# R Take-Home Assignment 3: OWID COVID-19 (Loops & Conditionals)

Your Name

2025-10-01

## Overview

In this assignment you will work with the **Our World in Data (OWID) COVID-19 dataset**, which tracks the spread and impact of the pandemic globally. We will focus on daily new cases for three countries — the Netherlands, Germany, and Italy — to explore how case numbers change over time.

Because daily counts can be noisy (for example, due to weekend reporting effects), you will use rolling averages and logical flags to highlight important peaks. This exercise gives practice with **loops and conditionals**, two fundamental programming tools that help automate repetitive tasks and make decisions based on data.

- **Source:** OWID compact CSV
- **Skills:** importing CSVs, subsetting, rolling averages, logical flags, time-series plots
- **Deliverable:** knitted **HTML** plus your completed `.Rmd`

## Preparation

You will need: `dplyr`, `ggplot2`, `readr`, `zoo`.

```
library(dplyr)
library(ggplot2)
library(readr)
library(zoo)
```

## 1. Load and subset the data

The dataset is large and contains many countries. Since we want to compare trends in only three European countries, we first need to import the full dataset and then narrow it down.

**Task:** Load the OWID dataset from the provided URL. Subset it to the Netherlands, Germany, and Italy. Convert the `date` column to type `Date`.

```
url <- 'https://catalog.ourworldindata.org/garden/covid/latest/compact/compact.csv'
owid <- read_csv(url, show_col_types = FALSE)
owid_small <- owid %>% filter(country %in% c('Netherlands','Germany','Italy')) %>% mutate(date = as.Date
head(owid_small)
```

```
## # A tibble: 6 x 61
##   country date       total_cases new_cases new_cases_smoothed
##   <chr>   <date>           <dbl>     <dbl>              <dbl>
## 1 Germany 2020-01-01          NA        NA                 NA
## 2 Germany 2020-01-02          NA        NA                 NA
## 3 Germany 2020-01-03          NA        NA                 NA
## 4 Germany 2020-01-04           1         1                 NA
## 5 Germany 2020-01-05           1         0                 NA
## 6 Germany 2020-01-06           1         0                 NA
## # i 56 more variables: total_cases_per_million <dbl>,
## #   new_cases_per_million <dbl>, new_cases_smoothed_per_million <dbl>,
## #   total_deaths <dbl>, new_deaths <dbl>, new_deaths_smoothed <dbl>,
## #   total_deaths_per_million <dbl>, new_deaths_per_million <dbl>,
## #   new_deaths_smoothed_per_million <dbl>, excess_mortality <dbl>,
## #   excess_mortality_cumulative <dbl>,
## #   excess_mortality_cumulative_absolute <dbl>, ...
```

## 2. Compute a 7-day moving average

Raw daily case counts jump up and down due to reporting schedules. A moving average smooths these fluctuations, making it easier to see real trends.

**Task:** For each of the three countries, calculate a 7-day rolling average of `new_cases`. Store it in a new column `ma7`.

```
owid_small <- owid_small %>% arrange(country, date) %>% group_by(country) %>% mutate(ma7 = rollmean(new_
head(owid_small)
```

```
## # A tibble: 6 x 62
##   country date       total_cases new_cases new_cases_smoothed
##   <chr>   <date>           <dbl>     <dbl>              <dbl>
## 1 Germany 2020-01-01          NA        NA                 NA
## 2 Germany 2020-01-02          NA        NA                 NA
## 3 Germany 2020-01-03          NA        NA                 NA
## 4 Germany 2020-01-04           1         1                 NA
## 5 Germany 2020-01-05           1         0                 NA
## 6 Germany 2020-01-06           1         0                 NA
## # i 57 more variables: total_cases_per_million <dbl>,
## #   new_cases_per_million <dbl>, new_cases_smoothed_per_million <dbl>,
## #   total_deaths <dbl>, new_deaths <dbl>, new_deaths_smoothed <dbl>,
## #   total_deaths_per_million <dbl>, new_deaths_per_million <dbl>,
## #   new_deaths_smoothed_per_million <dbl>, excess_mortality <dbl>,
## #   excess_mortality_cumulative <dbl>,
## #   excess_mortality_cumulative_absolute <dbl>, ...
```

## 3. Flag large values

Public health authorities often monitor when case numbers cross certain thresholds. Here, we will flag days with especially high case numbers.

**Task:** Add a logical column `flag_high` that is `TRUE` if `ma7 > 10000` and `FALSE` otherwise.

```
owid_small <- owid_small %>% mutate(flag_high = ma7 > 10000)
head(owid_small)
```

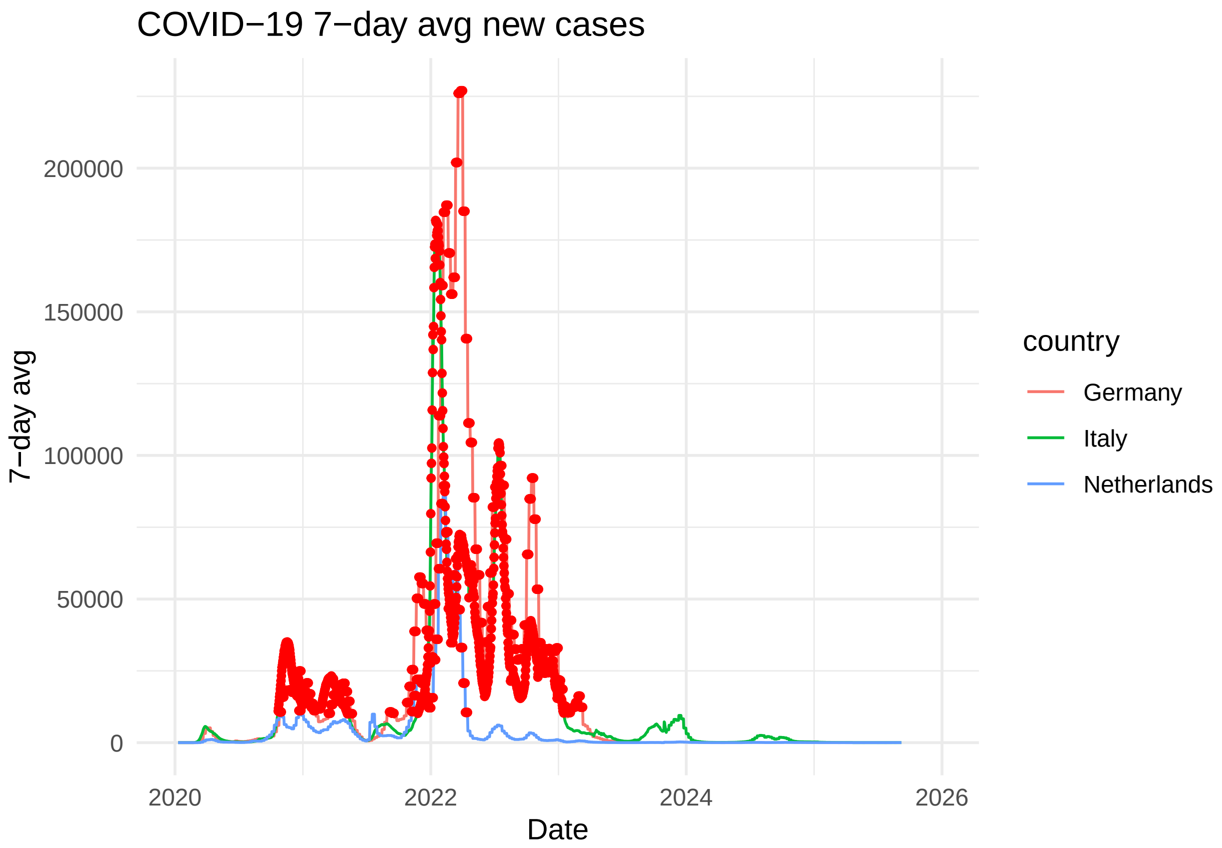
```
## # A tibble: 6 x 63
##   country date       total_cases new_cases new_cases_smoothed
##   <chr>   <date>           <dbl>     <dbl>              <dbl>
## 1 Germany 2020-01-01          NA        NA                 NA
## 2 Germany 2020-01-02          NA        NA                 NA
## 3 Germany 2020-01-03          NA        NA                 NA
## 4 Germany 2020-01-04           1         1                 NA
## 5 Germany 2020-01-05           1         0                 NA
## 6 Germany 2020-01-06           1         0                 NA
## # i 58 more variables: total_cases_per_million <dbl>,
## #   new_cases_per_million <dbl>, new_cases_smoothed_per_million <dbl>,
## #   total_deaths <dbl>, new_deaths <dbl>, new_deaths_smoothed <dbl>,
## #   total_deaths_per_million <dbl>, new_deaths_per_million <dbl>,
## #   new_deaths_smoothed_per_million <dbl>, excess_mortality <dbl>,
## #   excess_mortality_cumulative <dbl>,
## #   excess_mortality_cumulative_absolute <dbl>, ...
```

## 4. Plot with flagged points

A visualisation makes it easy to compare the three countries. Highlighting the flagged points will help you see when major peaks occurred.

**Task:** Create a line plot of the 7-day averages (ma7) over time for each country. Use different colours for the countries, and mark flagged points with an additional symbol (e.g. points in red).

```
ggplot(owid_small, aes(date, ma7, color=country)) + geom_line() +
  geom_point(data=subset(owid_small, flag_high), size=1, color='red') +
  labs(title='COVID-19 7-day avg new cases', x='Date', y='7-day avg') + theme_minimal()
```



## 5. Reflection

Finally, think about the patterns. Which country had the highest peaks? Did they occur at the same time? Did the threshold capture the waves you would expect?

**Task:** Write 3–4 sentences comparing the trends across the three countries and reflecting on how loops and conditionals helped structure the analysis.

*Write your reflection here…*