Solution: SimOb Inject 10 - Case Definition

## 1. Install packages and load libraries

# Load the required libraries into the current R session:  
pacman::p\_load(rio,   
 here,   
 tidyverse,   
 skimr,  
 plyr,  
 janitor,  
 lubridate,  
 gtsummary,   
 flextable,  
 officer,  
 epikit,   
 apyramid,   
 scales)

## 2. Import your data

# Import the clean data set:  
copdata <- rio::import(here::here("data", "Copenhagen\_clean1.rds"))

## 3. Identify cases

### a) Ate a meal at the school dinner

Keep in your dataset only those who ate a meal.

copdata <- copdata %>%   
 filter(meal == TRUE)

### b) Fell ill after the start of the meal

Define “fell ill” as any person having had diarrhoea with OR without blood, OR vomiting. Note that we the concept of having eaten a meal is already included as per one of the steps above:

copdata <- copdata %>%   
 mutate(gastrosymptoms = case\_when(  
 # Those had diarrhoea...  
 diarrhoea == TRUE |  
 #or bloody diarrhoea...  
 bloody == TRUE |  
 # or vomiting, are marked as TRUE (fell ill after the meal)  
 vomiting == TRUE ~ TRUE,  
 # The rest are FALSE. This includes those who ate a meal but had no symptoms (did not fell ill after the meal)  
 .default = FALSE)  
 )

### c) Fell ill within the time period of interest

Create a new meal\_datetime variable as per 11 Nov 2006, at 18:00h:

# Start with copdata:  
copdata <- copdata %>%   
 # Create new column for meal date and time:  
 mutate(meal\_datetime = lubridate::ymd\_hm("2006-11-11 18:00"))

Calculate incubation time and its median:

copdata <- copdata %>%   
 mutate(incubation = onset\_datetime - meal\_datetime,  
 incubation = as.numeric (incubation))  
  
  
median(as.numeric(copdata$incubation), na.rm = TRUE)

[1] 15

We see that the median incubation time is 15 hours. This is useful information, as incubation periods tend to be relatively pathogen-specific. We can now refine the case definition and limit the maximum incubation period from the meal time to 48 hours (2 days) after the meal, as the data points to a fast-acting bacterial toxin or a virus.

copdata <- copdata %>%   
 mutate(case = case\_when(  
 # Those who had symptoms <48h from the meal are cases (TRUE)  
 gastrosymptoms == TRUE &   
 onset\_datetime >= meal\_datetime &  
 onset\_datetime <= (meal\_datetime + days(2)) ~ TRUE,  
 # Those who had symptoms >48h from the meal are non-cases (FALSE)  
 gastrosymptoms == TRUE &   
 onset\_datetime > (meal\_datetime + days(2)) ~ FALSE,  
 # The rest are considered non-cases. Including, those who had no symptoms at all, who have missing data on the onset\_datetime variable, or who had symptoms before eating the meal   
 .default = FALSE)  
 )

Note that we may be incurring in misclassification bias with the code above. The last section indicates that if a person had clinical symptoms before eating the meal, they are considered as non-cases. However, it could be that a person had symptoms before the meal, and yet, still got infected by the pathogen when eating their meal (bad luck, we know…).

Moreover, if you remember from inject 9, there were a couple of people with an dayonset, but no starthour. The code we used (lubridate::ymd\_h with argument truncated = 2) results in dates with missing starthour being converted to date-time, with the missing time being set to 00:00 (midnight). This means that these two people don’t fulfill the case definition criteria because we marked their symptoms started early in the morning of Nov 11 (at 00:00), before the meal time (18:00), and thus, they did not “fell ill within the time period of interest”.

The two situations above are a reminder that you need to be both careful and aware of the implications of your data analysis decisions.

# Tabulate cases:  
janitor::tabyl(dat = copdata, case)

case n percent  
 FALSE 161 0.4270557  
 TRUE 216 0.5729443

Let’s have a look at how many people ate a meal, had symptoms, and were considered as cases after applying our case definition:

copdata %>%   
 summarise(atemeal = sum(meal == TRUE),  
 hadsympt = sum(gastrosymptoms == TRUE),  
 nb\_cases = sum(case == TRUE)  
 )

atemeal hadsympt nb\_cases  
1 377 216 216

# 4. Export clean data

Finally, we can save the cleaned data set before proceeding with descriptive analysis.

rio::export(x = copdata,   
 file = here::here("data", "Copenhagen\_clean2.rds"))