GD Notebook

DATACLEANING FOR GD

Load the libraries

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

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

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

}
```

##Function to apply SHA-256 hashing

```
sha256_hash <- function(data) {
  openssl::sha256(data)
}
```

Data loading

```
barometer_dt_raw <- readxl::read_excel("../Data/GD_opgeschoond/221122_data_RGD_DECIDE_nw.xlsx")
```

Data manipulation

```
barometer_dt <- barometer_dt_raw %>%
  dplyr::rename(
    Filenumber = Dossier_ID,
    Samplenumber = sample_id,
    Farm_ID = farm_ID,
    Project = project,
    Date = date
    ) %>%
  dplyr::mutate(
    Country ='The Netherlands',
    Lab_reference ='2',
    Sample_type = case_when(
      reason_of_sampling == 'Autopsy' ~ 'Autopsy',
      sample == 'BAL' ~ 'BAL',
      sample == 'SWABS' ~ 'Swab',
      sample == 'OTHER' ~ 'Unknown',
      TRUE ~ 'Missing'
    ),
```

```
    Diagnostic_test = case_when(
      test == 'PCR' ~ 'PCR',
      test == 'Kweek' ~ "Culture",
      TRUE ~ 'Missing'
    ),
    Breed = case_when(
      breed == "beef" ~ 'Beef',
      breed == "dairy" ~'Dairy',
      breed == "mixed" ~ 'Mixed',
      breed == "veal" ~ 'Veal',
      breed %in% c("other", "rearing", "unknown") ~ 'Unknown',
      TRUE ~ 'Unknown'
    ),
    Province = case_when(
      provincie == "DR" ~ 'Drenthe',
      provincie == "FL" ~'Flevoland',
      provincie == "FR" ~'Friesland',
      provincie == "GL" ~'Gelderland',
      provincie == "GR" ~'Groningen',
      provincie == "LB" ~'Limburg',
      provincie == "NB" ~'North Brabant',
      provincie == "NH" ~'North Holland',
      provincie == "OV" ~'Overijssel',
      provincie == "UT" ~'Utrecht',
      provincie == "ZH" ~'South Holland',
      provincie == "ZL" ~'Zeeland',
      TRUE ~ 'Missing'
    )

  )%>%
  dplyr::select(
    Filenumber,
    Diagnostic_test,
    Samplenumber,
    Country,
    Lab_reference,
    Sample_type,
    Breed,
    PM,
    MH,
    HS,
    MB,
    BRSV,
    PI3,
    BCV,
    Date,
    Province,
    Project,
    Farm_ID
  ) %>%
  dplyr::distinct() %>%
dplyr::mutate(
    Filenumber = sha256_hash(as.character(Filenumber)),
```

```
    Samplenumber = sha256_hash(as.character(Samplenumber)),
    Farm_ID = sha256_hash(as.character(Farm_ID))
    )
```

Filter data for 'monitoring' and 'no projects'

```
barometer_dt_filtered <- filter(barometer_dt, Project == 'monitoring' | Project == 'no project')
```

Floor date to 1st of the month

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

Aggregate data based on farm_ID and month (WIDE)

```
barometer_groupby <- barometer_dt_filtered %>%
  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_GD.csv", row.names=TRUE)
```

Write to excel

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