Loading_HMRC-data-trial_DELETE-LATER

B. Scheliga, A. Lofstedt

2022-10-26

Part ???: Cleaning HMRC Trade data to map UK seafood supply chains

This R Markdown document outlines how the HRMC trade data was complied and cleaned. The HRMC trade data ranges from 2009 - 2019.

The justification for the data included in this data set can be found in the supporting excel document.

Preparation

```
# It is good practice to load all needed libraries in the beginning of the scripted
#library(here)
library(tidyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v dplyr   1.0.9
## v tibble  3.1.8     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(vroom) #for loading and transforming data
library(data.table) # for fread()-function
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(mice) # md.pattern to show missing data
```

```
##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
#Loading file path to project folder
source("Data_filepath.R")# Data_filepath.R is listed in .gitignore-file. So, you will need to create th
```

Loading the HMRC trade data

```
############ HMRC Trade Data

# Filepath to RawData
filepath <- paste(data_dir,"RawData/csv-files", sep="")
# reading the HRMC trade data dataset (.csv)
#df_HRMC_2009 <- vroom(file=paste(filepath,"HMRC_UK_trade-2009.csv", sep="/"))
vec_filenames_HMRC <- list.files(filepath, pattern = "HMRC_UK_trade", full.names = TRUE)



print("The following files were loaded:")
```

```
## [1] "The following files were loaded:"
```

```
i = 1
# Loading the cleaned dataset
for(f in vec_filenames_HMRC){

    temp <- fread(f) # storing the data in a temporary
    assign(paste("df_HMRC_",i,sep=""),temp) # assigning the df a name based on input dataset
    print(paste(i,". ",sub(paste(".*",filepath,sep=""),"",f),sep="")) # print out which dataset files h
    i = i+1
    rm(temp)# removing temp object
}
```

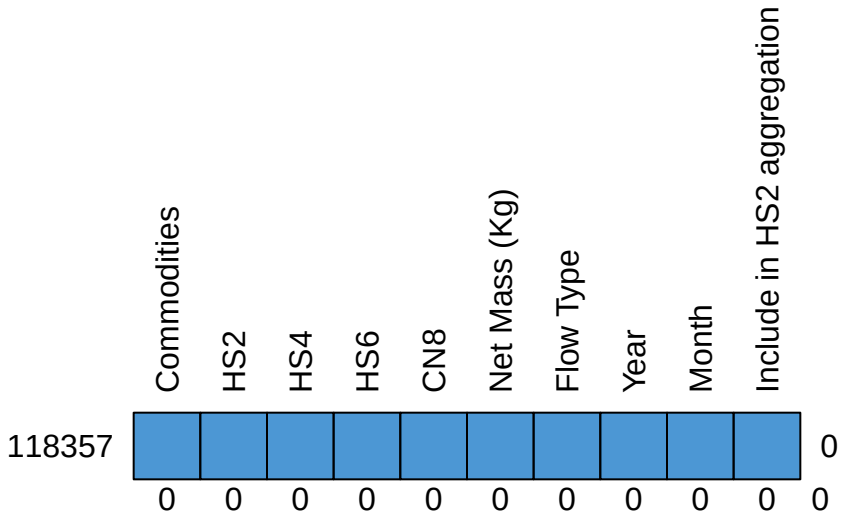
```
## [1] "1. /HMRC_UK_trade-2009.csv"
## [1] "2. /HMRC_UK_trade-201011.csv"
## [1] "3. /HMRC_UK_trade-201213.csv"
## [1] "4. /HMRC_UK_trade-201415.csv"
## [1] "5. /HMRC_UK_trade-201617.csv"
## [1] "6. /HMRC_UK_trade-201819.csv"
```

```
rm(i)

# Next step is to combine this all into big HMRC data.frame
df_HRMC_2009_to_2019 <- bind_rows(df_HMRC_1, df_HMRC_2, df_HMRC_3, df_HMRC_4, df_HMRC_5, df_HMRC_6)

rm(df_HMRC_1, df_HMRC_2, df_HMRC_3, df_HMRC_4, df_HMRC_5, df_HMRC_6) # removing the unneeded df
```

```
md.pattern(df_HRMC_2009_to_2019, rotate.names = TRUE)
```

Checking for missing values

```
##  /\     /\
## {  `---'  }
## {  0   0  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##        Commodities HS2 HS4 HS6 CN8 Net Mass (Kg) Flow Type Year Month
## 118357           1   1   1   1   1             1         1    1     1
##                  0   0   0   0   0             0         0    0     0
```

```
##        Include in HS2 aggregation
## 118357                          1 0
##                                 0 0
```

Loading the EUMOFA data

We use the EUMOFA data as a help to facilitate the classification and translation of the HRMC CN-8 codes in to our desired species and species type. The EUMOFA has a classification mapped to each CN-8 code. Using this, instead of crawling through the HMRC "Product name"-column and searching for specific key words its text, will mitigate a lot of potential miss-classification. As the HMRC "Product name", does not seem to use controlled vocabulary and has various spelling version of the same word (plural & single) included.

[MAYBE MORE DETAILS ON EUMOFA DATA HERE]

```
############### EUMOFA-file

# Filepath to EUMOFA-file with complete CN-8 codes
filepath <- paste(data_dir,"Methods/SpeciesTypeClassification", sep="")
# reading the EUMOFA-file
df_EUMOFA_CN8 <- vroom(file=paste(filepath,"EUMOFA_CN-8-values.csv", sep="/"))
```

```
## Rows: 9386 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Year of Reg, CN-8, Comment, CN-8 product name, Explanation
## dbl (2): Year, CF
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Note: the CN8 codes are related to the respective year, and have change over time.

# We are also loading the Annex 4 from the Metadata 2 - Data management EUFOMA https://www.eumofa.eu/su
# ANNEX 4 Correlation between Main commercial species(MCS)/Commodity Groups (CG) and CN-8 from 2001 to

df_EUMOFA_CN8_MCS_CG <- vroom(file=paste(filepath,"EUMOFA_Annex4.csv", sep="/"))
```

```
## Rows: 9917 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): CN8 code, Description, PS, PR, MCS_code, MCS_descr, CG code, CG
## dbl (1): Year
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# loading Species classification - translation from EUMOFA MCS & CG to the classification we want to us
df_Species_Class <- vroom(file=paste(filepath,"SpeciesTypeClassificationCode.csv", sep="/"))
```

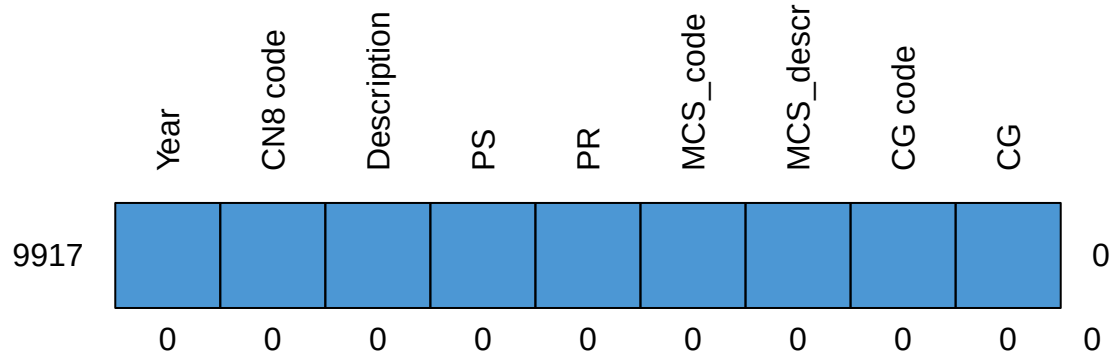
```
## Rows: 103 Columns: 4
## -- Column specification --------------------------------------------------------
```

```
## Delimiter: ","
## chr (4): EUMOFA_MCS, EUMOFA_CG, EUMOFA_MCS_AL, SpeciesType_AL
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
md.pattern(df_EUMOFA_CN8_MCS_CG, rotate.names = TRUE)
```

Checking for missing values

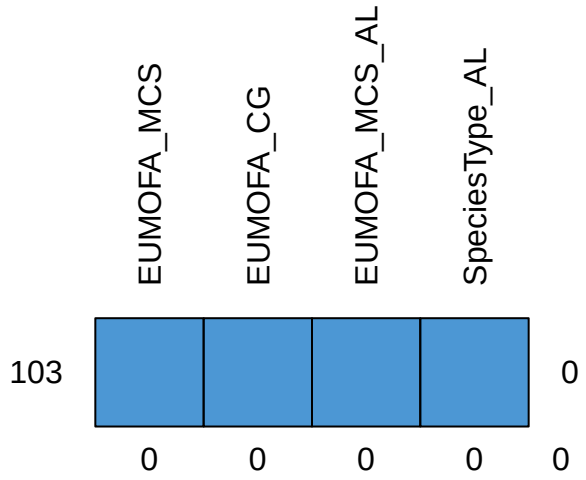
```
##  /\     /\
## {  `---'  }
## {  0   0  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##      Year CN8 code Description PS PR MCS_code MCS_descr CG code CG
## 9917    1        1           1  1  1        1         1       1  1 0
##         0        0           0  0  0        0         0       0  0 0
```

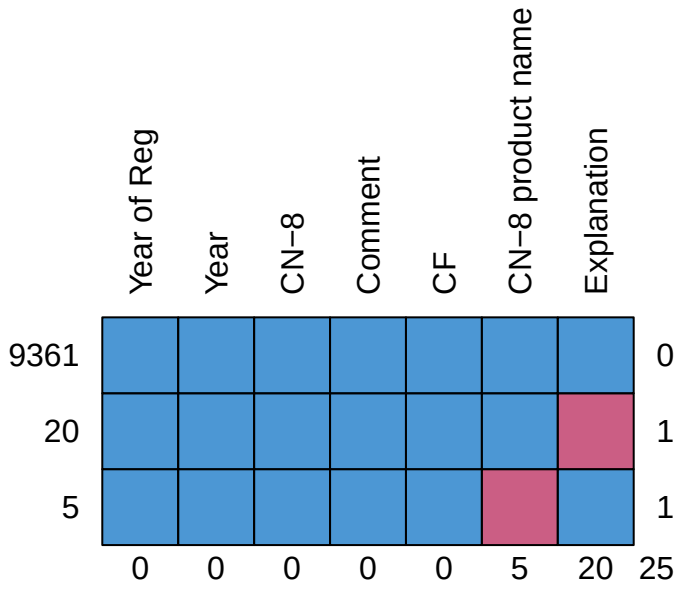
```
md.pattern(df_Species_Class, rotate.names = TRUE)
```

```
##  /\     /\
## {  `---'  }
## {  0   0  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##     EUMOFA_MCS EUMOFA_CG EUMOFA_MCS_AL SpeciesType_AL
## 103          1         1             1              1 0
##              0         0             0              0 0
```

```
md.pattern(df_EUMOFA_CN8, rotate.names = TRUE)
```




```
##      Year of Reg Year CN-8 Comment CF CN-8 product name Explanation
## 9361           1    1    1       1  1                 1           1  0
## 20             1    1    1       1  1                 1           0  1
## 5              1    1    1       1  1                 0           1  1
##                0    0    0       0  0                 5          20 25
```

We have 25 missing (NA) values in the EUMOFA dataset, 20 in the "Explanation"-column and 5 in the "CN-8 product name"-column. As annoying as NA-values are, in this instance we can ignore them. As these values are irrelevant for us and the next steps, since the EUMOFA Annex 4 dataset has no NA-values and the main commercial species (MCS) and commodity groups (CG) for the respective CN-8 code.

Processing the data

```
# need to remove the space from the 'CN-8'-column. But leave it as character, so we don't loose the "0"
# https://stackoverflow.com/questions/20309876/r-how-to-replace-in-a-string

df_EUMOFA_CN8$`CN-8` <- gsub("\\ ","", df_EUMOFA_CN8$`CN-8`)

#df_EUMOFA_CN8_2009 <- df_EUMOFA_CN8 %>% filter(Year %in% "2009" )

print("FYI: In the EUMOFA CN-8 product name-Column are large - symbols, which R does not recognize repla
```

```
## [1] "FYI: In the EUMOFA CN-8 product name-Column are large - symbols, which R does not recognize repl
```

```r
# Need to separate the CN8 code from the description in the CN8-column of the HMRC data
df_HRMC_2009_to_2019$Subset_aid_CN8 <- gsub(" .*$", "", df_HRMC_2009_to_2019$CN8) # subset string befor

# Thanks to our psychic abilities, we know that one of the CN8-codes in the HRMC Uk trade is missing it

# Check what will not be joined
df_HMRC_anti <- df_HRMC_2009_to_2019 %>% select('Net Mass (Kg)', 'Flow Type', 'Year','Month', 'Subset_a
#The following CN8 code items were not joined
unique(df_HMRC_anti$CN8)
```

```
##  [1] "03 HS2 Below Threshold Trade"
##  [2] "3074959"
##  [3] "03076000 Snails, live, fresh, chilled, frozen, salted, dried or in brine, even smoked, with or
##  [4] "05119910 Sinews or tendons of animal origin, parings and similar waste of raw hides or skins"
##  [5] "05119931 Raw natural sponges of animal origin"
##  [6] "05119939 Natural sponges of animal origin (excl. raw)"
##  [7] "05119985 Animal products, n.e.s.; dead animals, unfit for human consumption (excl. fish, crusta
##  [8] "23011000 Flours, meals and pellets, of meat or offal, unfit for human consumption; greaves"
##  [9] "03076010 Snails, smoked, even in shell, even cooked but not otherwise prepared (excl. sea snail
## [10] "03076090 Snails, live, fresh, chilled, frozen, salted, dried or in brine, even in shell (excl.
## [11] "16055800 Snails, prepared or preserved (excl. smoked and sea snails)"
```

It is item [2] "3074959", were we need to add a lead zero.

```r
# adding the lead 0 to "3074959" in the HRMC UK trade data
df_HRMC_2009_to_2019$Subset_aid_CN8 <- str_replace(df_HRMC_2009_to_2019$Subset_aid_CN8,"3074959","03074

# the "Flow Type" is also separate into "EU" and "Non-EU" Imports and Exports
df_HRMC_2009_to_2019$`Flow Type` <- str_replace(df_HRMC_2009_to_2019$`Flow Type`,"Non EU - ","") # need
df_HRMC_2009_to_2019$`Flow Type` <- str_replace(df_HRMC_2009_to_2019$`Flow Type`,"EU - ","")


# Now, we can join the HMRC trade data and the EUMOFA data
df_HMRC_inner <- df_HRMC_2009_to_2019 %>% select('Net Mass (Kg)', 'Flow Type', 'Year','Month', 'Subset_a
```

```r
# calculating annual sum, selecting needed columns and converting to Net Mass from kg to 1000 tonnes
df_HRMC.sum <- df_HMRC_inner %>% group_by(CN8 = Subset_aid_CN8,`Product name` =`CN-8 product name`, Yea
```

```
## `summarise()` has grouped output by 'CN8', 'Product name', 'Year', 'Commodity'.
## You can override using the `.groups` argument.
```

```r
rm(df_HMRC_inner)

df_HMRC_4DB <-df_HRMC.sum
```

```r
# mapping EUMOFA Main Commercial Species (MCS) and Commodity Group (CG) classification
df_HMRC_4DB <- df_HMRC_4DB %>% inner_join(., df_EUMOFA_CN8_MCS_CG, by = c('CN8' = 'CN8 code', 'Year'))

# Mapping our desired Species Species Type classification
df_HMRC_4DB <- df_HMRC_4DB %>% inner_join(., df_Species_Class, by = c('MCS_descr' = 'EUMOFA_MCS'))
```

```r
# We will now need to aggregated the weight values for some species again. As we have aggregate some so

df_HRMC.sum <- df_HMRC_4DB %>% group_by(Species = EUMOFA_MCS_AL, SpeciesType = SpeciesType_AL, Year, Co
```

```
## `summarise()` has grouped output by 'Species', 'SpeciesType', 'Year'. You can
## override using the `.groups` argument.
```

```r
df_HMRC_4DB <- df_HRMC.sum
```

```r
# Adding DataSupplier information
df_HMRC_4DB$DataSupplier <- "HMRC"
df_HMRC_4DB$DataSet <- "HMRC Overseas Trade data table - UK Trade Info"
#determining if the Fish is for Human consumption or not based on the CF from the EUMOFA-file
df_HMRC_4DB <- rename(df_HMRC_4DB, Value =`Net Mass (1000 t)`)
df_HMRC_4DB$Units <- "1000 tonnes"
df_HMRC_4DB$TemporalResolution <- "Annual"
df_HMRC_4DB$Flag <- "EXAMPLES: UC, SSTAlig"
df_HMRC_4DB$FlagDescription <- "EXAMPLES:Units changed, Species & Species Type aligned"
```

```r
# Selecting Columns
df_HMRC_4DB <- df_HMRC_4DB %>% ungroup %>%  select(DataSupplier, DataSet, Commodity, Species, SpeciesTyp

# Removing Non-food uses im- & exports
df_HMRC_4DB <- df_HMRC_4DB %>%  filter(!SpeciesType %in% "OtherNFU")

filepath <- paste(data_dir,"ProcessedData", sep="")

write.csv(df_HMRC_4DB, paste(filepath,"TradeData_HMRC_Preliminary-Cleaned.csv", sep="/"), row.names = FA
```