ARSIA Notebook

DATACLEANING FOR ARSIA

Load the libraries

```
# lib = where to save the library  (zonder = default)

package_list <- c("data.table", "tidyverse", "naniar", "stringr", "readr",  "dplyr", "magrittr", "readxl
                  )

for (pkg in package_list) {
  if (pkg %in% rownames(.packages()) == FALSE)
  {library(pkg, character.only = TRUE)}

}
```

##Function to apply SHA-256 hashing

```
# Function to apply SHA-256 hashing
sha256_hash <- function(data) {
  openssl::sha256(data)
}
```

Data loading

```
barometer_dt_raw <- readxl::read_excel("../Data/ARSIA/ARSIA_DECIDE_20221201.xlsx")
```

Data manipulation

```
barometer_dt <- barometer_dt_raw %>%
  dplyr::rename(
    Dossier = 'N° échantillon',
    Date = 'Date of Sample',
    Sample_type= 'Sample Type',
    Diagnostic_test = METH,
    Farm_ID = TRP,
    PM = P_multocida,
    MH = M_haemolytica,
    HS = H_somnus,
    MB = M_bovis,
    BRSV = BRSV,
    PI3 = PI3,
    BCV = Coronavirus
    ) %>%
  tidyr::separate(ADDRESS, c('Postal_code', 'City')) %>%
  dplyr::mutate(
    Postal_code = as.double(Postal_code),
```

```
  Filenumber = str_sub(Dossier, 1, 12),
  Samplenumber = str_sub(Dossier, -3),
  Country ='Belgium',
  Lab_reference ='3',
  Sample_type = case_when(
    Sample_type == "BAL" ~ 'BAL',
    Sample_type == "SWAB" ~'Swab',
    Sample_type == "CARCASS" ~ 'Autopsy',
    TRUE ~ 'Missing'
  ),
  Breed = case_when(
    SPECUL == "MEAT" ~ 'Beef',
    SPECUL == "MILK" ~'Dairy',
    SPECUL == "MXD" ~ 'Mixed',
    TRUE ~ 'Unknown'
  ),
  Province = case_when(
    between(Postal_code, 1000, 1299) ~ 'Brussels Hoofdstedelijk Gewest',
    between(Postal_code, 1300, 1499) ~ 'Waals-Brabant',
    between(Postal_code, 1500, 1999) ~ 'Vlaams-Brabant',
    between(Postal_code, 3000, 3499) ~ 'Antwerpen',
    between(Postal_code, 2000, 2999) ~ 'Limburg',
    between(Postal_code, 3500, 3999) ~ 'Limburg',
    between(Postal_code, 4000, 4999) ~ 'Luik',
    between(Postal_code, 5000, 5999) ~ 'Namen',
    between(Postal_code, 6000, 6599) ~ 'Henegouwen',
    between(Postal_code, 7000, 7999) ~ 'Henegouwen',
    between(Postal_code, 6600, 6999) ~ 'Luxemburg',
    between(Postal_code, 8000, 8999) ~ 'West-Vlaanderen',
    TRUE ~ 'Oost-Vlaanderen'
  )

)%>%
dplyr::select(
  Filenumber,
  Diagnostic_test,
  Samplenumber,
  Country,
  Lab_reference,
  Sample_type,
  Breed,
  PM,
  MH,
  HS,
  MB,
  BRSV,
  PI3,
  BCV,
  Date,
  Postal_code,
  Province,
  Farm_ID
) %>%
```

```
  dplyr::distinct() %>%
  dplyr::mutate(
    Filenumber = sha256_hash(as.character(Filenumber)),
    Samplenumber = sha256_hash(as.character(Samplenumber))
  )
```

Floor date to 1st of month

```
barometer_dt$Floored_date <- lubridate::floor_date(barometer_dt$Date, "month")
```

Aggregate data based on farm_ID and month (WIDE)

```
barometer_groupby <- barometer_dt %>%
  group_by(Lab_reference, Country, Breed, Floored_date, Province, Farm_ID, Diagnostic_test, Sample_type
  summarise(across(c(PM, MH, HS, MB, BRSV, PI3, BCV), max))
```

Convert to LONG

```
barometer_long <- barometer_groupby %>%
  tidyr::pivot_longer(
    cols = c('PM', 'MH', 'HS', 'MB', 'BRSV', 'PI3', 'BCV'),
    names_to = 'Pathogen',
    values_to = 'Result',
  )
```

Save file to csv (long version)

```
write.csv(barometer_long, "../Data/CleanedData/barometer_ARSIA.csv", row.names=TRUE)
```

Write to excel

```
writexl::write_xlsx(barometer_dt, "barometer_wide_ARSIA.xlsx")
```