Read code

Groupe_1

2025-02-18

Ce script propose des codes pour merger les bases ehcvm 2021 et hdx de la Côte d'ivoire. Pour ce faire, les variables du niveau le plus fin(niveau 3) de chaque base ont été nettoyées pour faciliter la fusion des bases. Dans un premier temps, nous avons nettoyé les bases, mergé et modifié manuelllement les communes qui n'avaient pas de correspondant(24). Dans un second temps, pour les communes qui n'avaient pas de correspondant, nous avons utilisé la distance de Levenshtein. Les communes restantes(3) après cette étape ont été modifées manuellement. Ce travail est le fruit du travail de :

————————-Khadidiatou Coulibaly————————-

————————-Jeanne De La Flèche ONANENA AMANA————-

————————-Samba DIENG————————————

————————-Tamsir NDONG————————————-

————————Elèves en ISE1-Cycle long————————


```
library(haven)
library(readxl)
library(labelled)
library(stringi)
library(dplyr)
```

```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Chargeons les données EHCVM
ehcvm_civ <- read_dta("ehcvm_individu_civ2021.dta")

# Copions les données pour les tests
ehcvm_civ_test <- ehcvm_civ
```

```
# On convertit les variables qualitatives en facteur
qualitative_vars <- names(ehcvm_civ_test)[sapply(ehcvm_civ_test, is.labelled)]
ehcvm_civ_test[qualitative_vars] <- lapply(ehcvm_civ_test[qualitative_vars], to_factor)

# On convertit sp_commune en facteur pour des vérifications
ehcvm_civ_test$sp_commune <- to_factor(ehcvm_civ_test$sp_commune)
```

```r
# Cette fonction nettoie les noms de communes
clean_text <- function(text) {
  text <- stri_trans_general(text, "Latin-ASCII")  # On supprime les accents
  text <- toupper(text)  # On convertir toutes les communes en majuscules
  text <- gsub("'", "", text)  # On supprimer les apostrophes
  text <- gsub("[^A-Z0-9 ]", "", text)  # On supprimer les autres caractères spéciaux sauf espaces et c
  return(text)
}
```

```r
# Appliquons la transformation et ajoutons une colonne avec les noms de communes nettoyés
ehcvm_civ_test$commune_clean <- sapply(ehcvm_civ_test$sp_commune, clean_text)
```

Le dictionnaire qui va suivre donne les noms des communes qui n'ont pas été fusionnées. On fait donc des identifications pour contourner ce problème.

```r
# Dictionnaire des corrections
corrections <- list(
  "ZOUANHOUNIEN" = "ZOUAN HOUNIEN",
  "TIENDIEKRO" = "TIE NDIEKRO",
  "SEDIEGO" = "SEDIOGO",
  "NZEKRESSESSOU" = "NZECREZESSOU",
  "NIAKARAMADOUGOU" = "NIAKARAMANDOUGOU",
  "NGUESSANKRO" =  "NGUESSANKRO DE BONGOUANOU",
  "MARHANDALLAH" = "MARANDALLAH",
  "KETRO BASSAM" = "KETROBASSAM",
  "KOKUMBO" = "KOKOUMBO",
  "KOUASSIDATTEKRO" = "KOUASSIDATEKRO",
  "KOUASSINIAGUNI" = "KOUASSIANIAGUINI",
  "KIMBIRILASUD" = "KIMBIRILA SUD",
  "GUEZONDUEKOUE" = "GUEZON",
  "GRANDLAHOU" = "GRAND LAHOU",
  "GRANDZATTRY" = "GRANDZATRY",
  "GAGORE" = "LAKOTA",
  "GODOUKO" = "GOUDOUKO",
  "GNAMANGUI" = "GNANMANGUI",
  "DAIRODIDIZO" = "DAIRO DIDIZO",
  "DIBRIASRIKRO" = "DIBRIASSIRIKRO",
  "DIARABANA" = "BOBIDIARABANA",
  "BEDYGOAZON" = "BEDIGAOZON",
  "BILIMORO" = "BILIMONO",
  "BOBI" = "BOBIDIARABANA",
  "ARRHA" = "ARRAH"
)
```

```r
# Appliquons les corrections sur la colonne commune_clean
ehcvm_civ_test$commune_clean <- sapply(ehcvm_civ_test$commune_clean, function(x) {
  if (x %in% names(corrections)) {
    return(corrections[[x]])
  } else {
    return(x)
  }
})
```

```r
# Cela étant fait, chargeons les données HDX
data_hdx <- read_excel("civ_adminboundaries_tabulardata.xlsx")

# On copie les données pour les tests
data_hdx_test <- data_hdx
# On sélectionne des colonnes pertinentes
data_hdx_test <- data_hdx_test %>% select(c("ADM3_FR","ADM3_PCODE"))
```

```r
# On ajoute une colonne nettoyée pour les communes de HDX en appliquant
# la fonction qu'on a définit ci-dessus.
data_hdx_test$commune_clean <- sapply(data_hdx_test$ADM3_FR, clean_text)
```

```r
# Trouvons les communes communes dans les deux bases (ehcvm et hdx)
communes_communes <- intersect(ehcvm_civ_test$commune_clean, data_hdx_test$commune_clean)
```

```r
# Trouvons les communes qui sont dans ehcvm et qui ne sont pas dans communes_communes
# L'objectif est de vérifier si le merge a bien été fait.
communes_uniques_ehcvm <- setdiff(ehcvm_civ_test$commune_clean, communes_communes)
print(communes_uniques_ehcvm)
```

```
## character(0)
```

```r
# Pour ne garder que les communes de façon unique,
# nous allons filtrer les bases pour ne garder que les communes communes
ehcvm_civ_test <- ehcvm_civ_test %>% filter(commune_clean %in% communes_communes)
data_hdx_test <- data_hdx_test %>% filter(commune_clean %in% communes_communes)
```

```r
# On merge les bases en conservant uniquement les communes communes ax deux bases
merged_data <- inner_join(ehcvm_civ_test, data_hdx_test, by = "commune_clean")
```

```r
# Ici on crée une nouvelle base contenant une seule occurrence par commune
merged_data_unique <- merged_data %>% distinct(commune_clean, .keep_all = TRUE)
```

##————————————Méthode2————————————

```r
library(haven)
library(readxl)
library(labelled)
library(stringi)
library(stringr)
library(tibble)
library(dplyr)
library(stringdist)
```

```
## Warning: le package 'stringdist' a été compilé avec la version R 4.4.2
```

```r
#######AUTRE METHODE POUR FUSIONNER LES BASES POUR OBTENIR LE CODE DU NIVEAU 3
#les bases sont réimportées pour ramener les bases à leurs états initiaux
# Chargement des données
hdx <- read_excel("civ_adminboundaries_tabulardata.xlsx", sheet = "ADM3")
ehcvm <- read_dta("ehcvm_individu_civ2021.dta")

hdx <- hdx %>% select(c("ADM3_FR","ADM3_PCODE"))
```

```r
# Nettoyage des données
ehcvm <- ehcvm %>%
  mutate(
    region = to_factor(region),
```

```
    departement = to_factor(departement),
    sp_commune = to_factor(sp_commune),
    milieu = to_factor(milieu)
  ) %>%
  select(country, region, departement, sp_commune, everything())
```

```
# Enlever les majuscules
ehcvm <- ehcvm %>%
  mutate(sp_commune = tolower(sp_commune))

hdx <- hdx %>%
  mutate(ADM3_FR = tolower(ADM3_FR))
```

```
# Fonction pour nettoyer les textes
clean_text_2 <- function(x) {
  x <- str_trim(x)
  x <- str_squish(x)
  x <- stri_trans_general(x, "Latin-ASCII")
  return(x)
}
```

```
# Appliquer le nettoyage aux deux bases
ehcvm <- ehcvm %>%
  mutate(sp_commune = clean_text_2(sp_commune))

hdx <- hdx %>%
  mutate(ADM3_FR = clean_text_2(ADM3_FR))
```

```
# Vecteur de référence (communes officielles, ici remplacé par hdx$ADM3_FR)
communes_officielles_clean <- hdx$ADM3_FR
```

```
# Nettoyage de la base de données my_data (assume que tu l'as chargée)
my_data <- ehcvm
```

```
# Calcul de la meilleure correspondance avec la distance de Levenshtein
my_data <- my_data %>%
  rowwise() %>%
  mutate(
    BestMatch = communes_officielles_clean[which.min(
      stringdist(sp_commune, communes_officielles_clean, method = "lv")
    )],
    MinDistance = min(
      stringdist(sp_commune, communes_officielles_clean, method = "lv")
    )
  ) %>%
  select(country, region, departement, sp_commune,BestMatch,MinDistance, everything())%>%
  ungroup()

no_match <- setdiff(my_data$sp_commune, my_data$Bestmatch)
```

```
## Warning: Unknown or uninitialised column: `Bestmatch`.
```

```
# Traitement du résultat : Normalisation si la distance est faible
my_data <- my_data %>%
  mutate(Commune_Standard = ifelse(MinDistance <= 3, BestMatch, NA))
my_data <- my_data %>%
```

```r
  mutate(Commune_Standard = case_when(
    sp_commune == "guezon_duekoue" ~ "guezon",
    sp_commune == "diarabana" ~ "bobi-diarabana",
    sp_commune == "bobi"~ "bobi-diarabana",
    MinDistance <= 3 ~ BestMatch,
    TRUE ~ NA_character_
  ))
```

```r
#Merge des deux bases avec Commune_Standard
# Fusionner les bases sur la variable d'identification pour obtenir la nouvelle base
#finale merged_data_2
merged_data_2 <- my_data %>%
  left_join(hdx %>% select(ADM3_FR, ADM3_PCODE),
            by = c("Commune_Standard" = "ADM3_FR"))%>%
  select(country, region, departement, sp_commune, Commune_Standard, ADM3_PCODE, everything())
```

```r
# Créer une nouvelle base contenant une seule occurrence par commune
merged_data_2_unique <- merged_data_2 %>% distinct(Commune_Standard, .keep_all = TRUE)

## Voir les  communes de ehcvm qui ne sont pas dans HDX L'objectif est qu'il retourne
# une liste vide montrant donc que toutes les communes de ehcvm ont trouvé leur correspondance
#dans HDX

commune_ehcvm <- setdiff(merged_data_2$Commune_Standard, hdx$ADM3_FR)
print(commune_ehcvm)
```

```
## character(0)
```