E\_pox\_Rpractical\_basic.r

xandrianou

2023-10-04

# R Practical  
# Material prepared by Xanthi Andrianou and Gianfranco Spiteri  
# October 2023  
  
############ "E-pox: Importing, cleaning and describing data"  
  
# Basic script to install packages  
  
# List of packages to install  
packages\_to\_install <- c("tidyverse", "janitor", "lubridate",  
 "jsonlite", "readxl",  
 "skimr", "gtsummary", "knitr", "apyramid")  
  
# For each package in the list, install if not already installed  
for (package in packages\_to\_install) {  
 if (!requireNamespace(package, quietly = TRUE)) {  
 install.packages(package, dependencies = TRUE)  
 }  
}  
  
# Remove the list  
rm(packages\_to\_install)  
  
  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.3 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lubridate)  
library(janitor)

##   
## Attaching package: 'janitor'  
##   
## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

library(glue)  
library(knitr)  
  
  
# 1. Importing  
  
## Case-based data  
  
CB\_Data0<-read.csv("input\_data/E\_pox\_case\_based\_data.csv")  
  
skimr::skim(CB\_Data0)

Data summary

|  |  |
| --- | --- |
| Name | CB\_Data0 |
| Number of rows | 2000 |
| Number of columns | 13 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 12 |
| numeric | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DateOfNotification | 0 | 1.00 | 10 | 10 | 0 | 179 | 0 |
| DateOfDiagnosis | 715 | 0.64 | 10 | 10 | 0 | 152 | 0 |
| DateOfOnset | 273 | 0.86 | 10 | 10 | 0 | 174 | 0 |
| Gender | 0 | 1.00 | 1 | 3 | 0 | 4 | 0 |
| SexualOrientation | 900 | 0.55 | 3 | 8 | 0 | 4 | 0 |
| Outcome | 0 | 1.00 | 1 | 3 | 0 | 2 | 0 |
| HIVStatus | 722 | 0.64 | 3 | 3 | 0 | 3 | 0 |
| Country | 0 | 1.00 | 8 | 8 | 0 | 5 | 0 |
| ClinicalSymptoms | 0 | 1.00 | 0 | 32 | 12 | 8 | 0 |
| Rash | 90 | 0.96 | 4 | 4 | 0 | 1 | 0 |
| SystemicSymptoms | 606 | 0.70 | 17 | 17 | 0 | 1 | 0 |
| Lesions | 973 | 0.51 | 7 | 7 | 0 | 1 | 0 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Age | 3 | 1 | 37.99 | 10.43 | 2 | 31 | 37 | 45 | 81 | ▁▆▇▂▁ |

# Importing another file type (see ppt)  
CB\_Data0\_json<-jsonlite::fromJSON("input\_data/E\_pox\_case\_based\_data.json")  
  
CB\_Data0\_xlsx<-readxl::read\_excel(path = "input\_data/E\_pox\_case\_based\_data.xlsx")  
  
  
## Aggregated data  
  
Agg\_Data0<-read.csv("input\_data/E\_pox\_aggregated\_data.csv")  
  
skimr::skim(CB\_Data0)

Data summary

|  |  |
| --- | --- |
| Name | CB\_Data0 |
| Number of rows | 2000 |
| Number of columns | 13 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 12 |
| numeric | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DateOfNotification | 0 | 1.00 | 10 | 10 | 0 | 179 | 0 |
| DateOfDiagnosis | 715 | 0.64 | 10 | 10 | 0 | 152 | 0 |
| DateOfOnset | 273 | 0.86 | 10 | 10 | 0 | 174 | 0 |
| Gender | 0 | 1.00 | 1 | 3 | 0 | 4 | 0 |
| SexualOrientation | 900 | 0.55 | 3 | 8 | 0 | 4 | 0 |
| Outcome | 0 | 1.00 | 1 | 3 | 0 | 2 | 0 |
| HIVStatus | 722 | 0.64 | 3 | 3 | 0 | 3 | 0 |
| Country | 0 | 1.00 | 8 | 8 | 0 | 5 | 0 |
| ClinicalSymptoms | 0 | 1.00 | 0 | 32 | 12 | 8 | 0 |
| Rash | 90 | 0.96 | 4 | 4 | 0 | 1 | 0 |
| SystemicSymptoms | 606 | 0.70 | 17 | 17 | 0 | 1 | 0 |
| Lesions | 973 | 0.51 | 7 | 7 | 0 | 1 | 0 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Age | 3 | 1 | 37.99 | 10.43 | 2 | 31 | 37 | 45 | 81 | ▁▆▇▂▁ |

Agg\_Data0\_json<-jsonlite::fromJSON("input\_data/E\_pox\_aggregated\_data.json")  
  
  
Agg\_Data0\_xlsx<-readxl::read\_excel(path = "input\_data/E\_pox\_aggregated\_data.xlsx")  
  
# Remove the files we do not need   
# within rm() we ask for the objects containing   
# the pattern "json" or "xlsx" to be removed from the environment  
rm(list = ls(pattern = "json|xlsx"))  
  
# 2. Cleaning  
  
## Case-based data  
  
# Start working using the CB\_Data0 dataset  
  
str(CB\_Data0)

## 'data.frame': 2000 obs. of 13 variables:  
## $ DateOfNotification: chr "2022-03-03" "2022-03-23" "2022-05-01" "2022-03-10" ...  
## $ DateOfDiagnosis : chr "2022-03-02" "2022-03-23" "2022-05-01" NA ...  
## $ DateOfOnset : chr "2022-02-27" "2022-03-13" "2022-04-19" "2022-02-27" ...  
## $ Age : int 39 46 49 45 41 37 40 36 27 36 ...  
## $ Gender : chr "M" "M" "F" "M" ...  
## $ SexualOrientation : chr "MSM" NA NA NA ...  
## $ Outcome : chr "A" "A" "UNK" "A" ...  
## $ HIVStatus : chr "POS" "UNK" NA "NEG" ...  
## $ Country : chr "CountryA" "CountryA" "CountryC" "CountryC" ...  
## $ ClinicalSymptoms : chr "Rash" "Rash" "Rash, Systemic symptoms" "Rash, Systemic symptoms, Lesions" ...  
## $ Rash : chr "Rash" "Rash" "Rash" "Rash" ...  
## $ SystemicSymptoms : chr NA NA "Systemic symptoms" "Systemic symptoms" ...  
## $ Lesions : chr NA NA NA "Lesions" ...

summary(CB\_Data0)

## DateOfNotification DateOfDiagnosis DateOfOnset Age   
## Length:2000 Length:2000 Length:2000 Min. : 2.00   
## Class :character Class :character Class :character 1st Qu.:31.00   
## Mode :character Mode :character Mode :character Median :37.00   
## Mean :37.99   
## 3rd Qu.:45.00   
## Max. :81.00   
## NA's :3   
## Gender SexualOrientation Outcome HIVStatus   
## Length:2000 Length:2000 Length:2000 Length:2000   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Country ClinicalSymptoms Rash SystemicSymptoms   
## Length:2000 Length:2000 Length:2000 Length:2000   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Lesions   
## Length:2000   
## Class :character   
## Mode :character   
##   
##   
##   
##

head(CB\_Data0)

## DateOfNotification DateOfDiagnosis DateOfOnset Age Gender SexualOrientation  
## 1 2022-03-03 2022-03-02 2022-02-27 39 M MSM  
## 2 2022-03-23 2022-03-23 2022-03-13 46 M <NA>  
## 3 2022-05-01 2022-05-01 2022-04-19 49 F <NA>  
## 4 2022-03-10 <NA> 2022-02-27 45 M <NA>  
## 5 2022-05-04 2022-05-04 2022-04-27 41 M <NA>  
## 6 2022-03-09 <NA> 2022-03-06 37 M <NA>  
## Outcome HIVStatus Country ClinicalSymptoms Rash  
## 1 A POS CountryA Rash Rash  
## 2 A UNK CountryA Rash Rash  
## 3 UNK <NA> CountryC Rash, Systemic symptoms Rash  
## 4 A NEG CountryC Rash, Systemic symptoms, Lesions Rash  
## 5 A NEG CountryA Rash, Systemic symptoms, Lesions Rash  
## 6 A POS CountryC Rash, Systemic symptoms, Lesions Rash  
## SystemicSymptoms Lesions  
## 1 <NA> <NA>  
## 2 <NA> <NA>  
## 3 Systemic symptoms <NA>  
## 4 Systemic symptoms Lesions  
## 5 Systemic symptoms Lesions  
## 6 Systemic symptoms Lesions

glimpse(CB\_Data0)

## Rows: 2,000  
## Columns: 13  
## $ DateOfNotification <chr> "2022-03-03", "2022-03-23", "2022-05-01", "2022-03-…  
## $ DateOfDiagnosis <chr> "2022-03-02", "2022-03-23", "2022-05-01", NA, "2022…  
## $ DateOfOnset <chr> "2022-02-27", "2022-03-13", "2022-04-19", "2022-02-…  
## $ Age <int> 39, 46, 49, 45, 41, 37, 40, 36, 27, 36, 34, 38, 68,…  
## $ Gender <chr> "M", "M", "F", "M", "M", "M", "F", "M", "M", "M", "…  
## $ SexualOrientation <chr> "MSM", NA, NA, NA, NA, NA, "HETERO", "MSM", NA, NA,…  
## $ Outcome <chr> "A", "A", "UNK", "A", "A", "A", "A", "A", "A", "UNK…  
## $ HIVStatus <chr> "POS", "UNK", NA, "NEG", "NEG", "POS", "NEG", "POS"…  
## $ Country <chr> "CountryA", "CountryA", "CountryC", "CountryC", "Co…  
## $ ClinicalSymptoms <chr> "Rash", "Rash", "Rash, Systemic symptoms", "Rash, S…  
## $ Rash <chr> "Rash", "Rash", "Rash", "Rash", "Rash", "Rash", "Ra…  
## $ SystemicSymptoms <chr> NA, NA, "Systemic symptoms", "Systemic symptoms", "…  
## $ Lesions <chr> NA, NA, NA, "Lesions", "Lesions", "Lesions", NA, NA…

# Working on the date variables  
  
CB\_Data <- CB\_Data0 %>%   
 # convert one date variable  
 mutate(DateOfNotification=ymd(DateOfNotification)) %>%   
 # Convert the date variables to date  
 mutate(across(starts\_with("Date"),  
 ~ ymd(.x))) %>%   
 # Add a variable with the week  
 mutate(WeekDate=floor\_date(DateOfNotification, unit="week", week\_start="Monday")) %>%   
 mutate(  
 Gender = case\_when(  
 Gender == "F" ~ "Female",  
 Gender == "M" ~ "Male",  
 Gender == "O" ~ "Other",  
 Gender == "UNK" ~ "Unknown",  
 .default = Gender  
 )  
 )  
  
  
head(CB\_Data)

## DateOfNotification DateOfDiagnosis DateOfOnset Age Gender SexualOrientation  
## 1 2022-03-03 2022-03-02 2022-02-27 39 Male MSM  
## 2 2022-03-23 2022-03-23 2022-03-13 46 Male <NA>  
## 3 2022-05-01 2022-05-01 2022-04-19 49 Female <NA>  
## 4 2022-03-10 <NA> 2022-02-27 45 Male <NA>  
## 5 2022-05-04 2022-05-04 2022-04-27 41 Male <NA>  
## 6 2022-03-09 <NA> 2022-03-06 37 Male <NA>  
## Outcome HIVStatus Country ClinicalSymptoms Rash  
## 1 A POS CountryA Rash Rash  
## 2 A UNK CountryA Rash Rash  
## 3 UNK <NA> CountryC Rash, Systemic symptoms Rash  
## 4 A NEG CountryC Rash, Systemic symptoms, Lesions Rash  
## 5 A NEG CountryA Rash, Systemic symptoms, Lesions Rash  
## 6 A POS CountryC Rash, Systemic symptoms, Lesions Rash  
## SystemicSymptoms Lesions WeekDate  
## 1 <NA> <NA> 2022-02-28  
## 2 <NA> <NA> 2022-03-21  
## 3 Systemic symptoms <NA> 2022-04-25  
## 4 Systemic symptoms Lesions 2022-03-07  
## 5 Systemic symptoms Lesions 2022-05-02  
## 6 Systemic symptoms Lesions 2022-03-07

skimr::skim(CB\_Data)

Data summary

|  |  |
| --- | --- |
| Name | CB\_Data |
| Number of rows | 2000 |
| Number of columns | 14 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 9 |
| Date | 4 |
| numeric | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Gender | 0 | 1.00 | 4 | 7 | 0 | 4 | 0 |
| SexualOrientation | 900 | 0.55 | 3 | 8 | 0 | 4 | 0 |
| Outcome | 0 | 1.00 | 1 | 3 | 0 | 2 | 0 |
| HIVStatus | 722 | 0.64 | 3 | 3 | 0 | 3 | 0 |
| Country | 0 | 1.00 | 8 | 8 | 0 | 5 | 0 |
| ClinicalSymptoms | 0 | 1.00 | 0 | 32 | 12 | 8 | 0 |
| Rash | 90 | 0.96 | 4 | 4 | 0 | 1 | 0 |
| SystemicSymptoms | 606 | 0.70 | 17 | 17 | 0 | 1 | 0 |
| Lesions | 973 | 0.51 | 7 | 7 | 0 | 1 | 0 |

**Variable type: Date**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| DateOfNotification | 0 | 1.00 | 2022-01-19 | 2022-08-24 | 2022-04-06 | 179 |
| DateOfDiagnosis | 715 | 0.64 | 2022-01-19 | 2022-08-17 | 2022-04-05 | 152 |
| DateOfOnset | 273 | 0.86 | 2022-01-12 | 2022-08-10 | 2022-03-28 | 174 |
| WeekDate | 0 | 1.00 | 2022-01-17 | 2022-08-22 | 2022-04-04 | 32 |

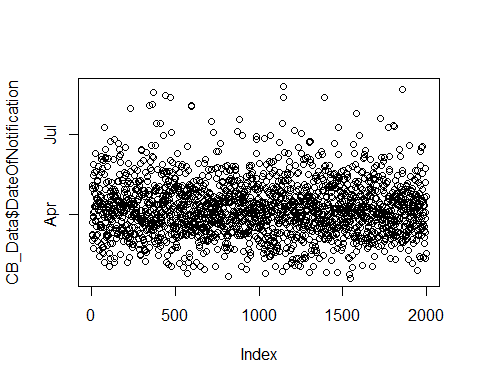
**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Age | 3 | 1 | 37.99 | 10.43 | 2 | 31 | 37 | 45 | 81 | ▁▆▇▂▁ |

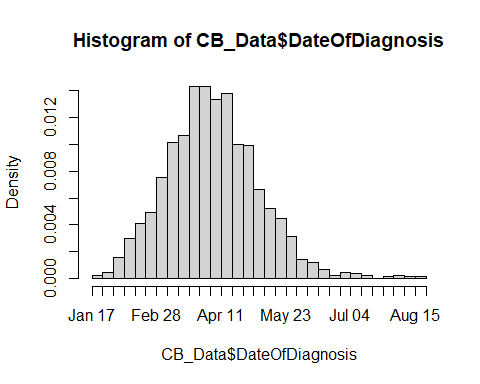
glimpse(CB\_Data)

## Rows: 2,000  
## Columns: 14  
## $ DateOfNotification <date> 2022-03-03, 2022-03-23, 2022-05-01, 2022-03-10, 20…  
## $ DateOfDiagnosis <date> 2022-03-02, 2022-03-23, 2022-05-01, NA, 2022-05-04…  
## $ DateOfOnset <date> 2022-02-27, 2022-03-13, 2022-04-19, 2022-02-27, 20…  
## $ Age <int> 39, 46, 49, 45, 41, 37, 40, 36, 27, 36, 34, 38, 68,…  
## $ Gender <chr> "Male", "Male", "Female", "Male", "Male", "Male", "…  
## $ SexualOrientation <chr> "MSM", NA, NA, NA, NA, NA, "HETERO", "MSM", NA, NA,…  
## $ Outcome <chr> "A", "A", "UNK", "A", "A", "A", "A", "A", "A", "UNK…  
## $ HIVStatus <chr> "POS", "UNK", NA, "NEG", "NEG", "POS", "NEG", "POS"…  
## $ Country <chr> "CountryA", "CountryA", "CountryC", "CountryC", "Co…  
## $ ClinicalSymptoms <chr> "Rash", "Rash", "Rash, Systemic symptoms", "Rash, S…  
## $ Rash <chr> "Rash", "Rash", "Rash", "Rash", "Rash", "Rash", "Ra…  
## $ SystemicSymptoms <chr> NA, NA, "Systemic symptoms", "Systemic symptoms", "…  
## $ Lesions <chr> NA, NA, NA, "Lesions", "Lesions", "Lesions", NA, NA…  
## $ WeekDate <date> 2022-02-28, 2022-03-21, 2022-04-25, 2022-03-07, 20…

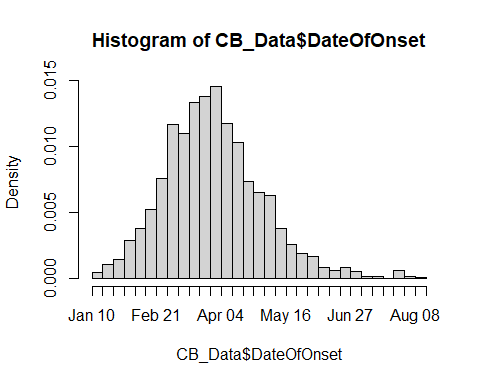
plot(CB\_Data$DateOfNotification)



hist(CB\_Data$DateOfDiagnosis, breaks="weeks")



hist(CB\_Data$DateOfOnset, breaks="weeks")



skimr::skim(CB\_Data)

Data summary

|  |  |
| --- | --- |
| Name | CB\_Data |
| Number of rows | 2000 |
| Number of columns | 14 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 9 |
| Date | 4 |
| numeric | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Gender | 0 | 1.00 | 4 | 7 | 0 | 4 | 0 |
| SexualOrientation | 900 | 0.55 | 3 | 8 | 0 | 4 | 0 |
| Outcome | 0 | 1.00 | 1 | 3 | 0 | 2 | 0 |
| HIVStatus | 722 | 0.64 | 3 | 3 | 0 | 3 | 0 |
| Country | 0 | 1.00 | 8 | 8 | 0 | 5 | 0 |
| ClinicalSymptoms | 0 | 1.00 | 0 | 32 | 12 | 8 | 0 |
| Rash | 90 | 0.96 | 4 | 4 | 0 | 1 | 0 |
| SystemicSymptoms | 606 | 0.70 | 17 | 17 | 0 | 1 | 0 |
| Lesions | 973 | 0.51 | 7 | 7 | 0 | 1 | 0 |

**Variable type: Date**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| DateOfNotification | 0 | 1.00 | 2022-01-19 | 2022-08-24 | 2022-04-06 | 179 |
| DateOfDiagnosis | 715 | 0.64 | 2022-01-19 | 2022-08-17 | 2022-04-05 | 152 |
| DateOfOnset | 273 | 0.86 | 2022-01-12 | 2022-08-10 | 2022-03-28 | 174 |
| WeekDate | 0 | 1.00 | 2022-01-17 | 2022-08-22 | 2022-04-04 | 32 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Age | 3 | 1 | 37.99 | 10.43 | 2 | 31 | 37 | 45 | 81 | ▁▆▇▂▁ |

# Cross-tabulate the country info and the date of notification  
janitor::tabyl(dat = CB\_Data,   
 WeekDate,   
 Country)

## WeekDate CountryA CountryB CountryC CountryD CountryE  
## 2022-01-17 2 1 0 0 0  
## 2022-01-24 4 1 2 1 1  
## 2022-01-31 7 10 6 2 2  
## 2022-02-07 9 10 8 3 1  
## 2022-02-14 23 3 12 4 0  
## 2022-02-21 25 16 11 7 6  
## 2022-02-28 41 16 21 15 12  
## 2022-03-07 61 22 40 14 5  
## 2022-03-14 51 30 37 16 8  
## 2022-03-21 72 39 37 21 11  
## 2022-03-28 83 46 35 23 4  
## 2022-04-04 77 38 47 22 12  
## 2022-04-11 86 35 46 23 8  
## 2022-04-18 63 27 36 12 7  
## 2022-04-25 44 25 33 14 10  
## 2022-05-02 44 17 22 11 6  
## 2022-05-09 39 10 21 6 1  
## 2022-05-16 30 11 19 5 2  
## 2022-05-23 21 8 9 5 0  
## 2022-05-30 12 3 9 3 2  
## 2022-06-06 7 9 5 2 0  
## 2022-06-13 2 5 6 1 0  
## 2022-06-20 2 3 3 3 0  
## 2022-06-27 3 3 1 1 0  
## 2022-07-04 0 2 2 1 1  
## 2022-07-11 2 0 3 0 0  
## 2022-07-18 1 0 1 0 1  
## 2022-07-25 0 0 0 1 0  
## 2022-08-01 2 1 0 1 0  
## 2022-08-08 3 0 0 0 1  
## 2022-08-15 0 0 2 0 0  
## 2022-08-22 0 0 0 0 1

# Cross-tabulate the country info and the gender variable  
janitor::tabyl(dat = CB\_Data,   
 Gender,   
 Country)

## Gender CountryA CountryB CountryC CountryD CountryE  
## Female 15 8 9 4 0  
## Male 800 382 463 213 102  
## Other 0 0 1 0 0  
## Unknown 1 1 1 0 0

## Aggregated data  
  
# Working on the Agg\_Data0 dataset  
  
str(Agg\_Data0)

## 'data.frame': 101 obs. of 3 variables:  
## $ DateRep: chr "2022-01-17" "2022-01-24" "2022-01-31" "2022-02-07" ...  
## $ Country: chr "CountryA" "CountryA" "CountryA" "CountryA" ...  
## $ Cases : int 2 4 11 18 40 64 105 164 213 284 ...

names(Agg\_Data0)

## [1] "DateRep" "Country" "Cases"

summary(Agg\_Data0)

## DateRep Country Cases   
## Length:101 Length:101 Min. : 1.0   
## Class :character Class :character 1st Qu.: 53.0   
## Mode :character Mode :character Median :163.0   
## Mean :242.4   
## 3rd Qu.:370.0   
## Max. :786.0

Agg\_Data <- Agg\_Data0 %>%   
 # Convert the date variables to date  
 mutate(across(.cols= starts\_with("Date"),  
 .fns = ~ ymd(.x))) %>%   
 # Convert character columns to factors:  
 mutate(across(.cols = where(is.character),   
 .fns = ~ as.factor(.x))) %>%   
 # Add a variable with the week  
 mutate(WeekDate=floor\_date(DateRep, unit="week", week\_start="Monday"))   
   
str(Agg\_Data)

## 'data.frame': 101 obs. of 4 variables:  
## $ DateRep : Date, format: "2022-01-17" "2022-01-24" ...  
## $ Country : Factor w/ 4 levels "CountryA","CountryB",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Cases : int 2 4 11 18 40 64 105 164 213 284 ...  
## $ WeekDate: Date, format: "2022-01-17" "2022-01-24" ...

summary(Agg\_Data)

## DateRep Country Cases WeekDate   
## Min. :2022-01-17 CountryA:28 Min. : 1.0 Min. :2022-01-17   
## 1st Qu.:2022-03-07 CountryB:26 1st Qu.: 53.0 1st Qu.:2022-03-07   
## Median :2022-04-18 CountryD:26 Median :163.0 Median :2022-04-18   
## Mean :2022-04-20 CountryE:21 Mean :242.4 Mean :2022-04-20   
## 3rd Qu.:2022-05-30 3rd Qu.:370.0 3rd Qu.:2022-05-30   
## Max. :2022-08-22 Max. :786.0 Max. :2022-08-22

# 3. Basic descriptives of the case-based data  
  
  
# nrow() used below gives the total number of rows in the dataset  
# Getting the total number of cases   
  
nrow(CB\_Data)

## [1] 2000

## Summary of the characteristics of the cases (tables and graphs)  
  
### Summary by country  
  
CB\_table <- CB\_Data %>%  
 # Select columns starting with "Date"  
 select(Country) %>%  
 # Create summary table (attention to handling of missing values)  
 gtsummary::tbl\_summary(missing = "ifany")   
CB\_table

## Table printed with {flextable}, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| **Characteristic** | **N = 2,000**1 |
| --- | --- |
| Country |  |
| CountryA | 816 (41%) |
| CountryB | 391 (20%) |
| CountryC | 474 (24%) |
| CountryD | 217 (11%) |
| CountryE | 102 (5.1%) |
| 1n (%) | |

### Time  
  
# Checking the different variables  
  
CB\_table1 <- CB\_Data %>%  
 # Select columns starting with "Date"  
 select(starts\_with("Date")) %>%  
 # Create summary table (attention to handling of missing values)  
 gtsummary::tbl\_summary(missing = "ifany")   
  
CB\_table1

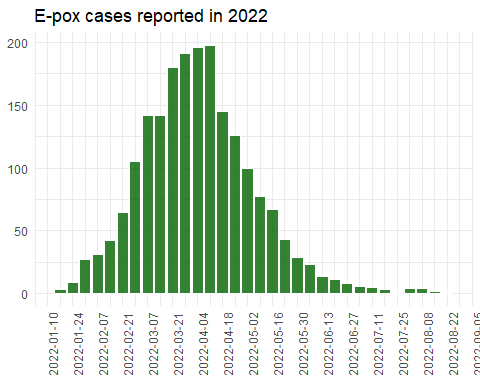
## Table printed with {flextable}, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| **Characteristic** | **N = 2,000**1 |
| --- | --- |
| DateOfNotification | 2022-01-19 to 2022-08-24 |
| DateOfDiagnosis | 2022-01-19 to 2022-08-17 |
| Unknown | 715 |
| DateOfOnset | 2022-01-12 to 2022-08-10 |
| Unknown | 273 |
| 1Range | |

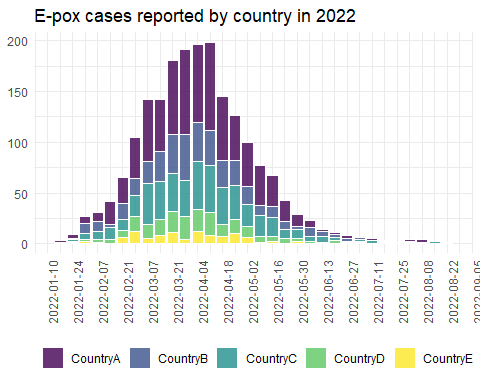
#### Epicurves (overall and by country)  
  
# First we can nake an object to summarise the data we want to  
# show on the epicurve  
EpiCurve\_object<- CB\_Data %>%   
 group\_by(WeekDate) %>%   
 summarise(n=n(), .groups = "drop")   
  
head(EpiCurve\_object)

## # A tibble: 6 × 2  
## WeekDate n  
## <date> <int>  
## 1 2022-01-17 3  
## 2 2022-01-24 9  
## 3 2022-01-31 27  
## 4 2022-02-07 31  
## 5 2022-02-14 42  
## 6 2022-02-21 65

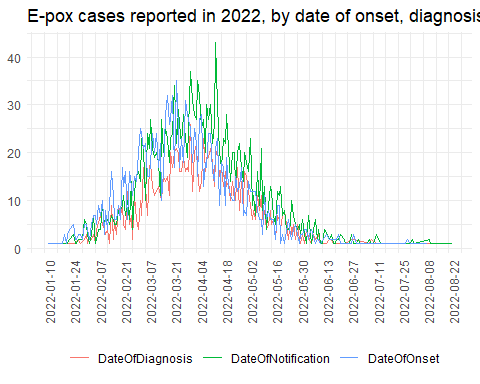
EpiCurve <- EpiCurve\_object %>%   
 ggplot() +  
 geom\_col(aes(x = WeekDate, y = n), fill="darkgreen", color="white", alpha=0.8) +  
 scale\_fill\_viridis\_d() +  
 scale\_x\_date(breaks = "2 weeks")+  
 theme\_minimal() +  
 theme(legend.position = "bottom",   
 axis.text = element\_text(size=9),  
 axis.title = element\_blank(),  
 axis.text.x = element\_text(angle=90),  
 legend.title = element\_blank()) +  
 labs(title="E-pox cases reported in 2022")  
  
EpiCurve



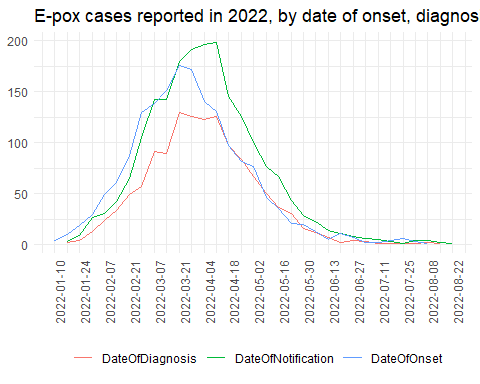
# The dataset to be used for the epicurve (and in general   
# in ggplot()) can be created right before the plot and it   
# can be passed directly to the ggplot()  
  
EpiCurve\_byCountry<-CB\_Data %>%   
 group\_by(Country, WeekDate) %>%   
 summarise(n=n(), .groups = "drop") %>%   
 ungroup() %>%   
 ggplot(.) +  
 geom\_col(aes(x = WeekDate, y = n, fill = Country), color="white", alpha=0.8) +  
 scale\_fill\_viridis\_d() +  
 scale\_x\_date(breaks = "2 weeks")+  
 theme\_minimal() +  
 theme(legend.position = "bottom",   
 axis.text = element\_text(size=9),  
 axis.title = element\_blank(),  
 axis.text.x = element\_text(angle=90),  
 legend.title = element\_blank()) +  
 labs(title="E-pox cases reported by country in 2022")  
  
EpiCurve\_byCountry



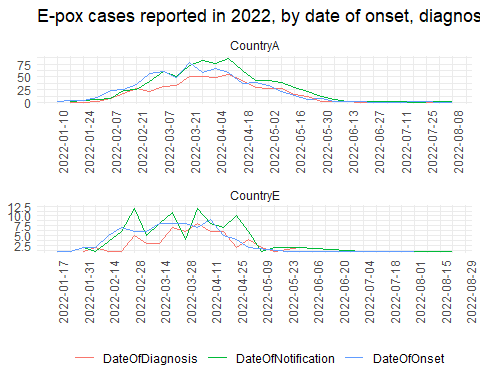
#### Plot the dates of onset, diagnosis, and notification  
  
#In the plots below, we use the dates of onset, diagnosis and notification and put them together to see whether there are difference and visualize the information to check if there are large delays. First, we try with dates and the plot is a messy (not easy to interpret.  
  
plot\_dates\_object <- CB\_Data %>%  
 # use the variables of the dates and make a longer dataset  
 # In the pivot\_longer() command we select the columns which   
 # we want to expand in long format and transform the dataset   
 pivot\_longer(  
 # all columns starting with "DateOf..." will be taken   
 cols=starts\_with("DateOf"),   
 # the names of the variables will be placed in a single column called "Indicator"  
 names\_to = "Indicator",   
 # the values (which are dates in this case) will be placed in a   
 # column called "Date  
 values\_to = "Date") %>%  
 # group by date and indicator (type of date, i.e., onset, diagnosis or notification)   
 group\_by(Indicator, Date) %>%   
 summarise(n=n(), .groups="drop") %>%   
 # we add a filter to remove any rows where a date value might not exist  
 filter(!is.na(Date))  
  
# then we pass the object to the ggplot()  
plot\_dates <- plot\_dates\_object %>%   
 ggplot() +  
 geom\_line(aes(x = Date, y = n, color=Indicator)) +  
 scale\_x\_date(breaks = "2 weeks")+  
 theme\_minimal() +  
 theme(legend.position = "bottom",   
 axis.text = element\_text(size=9),  
 axis.title = element\_blank(),  
 axis.text.x = element\_text(angle=90),  
 legend.title = element\_blank()) +  
 labs(title="E-pox cases reported in 2022, by date of onset, diagnosis and notification.")  
  
plot\_dates



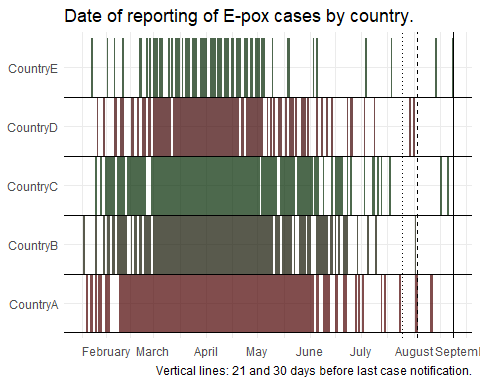
# Then, we try to plot by week and there is some improvement in the visualization.  
  
plot\_date\_w<- CB\_Data %>%  
 # use the variables of the dates and make a longer dataset  
 pivot\_longer(cols=starts\_with("DateOf"), names\_to = "Indicator", values\_to = "Date") %>%  
 mutate(Date=floor\_date(Date, unit="week", week\_start="Monday")) %>%   
 # group by date and indicator (type of date, i.e., onset, diagnosis or notification)   
 group\_by(Indicator, Date) %>%   
 summarise(n=n(), .groups="drop") %>%   
 filter(!is.na(Date)) %>%   
 ggplot() +  
 geom\_line(aes(x = Date, y = n, color=Indicator)) +  
 scale\_x\_date(breaks = "2 weeks")+  
 theme\_minimal() +  
 theme(legend.position = "bottom",   
 axis.text = element\_text(size=9),  
 axis.title = element\_blank(),  
 axis.text.x = element\_text(angle=90),  
 legend.title = element\_blank()) +  
 labs(title="E-pox cases reported in 2022, by date of onset, diagnosis and notification.")  
  
plot\_date\_w



# We can create the same graph by country.  
  
  
plot\_date\_w\_country<- CB\_Data %>%  
 # selecting only two countries  
 filter(Country=="CountryA"|Country=="CountryE") %>%   
 # use the variables of the dates and make a longer dataset  
 pivot\_longer(cols=starts\_with("DateOf"), names\_to = "Indicator", values\_to = "Date") %>%  
 mutate(Date=floor\_date(Date, unit="week", week\_start="Monday")) %>%   
 # group by date and indicator (type of date, i.e., onset, diagnosis or notification)   
 group\_by(Country, Indicator, Date) %>%   
 summarise(n=n(), .groups="drop") %>%   
 filter(!is.na(Date)) %>%   
 ggplot() +  
 geom\_line(aes(x = Date, y = n, color=Indicator)) +  
 # in this line we indicate that a graph should be created   
 # by country and that the axis should have different scales which  
 # are adjusted by the range of values in each country  
 facet\_wrap(~Country, ncol = 1, scales = "free")+  
 scale\_x\_date(breaks = "2 weeks")+  
 theme\_minimal() +  
 theme(legend.position = "bottom",   
 axis.text = element\_text(size=9),  
 axis.title = element\_blank(),  
 axis.text.x = element\_text(angle=90),  
 legend.title = element\_blank()) +  
 labs(title="E-pox cases reported in 2022, by date of onset, diagnosis and notification.")  
  
plot\_date\_w\_country



#### Graph on cases reported by day and country  
  
hm\_CB\_Data<-CB\_Data %>%   
 distinct(DateOfNotification, Country) %>%   
 mutate(n=1) %>%   
 ungroup() %>%   
 ggplot(.) +  
 geom\_tile(aes(x=DateOfNotification, y=as.factor(Country), fill = as.factor(Country)), alpha=0.7) +  
 geom\_hline(yintercept = seq(.5, nlevels(as.factor(CB\_Data$Country)), 1), linewidth = .2) +  
 geom\_vline(xintercept = max(CB\_Data$DateOfNotification, na.rm=T)-as.difftime(tim = 21, units = "days"), linetype=2) +  
 geom\_vline(xintercept = max(CB\_Data$DateOfNotification, na.rm=T)-as.difftime(tim = 30, units = "days"), linetype=3) +  
 geom\_vline(xintercept = max(CB\_Data$DateOfNotification, na.rm=T), linetype=1) +  
 scale\_fill\_hue(l=4) +  
 scale\_x\_date(date\_breaks = "1 month", date\_labels = "%B") +  
 theme\_minimal() +  
 theme(legend.position = "none",   
 axis.text = element\_text(size=9),  
 axis.title = element\_blank()) +  
 labs(caption="Vertical lines: 21 and 30 days before last case notification.",  
 title="Date of reporting of E-pox cases by country.")  
  
hm\_CB\_Data



### Demographics  
  
CB\_Data <- CB\_Data %>%   
 mutate(Age\_cat=cut(Age,   
 breaks = c(0, 14, 17, 30, 40, 50, 60, 999),   
 include.lowest = T))  
  
CB\_demo <- CB\_Data %>%  
 # Select demographics and other columns  
 select("Age\_cat") %>%  
 # Create summary table (attention to handling of missing values)  
 gtsummary::tbl\_summary(missing = "ifany")  
  
CB\_demo

## Table printed with {flextable}, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

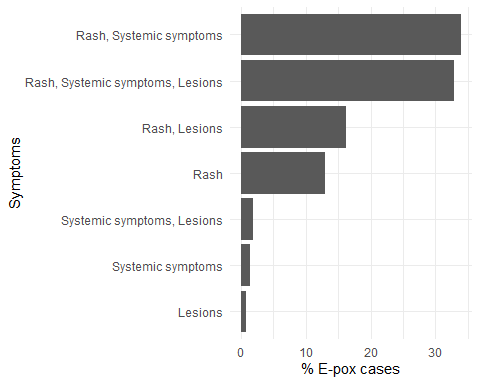
| **Characteristic** | **N = 2,000**1 |
| --- | --- |
| Age\_cat |  |
| [0,14] | 3 (0.2%) |
| (14,17] | 13 (0.7%) |
| (17,30] | 479 (24%) |
| (30,40] | 779 (39%) |
| (40,50] | 474 (24%) |
| (50,60] | 192 (9.6%) |
| (60,999] | 57 (2.9%) |
| Unknown | 3 |
| 1n (%) | |

### Clinical description  
  
clin\_desc\_table <- CB\_Data %>%  
 # Select demographics and other columns  
 select("ClinicalSymptoms") %>%  
 ## Create summary table (attention to handling of missing values)  
 gtsummary::tbl\_summary(missing = "ifany")   
  
clin\_desc\_table

## Table printed with {flextable}, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| **Characteristic** | **N = 2,000**1 |
| --- | --- |
| ClinicalSymptoms |  |
|  | 12 (0.6%) |
| Lesions | 14 (0.7%) |
| Rash | 257 (13%) |
| Rash, Lesions | 323 (16%) |
| Rash, Systemic symptoms | 676 (34%) |
| Rash, Systemic symptoms, Lesions | 654 (33%) |
| Systemic symptoms | 28 (1.4%) |
| Systemic symptoms, Lesions | 36 (1.8%) |
| 1n (%) | |

clinical\_desc <- CB\_Data %>%   
 filter(ClinicalSymptoms!="") %>%   
 # Group by symptoms  
 group\_by(ClinicalSymptoms) %>%   
 # Count for each symptom:  
 summarise(count = n(), .groups="drop") %>%   
 mutate(prop=(count/sum(count))\*100) %>%   
 # Create plot:  
 ggplot(aes(  
 # ordering the symptoms by count  
 x = reorder(ClinicalSymptoms, desc(prop), decreasing = TRUE),   
 y = prop)) +  
 # Display bars as proportions  
 geom\_bar(stat = "identity") +  
 #x axis label  
 xlab("Symptoms") +  
 #y axis label  
 ylab("% E-pox cases") +  
 # flip the axis  
 coord\_flip() +  
 theme\_minimal()  
  
# Print plot:  
clinical\_desc



### Outcome, HIV status  
Outcome\_HIV <- CB\_Data %>%  
 # Select columns  
 select("Outcome", "HIVStatus") %>%  
 mutate(Outcome=case\_when(Outcome=="UNK"~"Unknown",  
 Outcome=="A"~"Alive",  
 .default = Outcome)) %>%  
 mutate(HIVStatus=case\_when(HIVStatus=="POS"~"Positive",  
 HIVStatus=="NEG"~"Negative",  
 .default = HIVStatus)) %>%   
 # Create summary table (attention to handling of missing values)  
 gtsummary::tbl\_summary(by = Outcome) %>%  
 gtsummary::add\_overall()   
  
Outcome\_HIV

## Table printed with {flextable}, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

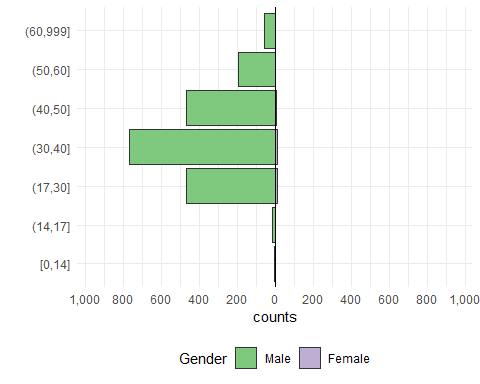
| **Characteristic** | **Overall**, N = 2,0001 | **Alive**, N = 1,4051 | **Unknown**, N = 5951 |
| --- | --- | --- | --- |
| HIVStatus |  |  |  |
| Negative | 525 (41%) | 513 (41%) | 12 (60%) |
| Positive | 307 (24%) | 304 (24%) | 3 (15%) |
| UNK | 446 (35%) | 441 (35%) | 5 (25%) |
| Unknown | 722 | 147 | 575 |
| 1n (%) | | | |

### Sexual orientation  
  
sexorient <- CB\_Data %>%  
 filter(!is.na(SexualOrientation)) %>%   
 # Select columns  
 select("SexualOrientation") %>%  
 gtsummary::tbl\_summary()   
  
sexorient <- CB\_Data %>%  
 filter(!is.na(SexualOrientation)) %>%   
 # Select columns  
 select("SexualOrientation") %>%  
 mutate(SexualOrientation=case\_when(SexualOrientation=="BISEXUAL"~"Bisexual",  
 SexualOrientation=="HETERO"~"Heterosexual",  
 SexualOrientation=="MSM"~"MSM/homo or bisexual male",  
 SexualOrientation=="UNK"~"Unknown or undetermined",  
 .default=SexualOrientation)) %>%   
 gtsummary::tbl\_summary()   
  
  
sexorient

## Table printed with {flextable}, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| **Characteristic** | **N = 1,100**1 |
| --- | --- |
| SexualOrientation |  |
| Bisexual | 7 (0.6%) |
| Heterosexual | 46 (4.2%) |
| MSM/homo or bisexual male | 833 (76%) |
| Unknown or undetermined | 214 (19%) |
| 1n (%) | |

# Optional - Age and gender pyramid  
  
  
# Creating the age and gender pyramid (first filtering out non M or F rows)  
agegender\_pyramid <- CB\_Data %>%  
 filter(Gender=="Female"|Gender=="Male" & !is.na(Age\_cat)) %>%   
 # Create age gender pyramid  
 apyramid::age\_pyramid(  
 # Specify column containing age groups  
 age\_group = "Age\_cat",  
 # Splitting by Gender  
 split\_by = "Gender",   
 # Don't show midpoint on the graph:  
 show\_midpoint = FALSE) +  
 theme\_minimal() +  
 theme(legend.position = "bottom") +  
 labs(x="")  
  
  
agegender\_pyramid



###### (Optional) Exploring trends and progress ###############  
  
# In this part of the analysis we go beyond describing the basic information bue we are also checking how the trends of the reported cases might have changed.   
# Therefore, we work with data reported before the last 3 weeks in which the data collected might be incomplete.  
  
# Create the week labels using dates to be used when we want to automate printing the text  
week\_xx <- paste0(format(as.Date(  
 max(CB\_Data$WeekDate) - as.difftime(tim = 3, units = "weeks")  
), format = "%d/%m"),  
"-",  
format(as.Date(  
 max(CB\_Data$WeekDate) - as.difftime(tim = 15, units = "days")  
),  
format = "%d/%m"))  
  
week\_xx

## [1] "01/08-07/08"

week\_yy <-  
 paste0(format(as.Date(  
 max(CB\_Data$WeekDate) - as.difftime(tim = 4, units = "weeks")  
 ), format = "%d/%m"),  
 "-",  
 format(as.Date(  
 max(CB\_Data$WeekDate) - as.difftime(tim = 22, units = "days")  
 ), format = "%d/%m"))  
  
week\_yy

## [1] "25/07-31/07"

# Summarise the data by week  
WeekCB\_Data <- CB\_Data %>%  
 group\_by(WeekDate) %>%  
 summarise(n = n(), .groups="drop")  
  
# filter for the weeks of interest (there might different ways to do it)  
ltw <- WeekCB\_Data %>%  
 filter(  
 WeekDate < max(CB\_Data$WeekDate) - as.difftime(tim = 2, units = "weeks") &  
 WeekDate >= max(CB\_Data$WeekDate) - as.difftime(tim = 4, units = "weeks")  
 ) %>%  
 mutate(w\_n = case\_when(  
 WeekDate == as.Date(max(CB\_Data$WeekDate) - as.difftime(tim = 3, units = "weeks")) ~  
 "current",  
 WeekDate == as.Date(max(CB\_Data$WeekDate) - as.difftime(tim = 4, units = "weeks")) ~  
 "previous"  
 ))  
  
ltw

## # A tibble: 2 × 3  
## WeekDate n w\_n   
## <date> <int> <chr>   
## 1 2022-07-25 1 previous  
## 2 2022-08-01 4 current

# % change in the weeks of interest  
round(((filter(ltw, w\_n=="current")$n-filter(ltw, w\_n=="previous")$n)/filter(ltw, w\_n=="previous")$n)\*100, digits=1)

## [1] 300

# Number of cases in the "current" week   
filter(ltw, w\_n=="current")$n

## [1] 4

# Number of cases in the "previous" week  
filter(ltw, w\_n=="previous")$n

## [1] 1

# Getting the week with max number of cases  
week\_max <-  
 paste0(format(as.Date(filter(  
 WeekCB\_Data, n == max(WeekCB\_Data$n)  
 )$WeekDate), format = "%d/%m"),  
 "-",  
 format(as.Date(  
 filter(WeekCB\_Data, n == max(WeekCB\_Data$n))$WeekDate + as.difftime(tim = 6, units = "days")  
 ), format = "%d/%m"))  
  
week\_max

## [1] "11/04-17/04"

# The highest number of cases since the beginning of the outbreak was reported on   
week\_max

## [1] "11/04-17/04"

# number of cases:   
filter(WeekCB\_Data, n==max(WeekCB\_Data$n))$n

## [1] 198

# % Change in the number of cases since the week with the max number of cases:  
round(((filter(ltw, w\_n=="current")$n-max(WeekCB\_Data$n))/max(WeekCB\_Data$n))\*100, digits=1)

## [1] -98

# difference:   
filter(ltw, w\_n=="current")$n-max(WeekCB\_Data$n)

## [1] -194

# Days since last case  
days\_since\_last\_case <- CB\_Data %>%  
 group\_by(Country) %>%  
 summarise(days\_last\_report = as.integer(difftime(  
 time1 = max(CB\_Data$DateOfNotification, na.rm = T),  
 time2 = max(DateOfNotification),  
 units = "days"  
 )), .groups="drop") %>%  
 ungroup() %>%  
 mutate(  
 category = case\_when(  
 days\_last\_report <= 21 ~ "21 or less",  
 days\_last\_report <= 30 ~ "22-30 days",  
 days\_last\_report > 30 ~ "More than 30 days"  
 )  
 )  
  
summary\_days\_since\_last <- days\_since\_last\_case %>%  
 group\_by(category) %>%  
 summarise(n\_countries = n(), .groups="drop") %>%  
 adorn\_totals()  
  
  
kable(summary\_days\_since\_last, caption=paste0("Number of countries and days since last case and until ", max(CB\_Data$DateOfNotification)))

Number of countries and days since last case and until 2022-08-24

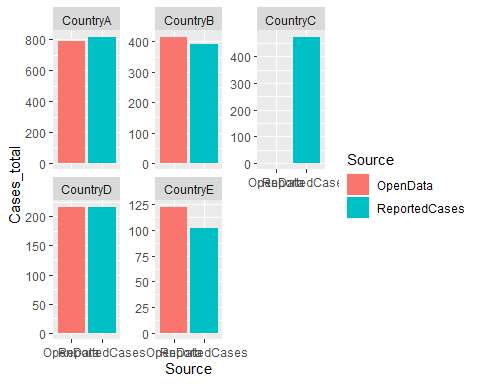
| category | n\_countries |
| --- | --- |
| 21 or less | 3 |
| 22-30 days | 2 |
| Total | 5 |

kable(days\_since\_last\_case, caption=paste0("Number of countries and days since last case and until ", max(CB\_Data$DateOfNotification)," - complete table"))

Number of countries and days since last case and until 2022-08-24 - complete table

| Country | days\_last\_report | category |
| --- | --- | --- |
| CountryA | 12 | 21 or less |
| CountryB | 22 | 22-30 days |
| CountryC | 3 | 21 or less |
| CountryD | 23 | 22-30 days |
| CountryE | 0 | 21 or less |

# 4 (Optional). Combine case-based and aggregated data and generate a summary of the information  
  
########### Prepare a summary using the aggregated and case-based data #########  
# Get the weekly data and prepare them  
  
# --- for the aggregated ones, add the new cases from the cumulative  
Agg\_Data\_weekly <- Agg\_Data %>%   
 rename(Cases\_total="Cases") %>%   
 mutate(Source="OpenData") %>%   
 group\_by(Country) %>%   
 arrange(Country, DateRep) %>%  
 # after arranging/sorting we have to subtract the previous cumulative   
 # cases from the "current"  
 # since for the first value this cannot happen we use the starting cumulative number  
 mutate(Cases=Cases\_total-lag(Cases\_total)) %>%   
 mutate(Cases=case\_when(is.na(Cases)~Cases\_total,  
 .default = Cases))  
  
  
# --- for the case-based data we need to summarise and then add the cumulative number  
CB\_Data\_weekly <- CB\_Data %>%   
 group\_by(Country, WeekDate) %>%   
 arrange(WeekDate) %>%   
 summarise(Cases=n(), .groups="drop") %>%   
 ungroup() %>%   
 group\_by(Country) %>%   
 mutate(Cases\_total=cumsum(Cases),  
 Source="ReportedCases")   
  
  
# Check the cumulative cases by week in case-based and aggregated data  
# for the most recent week  
Agg\_Data\_total <- Agg\_Data\_weekly %>%   
 group\_by(Country) %>%   
 filter(WeekDate==max(WeekDate)) %>%   
 select(c("Country", "Cases\_total", "Cases", "Source"))  
  
  
CB\_Data\_total <- CB\_Data\_weekly %>%   
 group\_by(Country) %>%   
 filter(WeekDate==max(WeekDate)) %>%  
 select(c("Country", "Cases\_total", "Cases", "Source"))  
  
# using the combined dataset we make it wide and check the difference between   
# the case-based and the aggregated data  
combined\_df <- bind\_rows(Agg\_Data\_total, CB\_Data\_total)  
  
combined\_df\_wide <- combined\_df %>%   
 select(-Cases) %>%   
 pivot\_wider(id\_cols = "Country",   
 names\_from = "Source", values\_from = "Cases\_total", values\_fill = 0) %>%   
 mutate(difference=OpenData-ReportedCases)  
  
# we create a plot with the data by source and country to visualise the difference  
comparison\_plot = ggplot() +  
 geom\_col(data = combined\_df, aes(x = Source, y = Cases\_total, fill = Source)) +  
 facet\_wrap("Country", scales = "free\_y")  
  
comparison\_plot



# select for which countries we will use the case-based data  
cb\_countries <- combined\_df\_wide %>%   
 filter(OpenData <= ReportedCases) %>%   
 pull(Country)  
  
  
# Make a list of the countries for which case-based data were used   
list\_CB\_countries <- paste(cb\_countries, collapse = ", ")  
  
list\_CB\_countries

## [1] "CountryA, CountryD, CountryC"

# from the case based data we select only the countries we want and add them to the aggregated  
cb\_selected <- CB\_Data\_weekly %>%   
 filter(Country %in% cb\_countries) %>%   
 select(Country, WeekDate, Cases\_total, Cases, Source)  
  
# From the aggregated data remove the countries we do not need   
Agg\_selected <- Agg\_Data\_weekly %>%   
 filter(!Country %in% cb\_countries) %>%   
 select(Country, WeekDate, Cases\_total, Cases, Source)  
  
# Create the final dataset with the selected countries from each source  
sum\_db <- bind\_rows(cb\_selected, Agg\_selected)  
  
# Summary of the outbreak based on the both the case-based and aggregated data  
  
since\_date<- glue("{day(max(sum\_db$WeekDate)-weeks(3))} {month(max(sum\_db$WeekDate)-weeks(3), label = T, abbr = F)} {year(max(sum\_db$WeekDate)-weeks(3))}")  
  
since\_date

## 1 August 2022

as\_of\_date<- glue("{day(max(sum\_db$WeekDate))} {month(max(sum\_db$WeekDate), label = T, abbr = F)} {year(max(sum\_db$WeekDate))}")  
  
as\_of\_date

## 22 August 2022

# In last 4 weeks - since we use the weekly data  
since\_last\_update <- sum\_db %>%   
 # taking the data of the last 4 weeks  
 filter(WeekDate>=max(sum\_db$WeekDate)-weeks(3)) %>%   
 group\_by(Country) %>%   
 summarise(n=sum(Cases), .groups = "drop") %>%   
 arrange(desc(n))  
  
since\_last\_update\_total <- sum(since\_last\_update$n)  
  
since\_last\_update\_sentence <- paste0(paste0(since\_last\_update$Country,   
 " (", since\_last\_update$n, ")"),  
 collapse = ", ") %>%  
 stringi::stri\_replace\_last(fixed = ",", replacement = " and")  
  
  
## E-pox situation update, as of the week of `r as\_of\_date`  
  
### Update  
  
# Since the last update on the week of   
since\_date

## 1 August 2022

# and as of the week of   
as\_of\_date

## 22 August 2022

# Number of E-pox cases reported  
since\_last\_update\_total

## [1] 16

# from   
nrow(since\_last\_update) # countries

## [1] 5

since\_last\_update\_sentence

## [1] "CountryA (5), CountryE (5), CountryB (3), CountryC (2) and CountryD (1)"

# Estimating the total number of cases from the final dataset  
totals <- sum\_db %>%   
 group\_by(Country) %>%   
 summarise(Total\_cases=sum(Cases), .groups="drop")   
  
# Making a list of the countries and their total number of cases to be used in the summary  
totals\_sentence <- paste0(paste0(totals$Country,   
 " (", totals$Total\_cases, ")"),  
 collapse = ", ") %>%  
 stringi::stri\_replace\_last(fixed = ",", replacement = " and")  
  
### Summary  
sum(sum\_db$Cases) # E-pox cases have been reported from

## [1] 2045

n\_distinct(sum\_db$Country) # countries

## [1] 5

totals\_sentence

## [1] "CountryA (816), CountryB (415), CountryC (474), CountryD (217) and CountryE (123)"

### Epicurve  
  
epicurve\_db <- sum\_db %>%   
 group\_by(WeekDate) %>%   
 summarise(Cases=sum(Cases)) %>%   
 ungroup()   
  
epicurve\_all <- epicurve\_db %>%   
 mutate(Completeness=case\_when(WeekDate<=max(sum\_db$WeekDate)-days(21)~"Complete",  
 .default="Incomplete")) %>%   
 ggplot() +   
 # Add geom\_bar:  
 geom\_col(aes(x = WeekDate, y=Cases, fill=Completeness)) +   
 scale\_x\_date(date\_breaks = "4 week") +  
 scale\_fill\_manual(values = c("#65B32E","#9CC65A")) +  
 # Remove unnecessary grid lines:  
 theme\_bw()+  
 theme(legend.position = "bottom")+  
 # Update x and y axis labels:  
 labs(x = "Reporting Week",   
 y = "Number of cases",  
 title="E-pox cases")   
  
epicurve\_all

