lines(annees, mean_fd_values, col="blue", lwd=2)
```



La moyenne fonctionnelle montre bien une tendance générale à la baisse dans la fréquence d'attribution de ces 20 prénoms. Cette diminution progressive traduit une diversification croissante des choix de prénoms dans la population française sur la période étudiée.

```
matplot(annees, fd_values,
        type="l",
        col="gray",
        xlab="Années",
        ylab="Occurrences",
        main="Intervalle de confiance de 95%")

sd_fd <- sd.fd(fd_spline$fd)

lines(annees, eval.fd(annees, mean_fd) + 2 * eval.fd(annees, sd_fd), col = "red", lty = 2, lwd=2)
lines(annees, eval.fd(annees, mean_fd) - 2. * eval.fd(annees, sd_fd), col = "red", lty = 2, lwd=2)
```



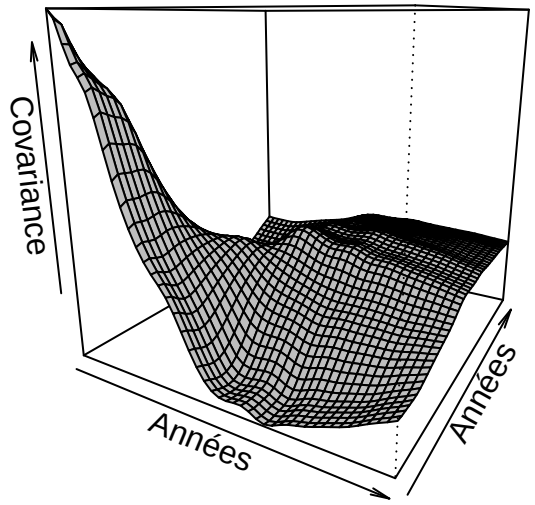
L'écart-type fonctionnel montre une importante disparité de ces 20 prénoms en 1980. Cette disparité se réduit progressivement, suggérant qu'au sein de ces prénoms populaires, il y a moins de prénoms dominant.

5.3 Covariance et Corrélation Fonctionnelles

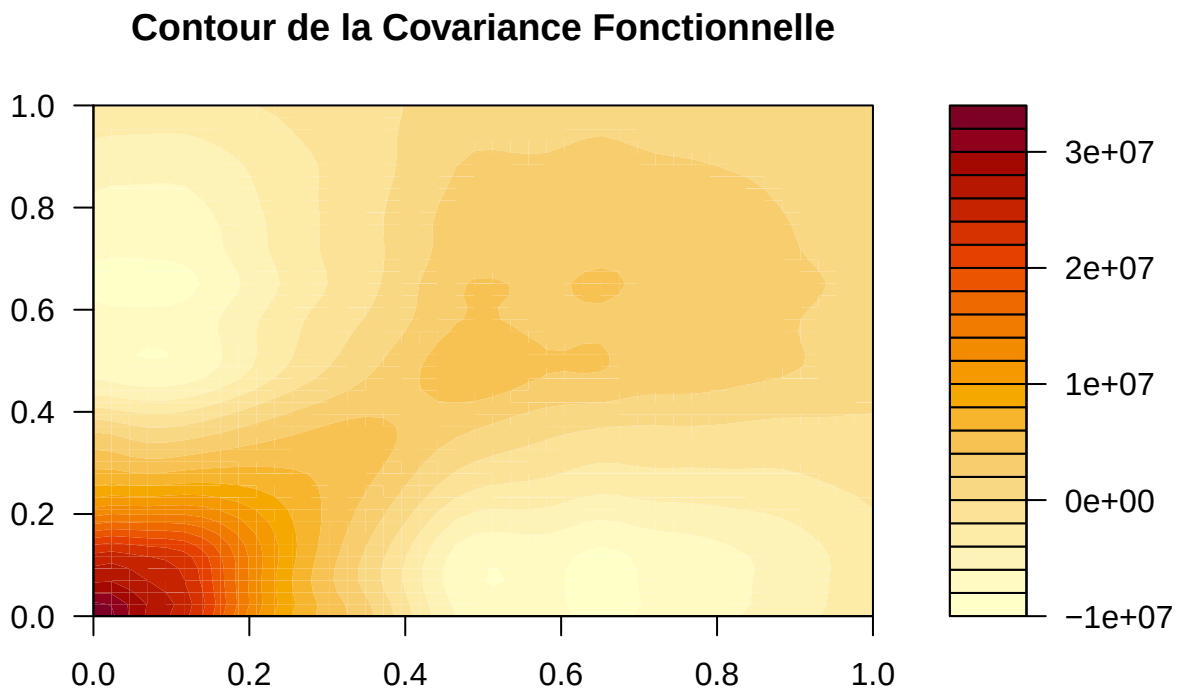
```
cov_fd <- var.fd(fd_spline$fd)
surfcov = eval.bifd(seq(1980, 2023, by=1),
                    seq(1980, 2023, by=1),
                    cov_fd)
```

```
persp(surfcov,
      col = "gray",
      theta = 30,
      xlab = "Années",
      ylab = "Années",
      zlab = "Covariance",
      main = "Surface 3D de Covariance Fonctionnelle")
```

Surface 3D de Covariance Fonctionnelle



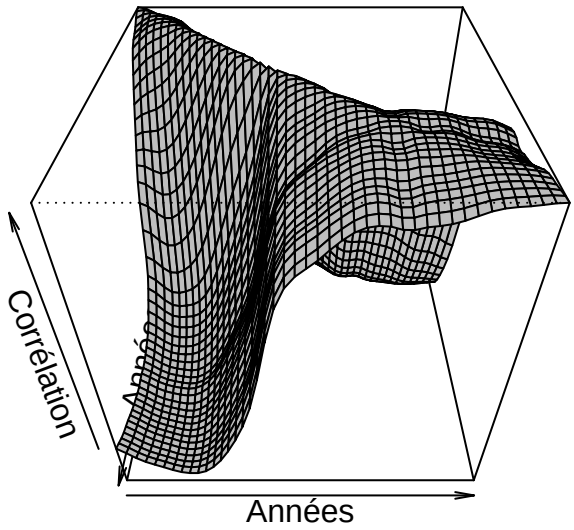

```
filled.contour(surfcov, main = "Contour de la Covariance Fonctionnelle")
```



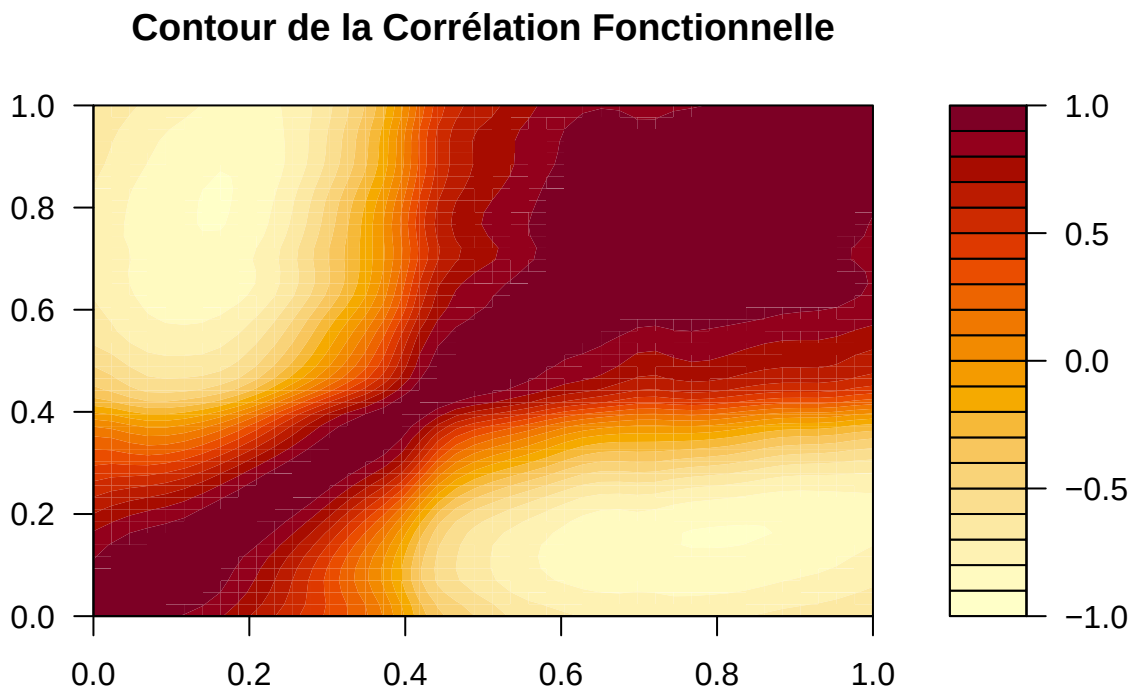
Ensemble, ces deux représentations graphiques mettent en évidence les dynamiques d'évolution de la popularité de ces 20 prénoms, passant d'une forte corrélation initiale, traduit par une variabilité importante, à une diversification progressive, indiquant une évolution plus indépendante de chaque prénom.

```
cor_fd <- cor.fd(seq(1980, 2023, by=1),
                 fd_spline$fd)

persp(cor_fd,
      col = "gray",
      theta = 90,
      phi = 40,
      xlab = "Années",
      ylab = "Années",
      zlab = "Corrélation",
      main = "Surface de Corrélation Fonctionnelle")
```

Surface de Corrélation Fonctionnelle



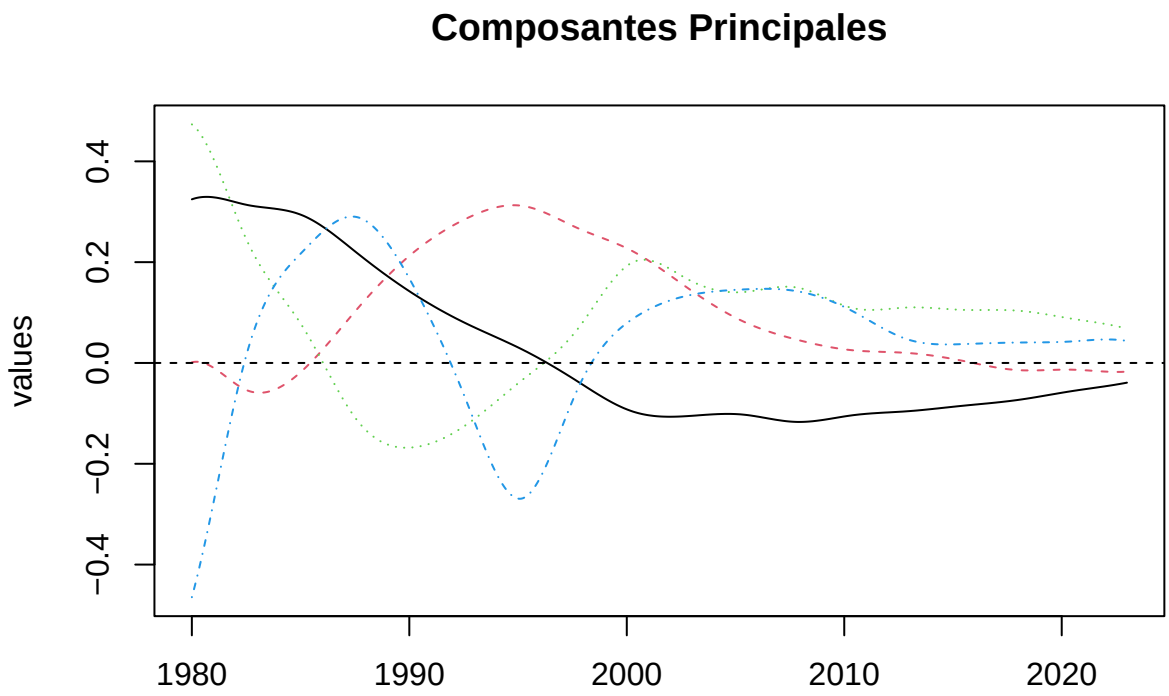

```
filled.contour(cor_fd,
               main = "Contour de la Corrélation Fonctionnelle")
```



Ces deux représentations graphiques révèlent une nouvelle fois une forte interdépendance initiale et une diversification progressive, avant une reconvergence en fin de période. Cela semble indiquer encore une fois une diversification des goûts.

5.4 ACP Fonctionnelle

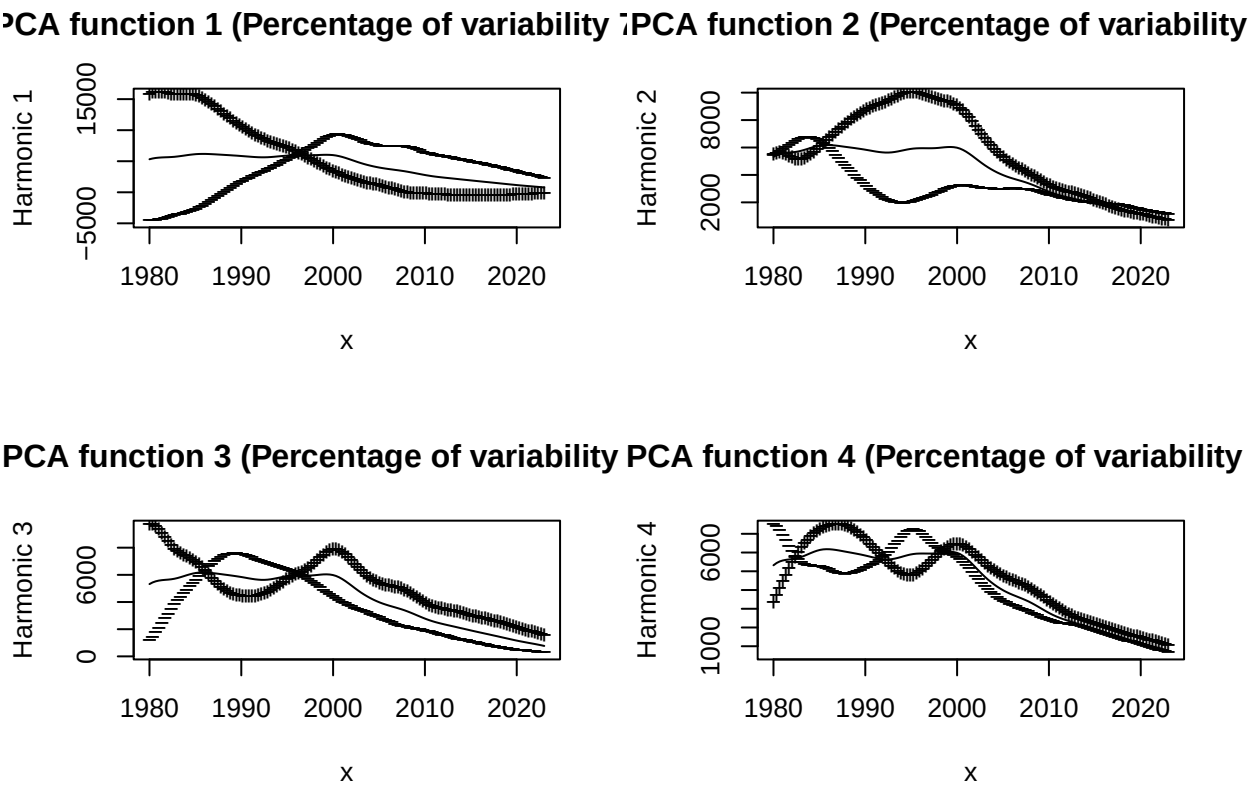
```
fd_acp <- pca.fd(fd_spline$fd,
                 nharm = 4,
                 centerfns = TRUE)

plot(fd_acp$harmonics, main = "Composantes Principales")
```




```
## [1] "done"
```

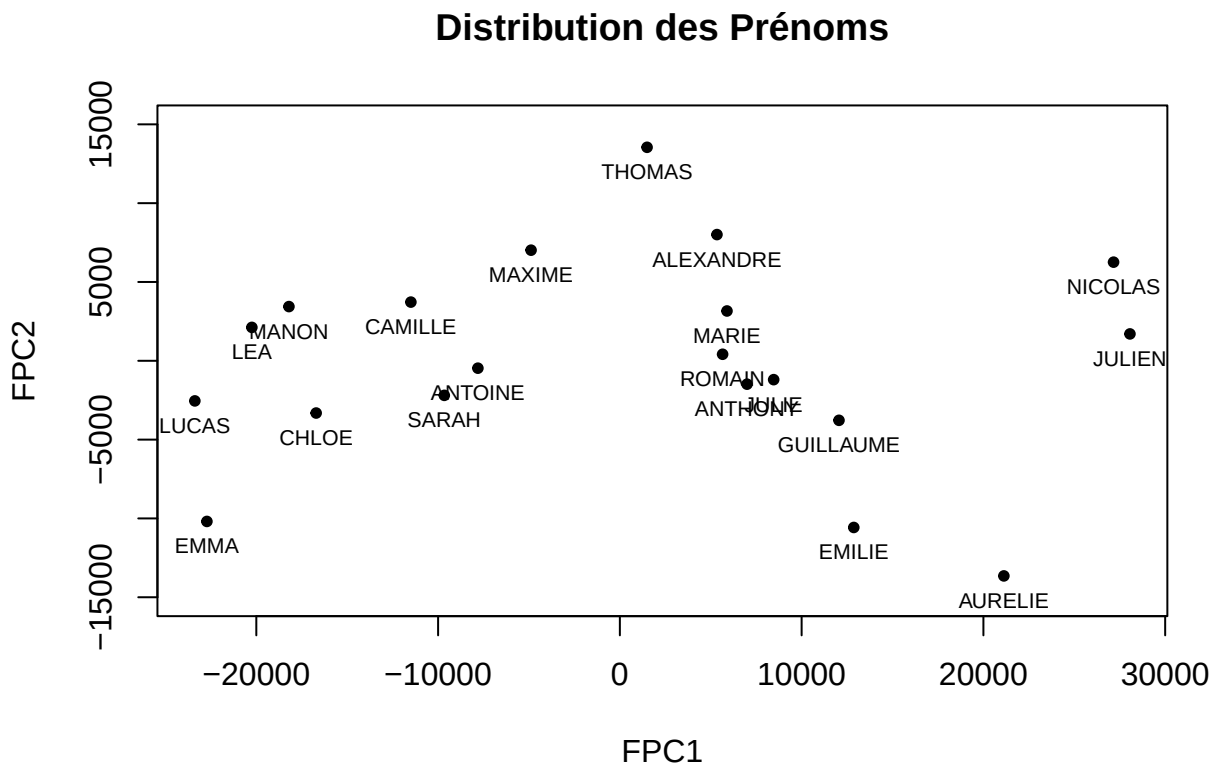
L'analyse des composantes principales révèle une tendance principale de déclin continu (composante 1), accompagnée de tendances cycliques secondaires qui reflètent probablement les effets de mode et les variations temporelles dans le choix des prénoms.

```
par(mfrow=c(2,2))
plot.pca.fd(fd_acp)
```



La première composante principale, qui explique à elle seule ~80% de la variance totale, montre que l'évolution des prénoms donnés à la naissance peut être décrite par une tendance générale de déclin. Les autres composantes, bien que peu informatives, révèlent que l'évolution de ces prénoms est plus complexes, avec des phases de cycles plus subtiles.

```
prenoms <- fd_spline$fd$fdnames$reps

plot(fd_acp$scores[,1],
     fd_acp$scores[,2],
     xlab = "FPC1",
     ylab = "FPC2",
     pch = 20,
     main = "Distribution des Prénoms",
     ylim = c(-15000, 15000))

text(fd_acp$scores[,1],
     fd_acp$scores[,2],
     labels = prenoms,
     cex = 0.7,
     pos = 1)
```



Ce graphique qui représente la distribution de ces prénoms dans le premier plan factoriel permet d'identifier des regroupements :
- A droite, les prénoms Nicolas et Julien ont mieux résisté au déclin global. - En haut, les prénoms Thomas, Maxime et Alexandre ont connu des pics de popularité plus récents. - En bas, Emma, Emilie et Aurélie ont atteint leur pic de popularité plus tôt.

Dans l'ensemble, ce graphique permet de visualiser la diversité des tendances de popularité des prénoms, tout en mettant en évidence les principaux profils d'évolution identifiés par l'analyse en composantes principales.

6. Conclusion

Les résultats obtenus montrent des tendances intéressantes. Le lissage des courbes de fréquence permet de dégager une vision plus claire des évolutions, mettant en évidence des pics de popularité, suivis de périodes de déclin ou de stabilité.
L'analyse a également révélé une certaine convergence vers des prénoms moins concentrés, suggérant une diversification des choix. Les dérivées premières et secondes des courbes ont montré des périodes de changement rapide, illustrant l'influence des effets de mode sur la popularité des prénoms.
Ce travail offre un aperçu sur la manière ont les choix sociaux et culturels évoluent dans le temps.