

Mise à jour du COG - INSEE - Fichier des personnes décédées

Le Fichier des personnes décédées

Suite à l'avis favorable émis par la Commission d'accès aux documents administratifs (CADA) le 17 mai 2019¹, le Fichier des personnes décédées — produit par l'INSEE à partir des enregistrements communaux depuis 1970 — est librement accessible depuis décembre 2019. Il contient les informations suivantes sur chaque personne décédée : le nom et le prénom, le sexe, la date de naissance, le code et le libellé de la commune (pour les personnes nées en France métropolitaine ou dans les DOM/TOM/COM) ou du pays de naissance (pour les personnes nées à l'étranger), la date du décès, le code de la commune de décès, et enfin le numéro d'acte de décès.

Les caractéristiques d'exhaustivité (ensemble des personnes décédées en France) et de désagrégation élevée tant sur le plan spatial (échelon communal), temporel (date au jour) que social (individu) des données ainsi que l'homogénéité et la stabilité de ses variables au cours du temps constituent des opportunités d'analyse certaines.

Sur Datagouv, le fichier des personnes décédées est disponible en tronçons annuels ou trimestriels. Une version concaténée et prête à l'emploi du fichier a été construite par C. Quest, et publiée à cette adresse sur Datagouv: http://data.cquest.org/insee_deces/

```
# Chargement du jeu de données
insee_deces <- readr::read_csv("insee_deces.csv",
                              col_types = cols(code_lieu_deces = col_character(),
                                                code_lieu_naissance = col_character(),
                                                date_deces = col_date(format = "%Y-%m-%d"),
                                                date_naissance = col_character(),
                                                numero_acte_deces = col_character(),
                                                pays_naissance = col_character()))

## Warning: 646 parsing failures.
##   row      col      expected actual      file
## 154839 date_deces date like %Y-%m-%d 1983      'insee_deces.csv'
## 274006 date_deces date like %Y-%m-%d 1983-01 'insee_deces.csv'
## 311595 date_deces date like %Y-%m-%d 1983-01 'insee_deces.csv'
## 388589 date_deces date like %Y-%m-%d 1983-07 'insee_deces.csv'
## 593203 date_deces date like %Y-%m-%d 1984-01 'insee_deces.csv'
## .....
## See problems(...) for more details.

# Retirer les doublons
insee_deces <- insee_deces[!duplicated(insee_deces),]

## [1] "Le fichier comporte 39337 COG uniques de naissance."
## [1] "Le fichier comporte 39424 COG uniques de décès."
```

¹Cf. un extrait de l'avis du CADA du 17 mai 2019 « Le fichier des personnes décédées étant établi par l'INSEE dans le cadre de ses missions de service public, il constitue, à ce titre, un document administratif. Ne contenant pas de mentions relatives à la vie privée, il est communicable à toute personne qui en fait la demande et publiable en ligne en application des dispositions des articles L311-9 et L312-1-1 du code des relations entre le public et l'administration, sans occultation puisqu'il ne comporte pas de données personnelles. »

Cependant, les variables indiquant les codes des communes de naissance et de décès nécessitent une étape préalable d'harmonisation avant d'être utilisées.

1. Le référentiel des codes commune utilisé dans le fichier n'est pas le même selon la date d'enregistrement du décès. La conversion des codes commune utilisés dans le référentiel 2020 permet de mettre à jour 923 915 enregistrements en ce qui concerne les lieux de naissance et 77 273 enregistrements en ce qui concerne les lieux de décès.
2. La prise en compte des collectivités d'outre-mer sous le régime de l'exception législative (absentes du COG 2020) et des modifications ayant affecté les codes de communes érigées en collectivités d'outre-mer dans les années 2000 permet de mettre à jour 26 637 enregistrements en ce qui concerne les lieux de naissance et 12173 enregistrements en ce qui concerne les lieux de décès.
3. L'harmonisation des codes faisant référence à des lieux situés à l'étranger concerne quant à elle 2 141 829 enregistrements pour les naissances et 189 321 enregistrements pour les décès.
4. Des erreurs d'enregistrement des codes communes existent et certaines peuvent être corrigées. Les corrections proposées ici permettent de mettre à jour l'ensemble des codes restés sans correspondance en ce qui concerne les lieux de naissance, soit 4 429 enregistrements. Par ailleurs, seuls 1 648 enregistrements restent sans correspondance en ce qui concerne les lieux de décès.

1. Harmoniser les codes géographiques

Le Code Officiel Géographique (COG) est un document produit chaque année par l'Insee. Ce document rassemble les codes et libellés des communes, des cantons, des arrondissements, des départements, des régions et des pays et territoires étrangers au 1er janvier.

Alors que la plupart des jeux de données disponibles en libre accès sur le site de l'Institut National de la Statistique et des Etudes Economiques sont harmonisés, c'est-à-dire que les codes géographiques renseignés ont été convertis dans le Code Officiel Géographique le plus récent (c'est le cas des données du recensement notamment), ce n'est pas le cas des fichiers des personnes décédées. Au sein de ces derniers, les lieux de naissance et de décès des individus sont enregistrés dans le Code Officiel Géographique en vigueur au moment du décès. Aussi, dans le cadre d'une analyse diachronique tenant compte de la distribution géographique des naissances et des décès, il est nécessaire d'harmoniser ces informations, c'est-à-dire de tenir compte des modifications de la géographie communale survenues depuis le décès de sorte que tous les enregistrements de la base de données soient dans le même COG.

Les modifications de la géographie communale française depuis 1970

Le découpage administratif des territoires français n'est pas stable dans le temps.

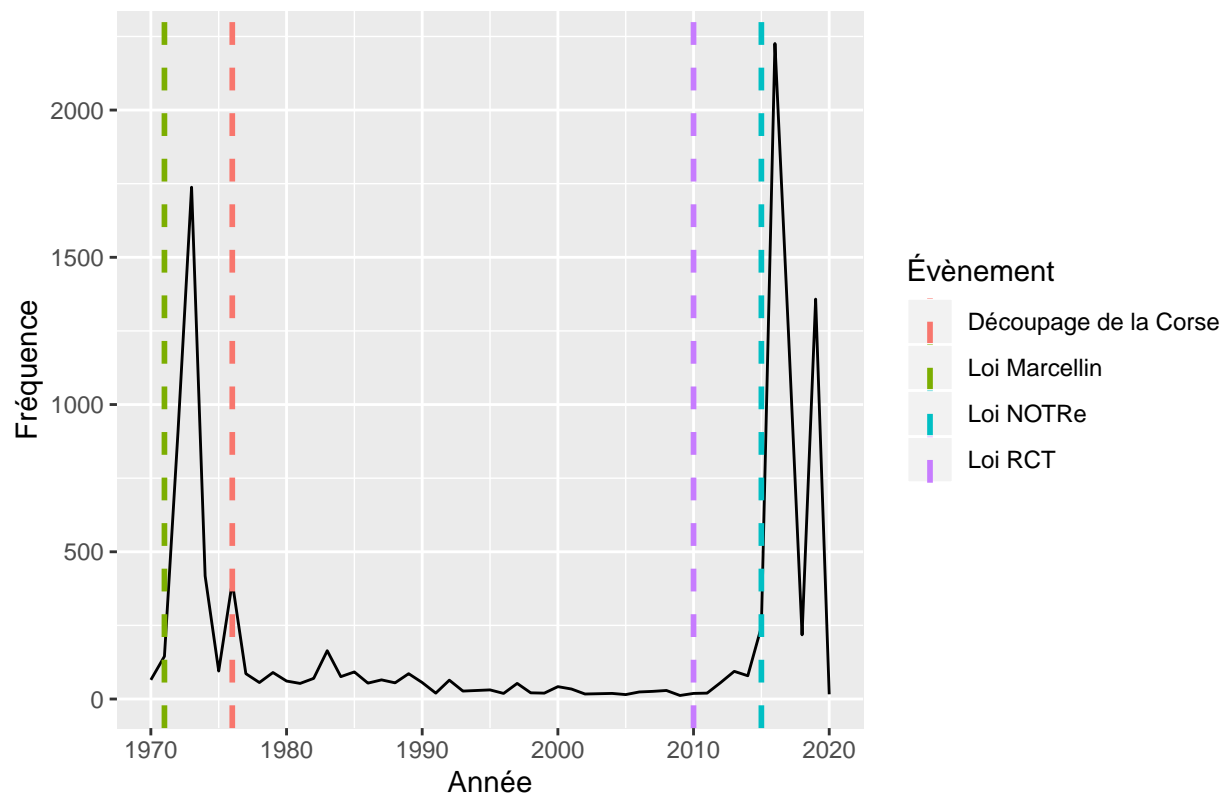
La table des mouvements communaux permet de constater qu'entre le 1er janvier 1970 et le 1er janvier 2020, 10 921 évènements ont affecté la géographie communale française. Certaines communes ont changé de département (la Corse a été scindée en deux départements en 1976 par exemple) ce qui a entraîné la modification de leur code communal, d'autres ont fusionné ou défusionné.

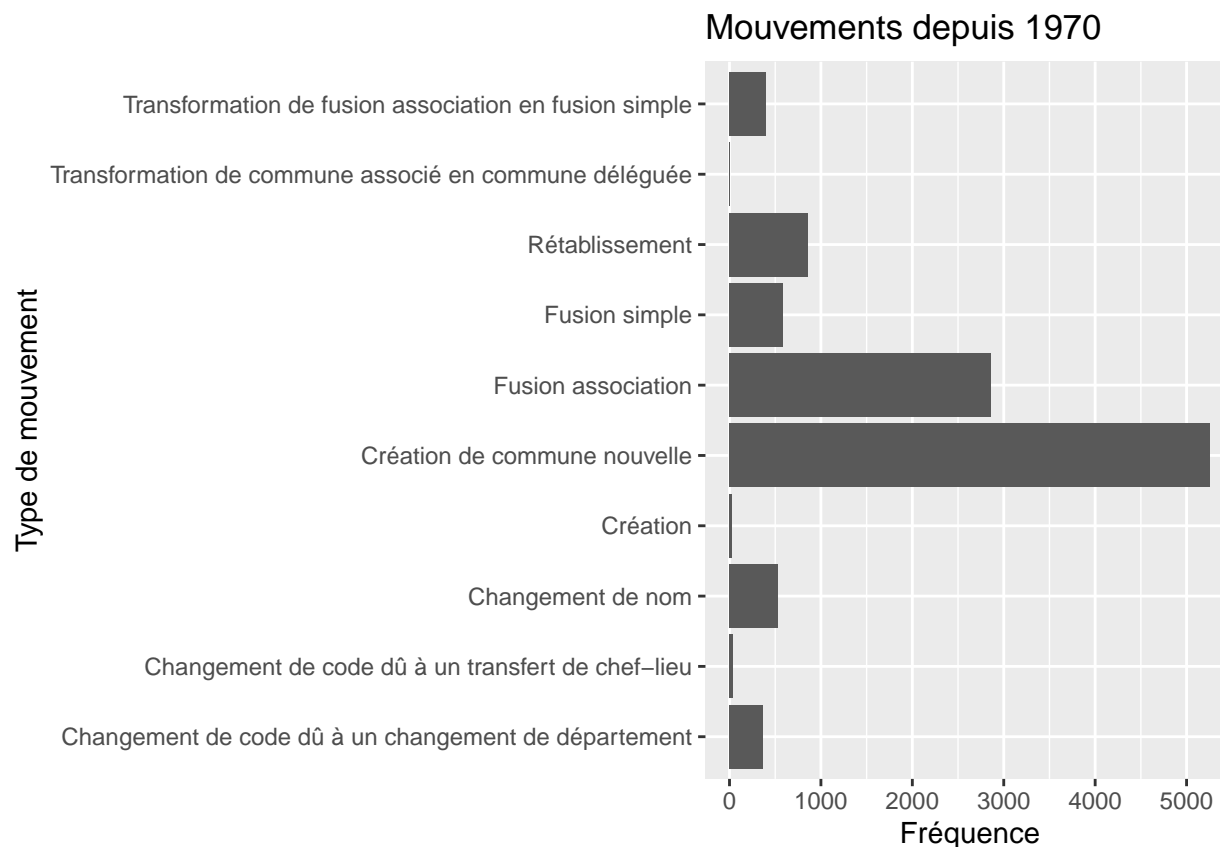
```
# Chargement de la table des mouvements communaux
mvtcommune <- readr::read_csv("mvtcommune2020.csv")
```

```
## Parsed with column specification:
## cols(
##   mod = col_double(),
##   date_eff = col_date(format = ""),
##   typecom_av = col_character(),
##   com_av = col_character(),
```

```
## tncc_av = col_double(),
## ncc_av = col_character(),
## nccenr_av = col_character(),
## libelle_av = col_character(),
## typecom_ap = col_character(),
## com_ap = col_character(),
## tncc_ap = col_double(),
## ncc_ap = col_character(),
## nccenr_ap = col_character(),
## libelle_ap = col_character()
## )
```

Mouvements communaux par année depuis 1970





N.B.: Le package COGugaison spécialement développé pour harmoniser ces variables n'est pas utilisé car: - le référentiel le plus ancien utilisé par COGugaison date de 1968 et des codes commune en vigueur avant 1968 sont présents dans la base. - dans les cas où des communes auraient connu deux changements successifs dans une période inter-censitaire, l'harmonisation ne peut se faire automatiquement. - il est nécessaire de connaître le millésime de référence du code commune pour opérer sa transposition, or une partie des codes utilisés en date du décès correspondent à des référentiels périmés.

Procédure d'harmonisation appliquée au lieu de naissance

Le COG 2020 est requis.

```
## Parsed with column specification:
## cols(
##   typecom = col_character(),
##   com = col_character(),
##   reg = col_double(),
##   dep = col_character(),
##   arr = col_character(),
##   tncc = col_double(),
##   ncc = col_character(),
##   nccenr = col_character(),
##   libelle = col_character(),
##   can = col_character(),
##   comparent = col_character()
## )
```

1. Comparaison des codes commune de naissance avec le COG 2020

```
# Table de travail
sel <- insee_deces

# Comparaison avec le COG 2020
## sélection des naissances dont le code existe en 2020
sel0 <- sel[sel$code_lieu_naissance %in% COG$com, ]
## création d'une nouvelle variable code_nai20 pour stocker les codes harmonisés
sel0$code_nai20 <- sel0$code_lieu_naissance

## sélection des naissances dont le code n'existe pas en 2020
sel <- sel[!(sel$code_lieu_naissance %in% COG$com), ]

## [1] "Tous les enregistrements ont-ils été conservés ? TRUE"
## [1] "Il reste 2411 / 39337 codes uniques à traiter"
```

2. Recoder les communes non renseignées

```
# Coder les codes non renseignés comme code inconnu
sel1 <- sel[is.na(sel$code_lieu_naissance),]
sel1$code_nai20 <- paste("INCONNU")

# Sélection des naissances dont le code est renseigné
sel <- sel[!(is.na(sel$code_lieu_naissance)),]

# Recoder les codes naissance des communes fictives en 00990)
sel2 <- sel[substr(sel$code_lieu_naissance,3,5)=="990",]
sel2$code_nai20 <- paste("00990")

# Sélection des naissances dont le code n'est pas une commune fictive
sel <- sel[!(substr(sel$code_lieu_naissance,3,5)=="990"),]

## [1] "Tous les enregistrements ont-ils été conservés ? TRUE"
## [1] "Il reste 2317 / 39337 codes uniques à traiter"
```

3. Recoder en suivant les mouvements communaux

3.1 Préparer la table de passage à partir du fichier des mouvements communaux

- Les mouvements qui n'ont pas entraîné la disparition du code communal
- Les défusions: Quand une commune a défusionné, il n'est pas possible de savoir dans laquelle des communes résultantes la personne est née ou décédée. Nous faisons le choix de conserver la localisation de la naissance ou du décès dans la commune ayant conservé son code.

```
# On vérifie que le code ne disparaît jamais lors des défusions
mvtcommune$mod <- as.character(mvtcommune$mod)
defusions <- mvtcommune[mvtcommune$mod %in% c("20","21"),]

## [1] "Combien de codes ont disparu dans les défusions ? 0"
```

- Pour créer la table de passage entre codes anciens et COG 2020, on ignore certains mouvements qui n'impliquent pas de changement de code (par exemple, changement de nom de la commune, changement de statut...)

```
# Retirer certains types de mouvements
mvtcommune <- mvtcommune[!(mvtcommune$mod %in% c("10", "20", "21")),]
mvtcommune <- mvtcommune[!(mvtcommune$typecom_av %in% c("COMA", "COMD"))
                           &!(mvtcommune$typecom_ap %in% c("COMA", "COMD")),]
```

- Construction de la table de passage

```
# Retirer les changements de code en double
mvtcommune <- mvtcommune[!(duplicated(mvtcommune[,c(4,10)]))],]
# Conserver les changements qui modifient le code
mvtcommune <- mvtcommune[mvtcommune$com_av!=mvtcommune$com_ap,]

## Gérer les 3 cas "d'aller-retour" de code
# Changements de département successifs conduisant à la restauration du code initial
mvtcommune <- mvtcommune[!(mvtcommune$com_av %in% c("78143", "78620")),]
# L'Oudon:
# Le code 14624, supprimé en 1973 suite à une fusion, a été réaffecté à un ensemble
# de communes ayant refusionné après une défusion en 2014

mvtcommune$com_ap[mvtcommune$com_av=="14697"] <- "14654"
mvtcommune$com_ap[mvtcommune$com_av=="14697"] <- "14654"

# sélection des mouvements où com_av apparaît plusieurs fois
tab <- mvtcommune[duplicated(mvtcommune$com_av), 4]
#extraction de ces cas
tab1 <- mvtcommune[mvtcommune$com_av %in% tab$com_av, ]

# 2 cas de figure:

# Les communes supprimées:
# 6 cas de partition du territoire entre plusieurs autres communes.
# Il n'est pas possible d'attribuer un code commune unique en vigueur à ces communes.
# Ces enregistrements sont considérés comme non localisables.
# Le code commune "inconnu" est attribué.

tab2 <- tab1[tab1$mod == "30",]
# suppression de mvt_commune
mvtcommune <- mvtcommune[!((mvtcommune$com_av %in% tab2$com_av) &
                           (mvtcommune$com_ap %in% tab2$com_ap)) ,]
# remplacement par ligne de passage vers commune inconnue
tab2 <- tab2[!(duplicated(tab2$com_av)),]
tab2$com_ap <- "INCONNU"
tab2[,11:14] <- NA
mvtcommune <- rbind(mvtcommune, tab2)

# Gérer les cas où un code a été modifié puis restauré puis de nouveau modifié
tab3 <- tab1[tab1$mod != "30",]
# sélection des mouvements doublon les plus anciens et retrait de mvt_unique
tab3 <- tab3[duplicated(tab3$com_av),]
mvtcommune <- mvtcommune[!((mvtcommune$com_av %in% tab3$com_av) &
```

```

(mvtcommune$com_ap %in% tab3$com_ap) &
(mvtcommune$date_eff %in% tab3$date_eff)) ,]

# mouvements uniques: sélection des codes "après" qui ne réapparaissent pas
# dans les codes "avant"
mvt_unique <- mvtcommune[!(mvtcommune$com_ap %in% mvtcommune$com_av),]

# mouvements 1: sélection des codes "après" qui réapparaissent
# dans les codes "avant" (= autre changement)

mvt_1 <- mvtcommune[mvtcommune$com_ap %in% mvtcommune$com_av,]

# mouvements 2: sélection des codes "après" du mouvement 1 qui réapparaissent encore
# dans les codes "avant" (= autre changement)
mvt_2 <- mvt_1[mvt_1$com_ap %in% mvt_1$com_av,]

# mouvements 3: sélection des codes "après" du mouvement 2 qui réapparaissent encore
# dans les codes "avant" (= autre changement)
# 0 observation : aucun cas
# mvt_3 <- mvt_2[mvt_2$com_ap %in% mvt_2$com_av,]

# Supprimer les mvt_2 qui sont dans le mvt_1
mvt_1 <- mvt_1[!(mvt_1$com_ap %in% mvt_1$com_av),]

# Sélection des colonnes com_av et com_ap
mvt_unique <- mvt_unique[,c(4,10)]
mvt_1 <- mvt_1[,c(4,10)]
mvt_2 <- mvt_2[,c(4,10)]

colnames(mvt_1) <- c("com_av", "com_int")
colnames(mvt_2) <- c("com_av", "com_int")

# Jointure des tables de mouvements successifs
mvt_1 <- left_join(mvt_1, mvt_unique, by=c("com_int"="com_av"))
mvt_1 <- mvt_1[, -2]
mvt_2 <- left_join(mvt_2, mvt_1, by=c("com_int"="com_av"))
mvt_2 <- mvt_2[, -2]

# Création de la table de passage
passage <- rbind(mvt_unique, mvt_1, mvt_2)

# vérification que tous les com_ap sont bien dans le COG2020
unique(passage[!(passage$com_ap %in% COG$com),])

```

```

## # A tibble: 6 x 2
##   com_av com_ap
##   <chr>  <chr>
## 1 08227 INCONNU
## 2 51606 INCONNU
## 3 45117 INCONNU
## 4 60606 INCONNU
## 5 51385 INCONNU
## 6 77362 INCONNU

```

```
summary(duplicated(passage))
```

```
##      Mode   FALSE  
## logical    4388
```

```
rm(tab, tab1, tab2, tab3, mvt_unique, mvt_1, mvt_2, mvt_3)
```

```
## Warning in rm(tab, tab1, tab2, tab3, mvt_unique, mvt_1, mvt_2, mvt_3):  
## objet 'mvt_3' introuvable
```

3.2. Jointure avec la table de passage

```
sel <- left_join(sel, passage, by=c("code_lieu_naissance"="com_av"))
```

```
sel3 <- sel[!(is.na(sel$com_ap)),]  
sel3$code_nai20 <- sel3$com_ap
```

```
# on retire la colonne jointe  
sel3 <- sel3 %>% select(-com_ap)
```

```
# sélection des naissances dont le code n'est pas encore dans le bon référentiel  
sel <- sel[is.na(sel$com_ap),]  
# on retire la colonne jointe  
sel <- sel%>% select(-com_ap)
```

```
## [1] "Tous les enregistrements ont-ils été conservés ? TRUE"  
## [1] "Combien d'enregistrements ont été récupérés ? 923915"  
## [1] "Soit 3.71 \n          % de la base."  
## [1] "Il reste 395 / 39337 codes uniques à traiter"
```

4. Correction des codes absents du COG 2020 et de l'historique des mouvements communaux

4.1 Naissances à l'étranger

```
# codes commune à l'étranger  
sel4 <- sel[substr(sel$code_lieu_naissance,1,2) == "99",]  
sel4$code_nai20 <- paste("99999")  
sel <- sel[substr(sel$code_lieu_naissance,1,2) != "99",]  
  
# Non prise en compte de l'indépendance de l'Algérie dans l'enregistrement du code  
sel5 <- sel[sel$code_lieu_naissance %in% c("91352", "92352", "93352", "94352",  
                                          "91", "92", "20000", "123"),]  
sel5$code_nai20 <- paste("99999")  
sel <- sel[!(sel$code_lieu_naissance %in% c("91352", "92352", "93352", "94352",  
                                          "91", "92", "20000", "123")),]  
  
### vérification qu'il n'y a pas eu de perte d'enregistrement  
print(paste("Tous les enregistrements ont-ils été conservés ?",  
            nrow(sel0)+nrow(sel1)+ nrow(sel2) + nrow(sel3)+ nrow(sel4) +  
            nrow(sel5)+ nrow(sel)==nrow(insee_decés),  
            sep = " "))
```

```
## [1] "Tous les enregistrements ont-ils été conservés ? TRUE"
```



```
## [1] "Combien d'enregistrements ont été récupérés ? 2141829"
## [1] "Soit 8.59 % de la base."
## [1] "Il reste 155 / 39337 codes uniques à traiter"
```

4.2 Collectivités d'outre-mer placées sous le régime de l'exception législative

```
sel6 <- sel[substr(sel$code_lieu_naissance,1,2) == "98",]
sel6$code_nai20 <- sel6$code_lieu_naissance
sel <- sel[!(substr(sel$code_lieu_naissance,1,2) == "98"),]
```

4.3 Communes érigées en collectivités d'outre-mer, habituellement codées en 3 chiffres et dont le code en 5 positions a été modifié suite à leur changement de statut: St Barthelemy, St Martin, St Pierre, Miquelon

```
# Codes absent du COG: St Barthelemy, St Martin, St Pierre, Miquelon
sel7 <- sel[sel$code_lieu_naissance %in% c("97801","97701","97502","97501"),]
sel7$code_nai20 <- sel7$code_lieu_naissance

sel8 <- sel[sel$code_lieu_naissance %in% c("97127"),] # St Martin
sel8$code_nai20 <- paste("97801") # Nouvelle codification en 5 positions
sel9 <- sel[sel$code_lieu_naissance %in% c("97123"),] # St Barthelemy
sel9$code_nai20 <- paste("97701") # Nouvelle codification en 5 positions

sel <- sel[!(sel$code_lieu_naissance %in% c("97127","97123","97801",
                                           "97701","97501","97502")),]
```

```
## [1] "Combien d'enregistrements ont été récupérés ? 26637"
## [1] "Soit 0.11 % de la base."
## [1] "Il reste 15 / 39337 codes uniques à traiter"
```

5. Codification des codes restants comme inconnus

```
sel10 <- sel
sel10$code_nai20 <- paste("INCONNU")
print(paste("Les codes restés sans correspondance concernent",
            nrow(sel10),"enregistrements."))
```

```
## [1] "Les codes restés sans correspondance concernent 4429 enregistrements."
```

6. Reconstituer le fichier des personnes décédées avec le COG à jour

```
# rassembler les tables
sel <- rbind(sel0,sel1,sel2,sel3,sel4,sel5,sel6,sel7,sel8,sel9,sel10)

## [1] "Tous les enregistrements ont ils été conservés? TRUE"
insee_deces <- sel
rm(sel,sel0,sel1,sel2,sel3,sel4,sel5,sel6,sel7,sel8,sel9,sel10)
```

7. Propositions de corrections au cas par cas

À appliquer à la place des parties 5 et 6 pour les lieux de naissance et à adapter pour les lieux de décès.

Les corrections proposées ci-dessous portent sur les codes géographiques restés sans correspondances après les opérations précédentes (jointure entre le fichier des personnes décédées et le COG 2020, prise en compte des mouvements communaux recensés par l'INSEE, correction des codes faisant référence à des territoires étrangers et prise en compte de la codification spécifique aux collectivités d'outre-mer).

Ces corrections sont deux types: 1. Prise en compte de modifications de la géographie communale absentes de la table des mouvements communaux. 2. Correction d'erreurs de saisies supposées.

Il s'agit de propositions sans garantie d'exactitude (en particulier pour les corrections d'erreur de saisie) en l'absence d'autres sources d'information.

Ces corrections permettent de recoder l'ensemble des codes géographiques des lieux de naissance présents dans le fichier des personnes décédées entre 1970 et 2019. Par contre, seule une partie des codes géographiques des lieux de décès est recodée. Les codes géographiques des lieux de décès restés sans correspondance après l'harmonisation du COG sont beaucoup plus nombreux et le travail de correction au cas par cas est plus difficile, car chaque code présente peu d'occurrences et, car le fichier des personnes décédées ne contient pas de variable renseignant le libellé du lieu de décès comme c'est le cas pour le lieu de naissance.

7.1 Modifications de la géographie communale absentes de la table des mouvements communaux

```
# # Magny en Vexin
# sel10 <- sel[sel$code_lieu_naissance %in% c("95025", "95065"),]
# sel10$code_nai20 <- paste("95355")
# # Amélie les bains Palalda
# sel11 <- sel[sel$code_lieu_naissance %in% c("66131", "66235"),]
# sel11$code_nai20 <- paste("66003")
# # Hauteville Lompnes
# sel12 <- sel[sel$code_lieu_naissance %in% c("01220", "01459"),]
# sel12$code_nai20 <- paste("01185")
# # Thiaucourt-Regniéville
# sel13 <- sel[sel$code_lieu_naissance %in% c("54448"),]
# sel13$code_nai20 <- paste("54518")
# # Limey-Remenauville
# sel14 <- sel[sel$code_lieu_naissance %in% c("54454"),]
# sel14$code_nai20 <- paste("54316")
#
# # retirer les corrections aux cas par cas et vérifier que sel est vide
# sel <- sel[!(sel$code_lieu_naissance %in%
#             #c("95025", "95065", "66131", "66235", "01220", "01459", "54448", "54454")),]
```

7.2 Correction d'erreurs de saisie

```
# # Marseille
# sel15 <- sel[sel$code_lieu_naissance %in% c("13155"),]
# sel15$code_nai20 <- paste("13055")
#
# # Paris
# sel16 <- sel[sel$code_lieu_naissance %in% c("75212", "75213", "75214", "75215"),]
# sel16$code_nai20 <- paste("75056")
#
```

```

# # Corbeil-Essonnes
# sel17 <- sel[sel$code_lieu_naissance %in% c("91221"),]
# sel17$code_nai20 <- paste("91174")
#
# # Nantes
# sel18 <- sel[sel$code_lieu_naissance %in% c("44309"),]
# sel18$code_nai20 <- paste("44109")
#
# # Retirer les corrections aux cas par cas
# sel <- sel[!(sel$code_lieu_naissance %in%
#             # c("75212", "75213", "75214", "75215", "13155", "91221", "44309", "97501")),]

# print(paste("Il reste", length(unique(sel$code_lieu_naissance)), "/",
#             # length(unique(insee_deces$code_lieu_naissance)),
#             # "codes uniques à traiter", sep = " " ))

```

7.3. Reconstituer le fichier des personnes décédées avec le COG à jour

Une fois les corrections manuelles effectuées, reprendre le chunk {r agregation} au point 6 en ajoutant les sel10 à 18 au rbind.

Application au lieu de décès

La même procédure est appliquée aux lieux de décès

```

# Table de travail
sel <- insee_deces

# Sélection des deces dont le code existe en 2020
sel0 <- sel[sel$code_lieu_deces %in% COG$com, ]
## Création d'une nouvelle variable code_dec20 (codes harmonisés)
sel0$code_dec20 <- sel0$code_lieu_deces

## Sélection des deces dont le code n'existe pas en 2020
sel <- sel[!(sel$code_lieu_deces %in% COG$com), ]

```

```
## [1] "Tous les enregistrements ont-ils été conservés ? TRUE"
```

```
## [1] "Il reste 2320 / 39424 codes uniques à traiter"
```

```

# Coder les codes non renseignés ou nuls comme commune fictive
sel1 <- sel[is.na(sel$code_lieu_deces),]
sel1$code_dec20 <- paste("INCONN")
# sélection des deces dont le code est renseigné
sel <- sel[!(is.na(sel$code_lieu_deces)),]

```

```

sel2 <- sel[sel$code_lieu_deces=="00000",]
sel2$code_dec20 <- paste("INCONN")
# sélection des deces dont le code n'est pas nul
sel <- sel[sel$code_lieu_deces!="00000",]

```

```
## [1] "Tous les enregistrements ont-ils été conservés ? TRUE"
```

```

# recoder les codes décès des communes fictives en 00990
sel3 <- sel[substr(sel$code_lieu_deces,3,5)=="990",]

```

```

sel3$code_dec20 <- paste("00990")

# sélection des deces dont le code n'est pas une commune fictive
sel <- sel[!(substr(sel$code_lieu_deces,3,5)=="990"),]

## [1] "Tous les enregistrements ont-ils été conservés ? TRUE"
## [1] "Il reste 2316 / 39424 codes uniques à traiter"

### Prise en compte des mouvements communaux
sel <- left_join(sel, passage, by=c("code_lieu_deces"="com_av"))

sel4 <- sel[!(is.na(sel$com_ap)),]
sel4$code_dec20 <- sel4$com_ap
# on retire la colonne jointe
sel4 <- sel4 %>% select(-com_ap)

# sélection des deces dont le code n'est pas encore dans le bon référentiel
sel <- sel[is.na(sel$com_ap),]
# on retire la colonne jointe
sel <- sel%>% select(-com_ap)

## [1] "Tous les enregistrements ont-ils été conservés ? TRUE"
## [1] "Combien d'enregistrements ont été récupérés ? 77273"
## [1] "Soit 0.31 % de la base."
## [1] "Il reste 1283 / 39424 codes uniques à traiter"

# Codes commune à l'étranger
sel5 <- sel[substr(sel$code_lieu_deces,1,2) == "99",]
sel5$code_dec20 <- paste("99999")
sel <- sel[substr(sel$code_lieu_deces,1,2) != "99",]

# Non prise en compte de l'indépendance de l'Algérie dans l'enregistrement du code
sel6 <- sel[sel$code_lieu_deces %in% c("91352", "92352", "93352", "94352",
                                     "91", "92", "20000", "123"),]

# Un seul cas pour les lieux de décès
sel6$code_dec20 <- paste("99999")
sel <- sel[!(sel$code_lieu_deces %in% c("91352", "92352", "93352", "94352",
                                     "91", "92", "20000", "123")),]

## [1] "Tous les enregistrements ont-ils été conservés ? TRUE"
## [1] "Combien d'enregistrements ont été récupérés ? 189321"
## [1] "Soit 0.76 % de la base."
## [1] "Il reste 1046 / 39424 codes uniques à traiter"

## Collectivités d'outre-mer:
sel7 <- sel[substr(sel$code_lieu_deces,1,2) == "98",]
sel7$code_dec20 <- sel7$code_lieu_deces
sel <- sel[!(substr(sel$code_lieu_deces,1,2) == "98"),]
sel8 <- sel[sel$code_lieu_deces %in% c("97801", "97701", "97502", "97501"),]
sel8$code_dec20 <- sel8$code_lieu_deces
sel9 <- sel[sel$code_lieu_deces %in% c("97127"),]
sel9$code_dec20 <- paste("97801")

```

```

sel10 <- sel[sel$code_lieu_deces %in% c("97123"),]
sel10$code_dec20 <- paste("97701")
sel <- sel[!(sel$code_lieu_deces %in%
             c("97127", "97123", "97801", "97701", "97501", "97502")),]

#### progression ####
print(paste("Combien d'enregistrements ont été récupérés ?",
            nrow(sel7)+nrow(sel8)+nrow(sel9)+nrow(sel10)))

## [1] "Combien d'enregistrements ont été récupérés ? 12173"

print(paste("Soit",
            round((nrow(sel7)+nrow(sel8)+nrow(sel9)+nrow(sel10))/nrow(insee_deces)*100,digits=2),
            "% de la base.", sep=" "))

## [1] "Soit 0.05 % de la base."

print(paste("Il reste",length(unique(sel$code_lieu_deces)), "/",
            length(unique(insee_deces$code_lieu_deces)),
            "codes uniques à traiter", sep=" " ))

## [1] "Il reste 931 / 39424 codes uniques à traiter"

# Identifier les corrections restantes
a_recoder <- as.data.frame(table(as.factor(sel$code_lieu_deces)))
print(paste("Le nombre moyen d'occurences pour chaque code resté sans correspondance est",
            round(mean(a_recoder$Freq)), digits=2))

## [1] "Le nombre moyen d'occurences pour chaque code resté sans correspondance est 2 2"

print(paste("Le nombre maximal d'occurences pour un code resté sans correspondance est",
            max(a_recoder$Freq)))

## [1] "Le nombre maximal d'occurences pour un code resté sans correspondance est 29"

# Coder les codes restants comme inconnus
sel11 <- sel
sel11$code_dec20 <- paste("INCONNU")

## [1] "Les codes restés sans correspondances concernent 1648 enregistrements."

# Reconstituer le fichier des personnes décédées
sel <- rbind(sel0,sel1,sel2,sel3,sel4,sel5, sel6,sel7,sel8,sel9,sel10,sel11)

## [1] "Tous les enregistrements ont ils été conservés? TRUE"

insee_deces <- sel
rm(sel,sel0,sel1,sel2,sel3,sel4,sel5, sel6,sel7,sel8, sel9,sel10,sel11)

```

Export du résultat

```

#write.csv(insee_deces,"insee_deces_harmonise_COG2020.csv")

```