Tidyverse lab

Ariane Ducellier

9/1/2023

This lab will make use of tidyverse functions for data transformation to extract and manipulate metadata of seismic stations in the Northern California Seismic Network.

We want to download seismic waveforms from a seismic data archive of specific earthquakes. We are not sure what seismic sensors (stations) are operating at that time. The list of stations available in the seismic networks has more than 6000, that's way too many! So we want to filter only the seismic stations that are relevant for the research.

Libraries

Load the necessary libraries.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Read the data

This is the address of the website to download the data: NCEDC metadata

Task 1 First, you need to load the data into a tibble. You may use the following header:

```
header = c("Station", "Network", "Channel", "Location", "Rate",
           "Start_time", "End_time", "Latitude", "Longitude",
           "Elevation", "Depth", "Dip", "Azimuth", "Instrument")
```

Task 2 Now, we need to convert Start_time and End_time into a datetime format.

It turns out than only the following channels are relevant for the work we want to do:

- BHE, BHN, BHZ, BH1, BH2,

- EHE, EHN, EHZ, EH1, EH2,
- HHE, HHN, HHZ, HH1, HH2,
- SHE, SHN, SHZ, SH1, SH2.

That is, we want the channels that start with B, E, H or S and which second letter with an H.

Task 3 Filter the dataset to keep only the rows with the channels as defined above.

The seismic data archive that we are working on starts on 2007/07/01 and ends on 2009/07/01. We are only interested in stations that started recording before 2009/07/01 and ended recording after 2007/07/01.

Task 4 Filter the dataset to keep only stations that started recording before 2009/07/01 and ended recording after 2007/07/01.

The earthquakes we are interested in are located at latitude = 40.09N and longitude = -122.87E.

We want to keep the stations that are located less than 100 km from the earthquakes.

Task 5 Filter the dataset to keep only stations that are within 100 km from the earthquakes.

You may use this function to compute the distance from the station to the earthquakes using the latitude and the longitude:

```
get_dists <- function(df){
  lat0 = 40.09000
  lon0 = -122.87000
  a = 6378.136
  e = 0.006694470
  dx = (pi / 180.0) * a * cos(lat0 * pi / 180.0) /
    sqrt(1.0 - e * e * sin(lat0 * pi / 180.0) * sin(lat0 * pi / 180.0))
  dy = (3.6 * pi / 648.0) * a * (1.0 - e * e) /
    ((1.0 - e * e * sin(lat0 * pi / 180.0) * sin(lat0 * pi / 180.0)) ** 1.5)
  df$x = dx * (df$Longitude - lon0)
  df$y = dy * (df$Latitude - lat0)
  df$Distance = "^"("^"(df$x, 2.0) + "^"(df$y, 2.0), 1/2)
  return (df$Distance)
}
```

For the final step, we only want to keep the columns: Station, Network, Channel, Location, Latitude, Longitude, Elevation, Depth, Start_time, End_time. We want to group the stations:

- **Task 6** First, group the stations by Station, Network, Channel, Location, Latitude, Longitude, Elevation, Depth, and compute the minimum start time and the maximum end time.
- **Task 7** Second, group the stations by Station, Network, Location, Latitude, Longitude, Elevation, Depth, and concatenate all the channels for a given station in a single string, separated by a comma.