Carbon Footprint Exercise

Valentin Lucet

2023-10-14

```
library(readxl)
library(lubridate)
library(dplyr)
library(tidyr)
library(ggplot2)
```

Exercise Statement

A client sends us the following request:

To complete this exercise, you are invited to propose the best data representation based on your understanding ; you might have to look up on internet for units' conversions.

```
Dear Metrio Team,

We would like to see 2 data representations of our final GHG emissions (tonnes CO2eq).
Please provide 1 representation per countries, and 1 representation per year and energy
type.

Calculation formula:
Total GHG emissions (tCO2eq) = Energy consumption * Emission factors * Global Warming
Potential (GWP)

Thank you,

Marcel
```

Data ingestion and pre-processing

```
consumption <- read_xlsx("data/footprint_data.xlsx", sheet = "comsumption") %>%
  mutate(year = year(date))
structure <- read_xlsx("data/footprint_data.xlsx", sheet = "structure")
energy_scopes <- read_xlsx("data/footprint_data.xlsx", sheet = "energy_scopes")

EF_fuel <- read_xlsx("data/footprint_data.xlsx", sheet = "EF_fuel")
EF_elec <- read_xlsx("data/footprint_data.xlsx", sheet = "EF_elec")
energy_conversions <- read_xlsx("data/footprint_data.xlsx", sheet = "energy_conversions")
GWP <- read_xlsx("data/footprint_data.xlsx", sheet = "GWP")
```

Computations

```r
consumption_joined <- consumption %>%

  # Join to structure to get to country
  left_join(structure, by = "site") %>%

  # Collapse to useful dimensions
  group_by(country, year, site, energy_type, unit) %>%
  summarise(value = sum(value)) %>%
  ungroup() %>%

  # Convert MWh to kWh, gallons to Liters
  mutate(
    value = ifelse(unit == "MWh", value * 1e3, value),
    unit = ifelse(unit == "MWh", "kWh", unit)) %>%
  mutate(
    value = ifelse(unit == "Gallons", value * 3.78541, value),
    unit = ifelse(unit == "Gallons", "Liters", unit)
  ) %>%

  # Convert Liters to KWh based on the conversion table
  left_join(select(energy_conversions, -unit_to),
            by = c("energy_type", "unit")) %>%
  mutate(
    value = ifelse(!is.na(final_value) &
                     unit == "Liters", (value/initial_value)*final_value, value),
    unit = ifelse(!is.na(final_value) &
                     unit == "Liters", "kWh", unit)
  ) %>%
  select(-final_value, -initial_value) %>%

  # Convert propane and natural gas kwh to MMBTU
  mutate(
    value = ifelse(unit == "kWh" & energy_type %in% c("Propane", "Natural gas"),
                   value * 3.4121e-3, value),
    unit = ifelse(unit == "kWh" & energy_type %in% c("Propane", "Natural gas"),
                  "MMBtu", unit)
  ) %>%

  # Now that more things are in kWh/MMBtu, collapse to simplify
  group_by(country, year, site, energy_type, unit) %>%
  summarise(value = sum(value)) %>%
  ungroup() %>%

  # Multiply Electricity by corresponding EF to get emissions
  left_join(EF_elec, by = "country") %>%
  mutate(
    tonCO2e = ifelse(energy_type == "Electricity" &
                     unit == "kWh", (value * EF_kgCO2e_kwh)/1e3, NA)
  ) %>%
  select(-EF_kgCO2e_kwh) %>%

  # Join to the energy types and corresponding GWP and compute emissions
  left_join(EF_fuel, by = c("energy_type", "unit")) %>%
```

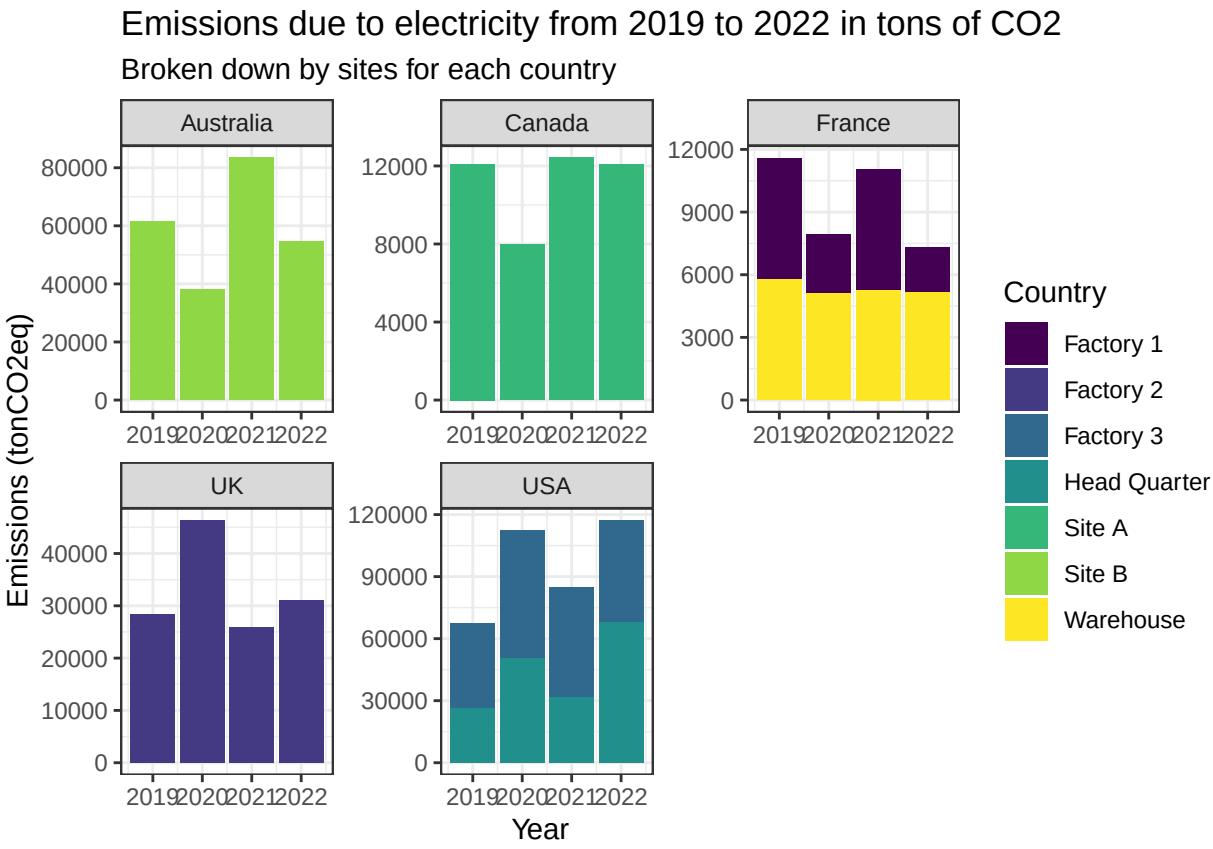
```r
  left_join(GWP, by = "GHG_type") %>%
  mutate(
    tonCO2e = ifelse(!is.na(GHG_type),
                     (value*gGHG_Unit * GWP_gCO2e_gGHG)/1e6, tonCO2e)
  ) %>%
  select(-gGHG_Unit, -GWP_gCO2e_gGHG, -GHG_type, -unit) %>%

  # Final collapse
  group_by(country, year, site, energy_type) %>%
  summarise(tonCO2e = sum(tonCO2e)) %>%
  ungroup()
```

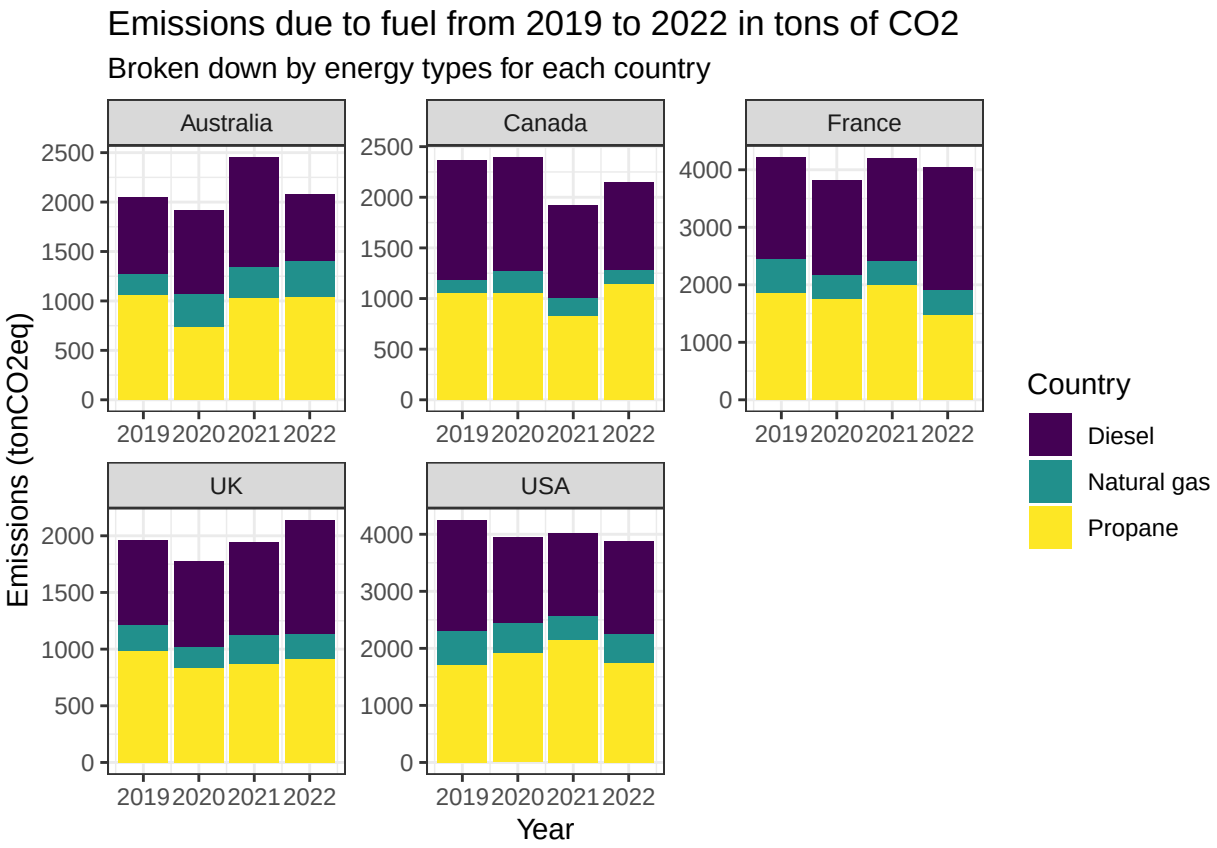
```
## `summarise()` has grouped output by 'country', 'year', 'site', 'energy_type'.
## You can override using the `.groups` argument.
## `summarise()` has grouped output by 'country', 'year', 'site', 'energy_type'.
## You can override using the `.groups` argument.
```

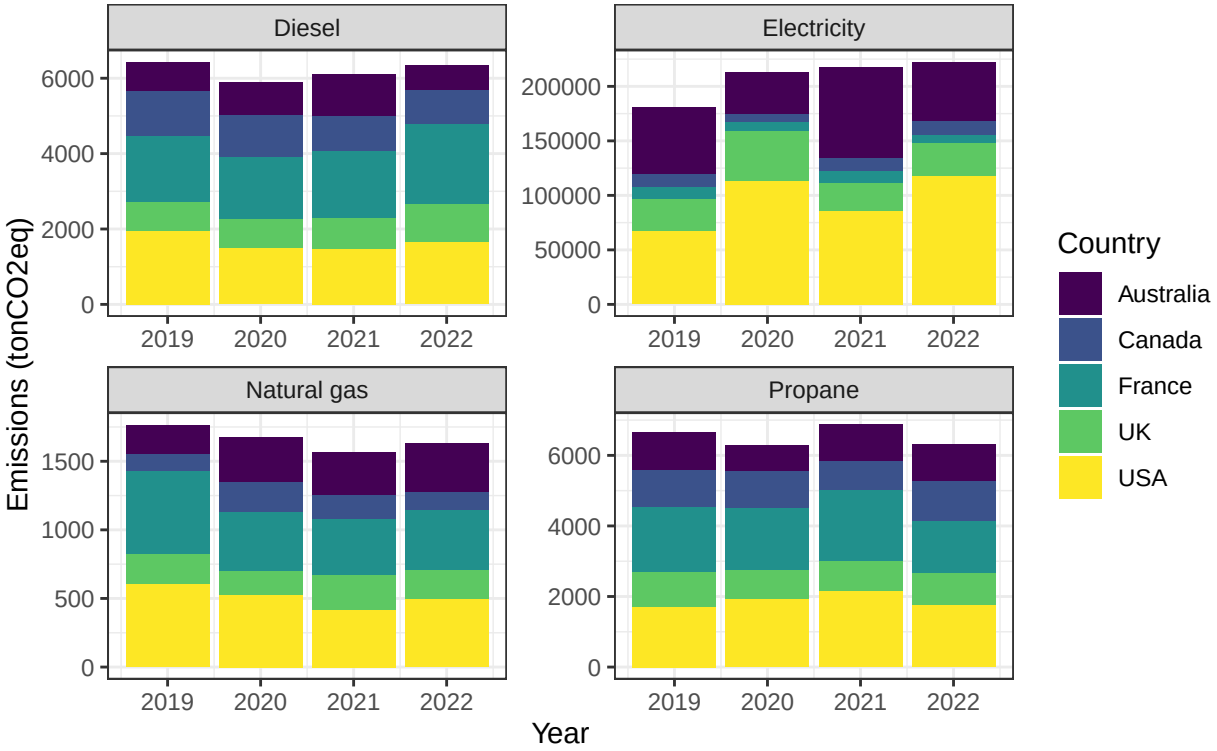
```
## Warning in left_join(., EF_fuel, by = c("energy_type", "unit")): Detected an unexpected many-to-many
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 1 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## `summarise()` has grouped output by 'country', 'year', 'site'. You can override
## using the `.groups` argument.
```

Data visualization




Emissions due to fuel from 2019 to 2022 in tons of CO2
Broken down by energy types for each country
Australia
Canada
France
UK
USA
Emissions (tonCO2eq)
Year
Country
Diesel
Natural gas
Propane


Emissions from 2019 to 2022 in tons of CO2
Broken down by energy types for each country
Diesel
Electricity
Natural gas
Propane
Emissions (tonCO2eq)
Year
Country
Australia
Canada
France
UK
USA