

Load libraries

First, we need to load the following libraries:

```
library(here)
library(readxl)
library(janitor)
library(dplyr)
library(stringr)
library(scales)
```

Download the dataset

You can access the full database here by submitting a query for the data you need. You will need to register with the database first before you can submit a query. For the purpose of this exercise, we will provide you with the full dataset that starts from 1900. This dataset is as of March 24, 2021.

Import dataset

Using the package `readxl` we can import the dataset that's in Excel format. If you preview the raw file in Excel, you will find that the first six rows are used as description of the dataset, therefore we need to make sure that R does not read rows that are not part of the dataset. The function `read_excel` provides a convenient option called `skip`, which tells R how many rows to skip before reading the dataset. In this case, we need to skip six rows.

```
em_dat <- readxl::read_excel(here::here("data/em-dat",
  "emdat-public-2021-03-24-query-uid-DuX1xq-raw.xlsx"),
  skip = 6)
```

Clean dataset

Using the `janitor` package, we can use the function `clean_names` to create consistent-looking variable names.

```
em_dat <- janitor::clean_names(em_dat)
```

Instead of recreating the object `em_dat` again, we can combine the previous two functions using the 'pipe' operator, `%>%`, which is loaded with the `tidyverse` package. This operator uses the previous output as the new input of the subsequent function.

```
em_dat <- readxl::read_excel(here::here("data/em-dat",
  "emdat-public-2021-03-24-query-uid-DuX1xq-raw.xlsx"),
  skip = 6) %>%
  janitor::clean_names()
```

In the above code chunk, we did the following:

1. Imported the dataset using `read_excel` function.
2. Took that dataset and cleaned all the variables' names using the `clean_names` function.
3. Called the output of the previous two processes `em_dat`.

The pipe operator is a powerful function that can reduce the amount of code you need to write.

```
em_dat_sub <- em_dat %>%
  dplyr::select(iso, country, year, disaster_type,
    total_deaths, no_injured, no_affected,
    no_homeless, total_affected, total_damages_000_us) %>%
  dplyr::filter(str_detect(disaster_type,
    "Drought|Extreme temperature|Flood|Storm|Wildfire"))
```

From 1900 to 2021, 46.8% of recorded disaster events are from the following types: Drought, Extreme temperature, Flood, Storm, and Wildfire. Furthermore, within the same time frame, 52.9% of total deaths are attributed to the disaster types that are more exacerbated by climate change. As for people affected by these disaster, approximately 96.3% are because of these five types of climate-related disasters.

Here we sum over disaster data by disaster type, year, and country. For example, we count all the damages that occurred in Bangladesh for the year 2018 by floods.

```
em_dat_climate <- em_dat_sub %>%
  dplyr::filter(as.numeric(year) >= 1990 &
    !(is.na(total_deaths) & is.na(no_injured) &
      is.na(no_affected) & is.na(no_homeless) &
      is.na(total_affected) & is.na(total_damages_000_us))) %>%
  dplyr::group_by(iso, country, year, disaster_type) %>%
  dplyr::summarise_all(funs(sum), na.rm = TRUE) %>%
  dplyr::mutate(country = str_remove(country,
    " \\\(the\\\)")) %>%
  dplyr::mutate(country = str_replace_all(country,
    c(`Korea \\\(the Republic of\\\) ` = "Republic of Korea",
      `Congo \\\(the Democratic of\\\) ` = "Democratic Republic of the Congo",
      `Tanzania, United Republic of ` = "United Republic of Tanzania",
      `Taiwan \\\(Province of China\\\) ` = "China")))) %>%
  dplyr::na_if(., 0)

utils::write.csv(em_dat_climate, here("scripts/cleaning/em-dat",
  "em-dat-clean.csv"), row.names = FALSE)

rm(em_dat, em_dat_climate, em_dat_sub)
```

Export as an R script for future use

Only run this chunk manually once within the .Rmd file. It produces an error when knitting it as a whole because of chunk label duplicates. As of May 12, 2021, there hasn't been a viable solution to run the code below when as part of the knitting process.

```
knitr::purl("em-dat-clean.Rmd", "em-dat-clean.R")  
knitr::write_bib(.packages(), "packages.bib")
```

Software used

Firke, Sam. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*, 2021. <https://github.com/sfirke/janitor>.

Müller, Kirill. *Here: A Simpler Way to Find Your Files*, 2020. <https://CRAN.R-project.org/package=here>.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2021. <https://www.R-project.org/>.

Wickham, Hadley. *Stringr: Simple, Consistent Wrappers for Common String Operations*, 2019. <https://CRAN.R-project.org/package=stringr>.

Wickham, Hadley, and Jennifer Bryan. *Readxl: Read Excel Files*, 2019. <https://CRAN.R-project.org/package=readxl>.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. *Dplyr: A Grammar of Data Manipulation*, 2021. <https://CRAN.R-project.org/package=dplyr>.

Wickham, Hadley, and Dana Seidel. *Scales: Scale Functions for Visualization*, 2020. <https://CRAN.R-project.org/package=scales>.