DGZ Notebook

Data cleaning for DGZ

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/DGZ/DECIDE_MTA_UGENT_14nov2022.xlsx")
barometer_aero_cult_raw <- readxl::read_excel("../Data/DGZ/DECIDE_MTA_UGENT_BAC_AERO_14nov2022.xlsx")
barometer_myco_cult_raw <- readxl::read_excel("../Data/DGZ/DECIDE_MTA_UGENTBAC_MYCO_14nov2022.xlsx")
```

Data manipulation AEROBIC CULTURE results

```
barometer_aero_cult <- barometer_aero_cult_raw %>%
  dplyr::rename(
    Filenumber = Dossiernummer,
    Pathogen_identification = 'KIEMSTAAL IDENTIFICATIE',
    Pathogen_result = 'KIEMSTAAL RESULTAAT',
    Samplenumber = 'Staalnummer'
  ) %>%
  dplyr::mutate(
    Parameter_code = 'BAC_AERO',
    Result = 'OK'
  ) %>%
  dplyr::select(
    Filenumber,
    Pathogen_identification,
    Pathogen_result,
    Parameter_code,
```

```
    Samplenumber,
    Result
  ) %>%
  dplyr::filter(
    Pathogen_identification %in% c("Pasteurella multocida","Mannheimia haemolytica","Histophilus somni"
  ) %>%
  dplyr::distinct() %>%
  dplyr::mutate(
    Filenumber_anon = sha256_hash(as.character(Filenumber)),
    Samplenumber_anon = sha256_hash(as.character(Samplenumber))
  ) %>%
  dplyr::select(-Filenumber, -Samplenumber)
```

Intermediate table is needed

```
df_samples <- data.frame(
  Result = c('OK', 'OK', 'OK', 'OK'),
  Parameter_code = c('BAC_AERO','BAC_AERO','BAC_AERO', 'BAC_MYCOPLASMA'),
  Diagnostic_test= c('Culture','Culture','Culture', 'Culture'),
  Pathogen_identification=c("Pasteurella multocida","Mannheimia haemolytica","Histophilus somni", 'Myco
)
```

Data manipulation MYCOPLASMA CULTURE results

```
# Data manipulation MYCOPLASMA CULTURE results
barometer_myco_cult <- barometer_myco_cult_raw %>%
  dplyr::rename(
    Filenumber = Dossiernummer,
    Pathogen_identification = 'KIEMSTAAL IDENTIFICATIE',
    Mycoplasma_result = 'KIEMSTAAL RESULTAAT',
    Samplenumber = 'Staalnummer'
  ) %>%
  dplyr::mutate(
    Parameter_code = 'BAC_MYCOPLASMA',
    Result = 'OK'
  ) %>%
  dplyr::select(
    Filenumber,
    Pathogen_identification,
    Mycoplasma_result,
    Parameter_code,
    Samplenumber,
    Result
  ) %>%
  dplyr::filter(
    Pathogen_identification %in% c("Mycoplasma bovis")
  ) %>%
  dplyr::distinct() %>%
  dplyr::mutate(
    Filenumber_anon = sha256_hash(as.character(Filenumber)),
    Samplenumber_anon = sha256_hash(as.character(Samplenumber))
  ) %>%
  dplyr::select(-Filenumber, -Samplenumber)
```

Data manipulation PCR results

```
barometer_dt <- barometer_dt_raw %>%
  dplyr::rename(
    Filenumber=Dossiernummer,
    Samplenumber = Staalnummer,
    Sample_type = Staaltype,
    Parameter_code = PARAMETER_CODE,
    Pathogen = Onderzoek,
    Result = Resultaat,
    Date = Creatiedatum,
    Postal_code = Postcode,
    Farm_ID = ANON_ID
    ) %>%
  dplyr::mutate(
    Country='Belgium',
    Diagnostic_test = case_when(
      Parameter_code %in% c('BAC_AERO','BAC_MYCOPLASMA') ~ 'Culture',
      TRUE ~ 'PCR'
    ),
    Lab_reference='1',
    Sample_type = case_when(
      Sample_type == "RU Broncho-alveolar lavage (BAL)" ~ 'BAL',
      Sample_type == "RU Anderen" ~'Unknown',
      Sample_type %in% c("RU Swabs", "RU Swab", 'RU Neusswab', 'RU Neusswabs') ~ 'Swab',
      Sample_type %in% c("RU Kadaver", "RU Organen") ~ 'Autopsy',
      TRUE ~ 'Missing'
    ),
    Breed = case_when(
      Bedrijfstype == 'VCALF' ~ 'Veal',
      is.na(MEAT) ~ 'Unknown',
      (as.numeric(MEAT)/as.numeric(TOTAL))>0.9 ~ 'Beef',
      (as.numeric(MILK)/as.numeric(TOTAL))>0.9 ~ 'Dairy',
      TRUE ~ 'Mixed'
    ),
    Pathogen = case_when(
      Pathogen %in% c(
        "AD Pasteurella multocida Ag (PCR)",
        "AD Pasteurella multocida Ag pool (PCR)",
        "AD P. multocida Ag (PCR)",
        "AD P. multocida Ag pool (PCR)") ~ 'Pasteurella multocida',
      Pathogen %in% c(
        "AD Mannheimia haemolytica Ag (PCR)",
        "AD Mannheimia haemolytica Ag pool (PCR)") ~ 'Mannheimia haemolytica',
      Pathogen %in% c("RU PI3 Ag (PCR)", "RU PI3 Ag pool (PCR)") ~ 'PI3',
      Pathogen %in% c("RU BRSV Ag (PCR)", "RU BRSV Ag pool (PCR)") ~ 'BRSV',
      Pathogen %in% c(
        "AD Histophilus somnus (PCR)",
        "AD Histophilus somnus Ag (PCR)",
        "AD Histophilus somnus Ag pool (PCR)",
        "AD Histophilus somni Ag (PCR)",
        "AD Histophilus somni Ag pool (PCR)") ~ 'Histophilus somni',
      Pathogen %in% c(
        "RU Mycoplasma bovis (PCR)",
```

```
        "RU Mycoplasma bovis Ag pool (PCR)",
        "RU Mycoplasma bovis Ag (PCR)") ~ 'Mycoplasma bovis',
      Pathogen %in% c("AD Corona Ag (PCR)", "AD Corona Ag pool (PCR)") ~ 'BCV'
    ),
      Province = case_when(
        between(as.numeric(Postal_code), 1000, 1299) ~ 'Brussels',
        between(as.numeric(Postal_code), 1300, 1499) ~ 'Walloon Brabant',
        between(as.numeric(Postal_code), 1500, 1999) ~ 'Flemish Brabant',
        between(as.numeric(Postal_code), 3000, 3499) ~ 'Antwerp',
        between(as.numeric(Postal_code), 2000, 2999) ~ 'Limburg',
        between(as.numeric(Postal_code), 3500, 3999) ~ 'Limburg',
        between(as.numeric(Postal_code), 4000, 4999) ~ 'Liège',
        between(as.numeric(Postal_code), 5000, 5999) ~ 'Namur',
        between(as.numeric(Postal_code), 6000, 6599) ~ 'Hainaut',
        between(as.numeric(Postal_code), 7000, 7999) ~ 'Hainaut',
        between(as.numeric(Postal_code), 6600, 6999) ~ 'Luxembourg',
        between(as.numeric(Postal_code), 8000, 8999) ~ 'West Flanders',
        TRUE ~ 'East Flanders'
            )
  ) %>%
  dplyr::select(
    Filenumber,
    Diagnostic_test,
    Samplenumber,
    Country,
    Lab_reference,
    Sample_type,
    Breed,
    Parameter_code,
    Result,
    Pathogen,
    Date,
    Province,
    Farm_ID
  ) %>%
  dplyr::distinct() %>%
  dplyr::mutate(
    Filenumber_anon = sha256_hash(as.character(Filenumber)),
    Samplenumber_anon = sha256_hash(as.character(Samplenumber))
  ) %>%
  dplyr::select(-Filenumber, -Samplenumber)
```

Join all three files

```
barometer <-
  barometer_dt %>%
  dplyr::left_join(df_samples, by = c('Diagnostic_test','Result', 'Parameter_code')) %>%
  dplyr::left_join(
      barometer_aero_cult, by = c('Filenumber_anon', 'Samplenumber_anon', 'Result', 'Parameter_code', '
      ) %>%
  dplyr::left_join(
      barometer_myco_cult, by = c('Filenumber_anon', 'Samplenumber_anon', 'Result', 'Parameter_code', '
  ) %>%
```

```
  dplyr::mutate(
    Floored_date = lubridate::floor_date(Date, "month"),
      Pathogen = case_when(

      (Pathogen == 'Pasteurella multocida') ~ 'PM',
      (Pathogen == 'Histophilus somni') ~ 'HS',
      (Pathogen == 'Mannheimia haemolytica') ~ 'MH',
      (Pathogen == 'Mycoplasma bovis') ~ 'MB',
      TRUE ~ Pathogen
    ),
    Pathogen = case_when(
      (Pathogen_identification == 'Pasteurella multocida') ~ 'PM',
      (Pathogen_identification == 'Histophilus somni') ~ 'HS',
      (Pathogen_identification == 'Mannheimia haemolytica') ~ 'MH',
      (Pathogen_identification == 'Mycoplasma bovis') ~ 'MB',
      TRUE ~ Pathogen
    ),
      Result = case_when(
        Result %in% c("Twijfelachtig (PCR)", "POSITIEF", "GEDETECTEERD", "GEDETECTEERD (sterk)", "GEDETE
        "GEDETECTEERD (matig)","GEDETECTEERD (zeer sterk)", "GEDETECTEERD (zeer zwak)") ~ 1,
        Result %in% c("negatief", "Niet gedetecteerd") ~ 0,
        Result %in% c("NI", "niet interpreteerbaar", "Inhibitie") ~ as.numeric(NA),
      Parameter_code == 'BAC_AERO' & is.na(Pathogen_result) ~ 0,
      Parameter_code == 'BAC_AERO' & !is.na(Pathogen_result) ~ 1,
      Parameter_code == 'BAC_MYCOPLASMA' & is.na(Mycoplasma_result) ~ as.numeric(NA),
      Parameter_code == 'BAC_MYCOPLASMA' & Mycoplasma_result == 'neg' ~ 0,
      Parameter_code == 'BAC_MYCOPLASMA' & sjmisc::str_contains(Mycoplasma_result, 'POS') ~ 1,
      TRUE ~ as.numeric(NA)
    )
  ) %>%
  group_by(
    Lab_reference,
    Country,
    Breed,
    Floored_date,
    Province,
    Farm_ID,
    Diagnostic_test,
    Sample_type,
    Pathogen
    ) %>%
  summarise(across(c(Result), max))
```

Save file (long version)

```
write.csv(barometer, "../Data/CleanedData/barometer_DGZ.csv", row.names=TRUE)
```