Tidying ECCC Air Quality Data

David Hall

2020-11-18

Source

Air quality data is taken from Environment and Climate Change Canada's (ECCC) National Air Pollution Surveillance Program (NAPS). This specific applet uses *annual continuous hourly measurements*. Each of these files contains the hourly measurements of a single pollutant (i.e. O_3) across all operational NAPS station for the given year.[1] There is at least a year delay between collection and publication of data.[2]

Data organization

Each yearly pollutant dataset can be downloaded as a `.csv` from the ECCC website. As listed on the dataset preamble:

- Measurements are reported in parts-per-billion (ppb)
- Data is ending local standard time. (i.e. H01 is the hourly measurements from 00:00 to 01:00).
- Zeros are valid values
- `-999` denotes no data available.

However, the structure of this dataset have some incompatible elements with the `tidyverse` ecosystem, these include:

- Matrix style layout, with each row corresponding to a day, and columns for each hourly report.
- Bilingual headers separated by "//"
- Separation of *date* and *time*

Converting ECCC NAPS data for Applet

Packages used include:

```r
library(tidyverse)
library(anytime) # Quicker than tiverse's lubridate package
```

Importing and Tidying Data

I wrote a quick function called `ECCCTidy` which tidies a single ECCC NAPS `.csv` file. In other words, it converts the ECCC 'matrix' layout into a 'long' layout where each row is the measurement of that specific pollutant at a given date-time and location. *Note* that all columns

[1] Files can be found on the ECCC website here

[2] At the time of writing, the most recent annual report is from 2018

start with a capital letter to maintain consistency with the original NAPS dataset.

```
ECCCTidy <- function(file, rows = Inf){

  # Getting pollutant from file name
  chem <- sub("\\_.*", "", file)


  # Skipping ECCC header when importing file
  df <- read_csv(file, skip = 7, n_max = rows)


  # Actually tidying ECCC file
  df <- df %>% rename_all(funs(gsub("\\..*","", make.names(names(df))))) %>%
    pivot_longer(
      cols = starts_with("H"),
      names_to = "Hour",
      values_to = chem)  %>%
    mutate(Date_time = anytime(paste(Date, str_sub(Hour, -2, -1), ":00"))) %>%
    select(-c(Date, Hour, Pollutant)) %>%
  relocate(Date_time, .before = chem)

  df
}
```

Transforming Data for Applet

I wrote another function which will combine tidied O_3 and NO_2 datasets,[3] and calculate the O_x value at each given time. All `-999` values are converted to `NA`, and therefore are not used in any subsequent plotting/calculations. I also added a `rows` input where you can specify the number of rows you want to import from the `.csv`. The default is `Inf` (read every row), but you can specify smaller numbers when testing stuff.

```
# Test data to combine
NO2 <- ECCCTidy("NO2_2018.csv", rows = 2500)
O3 <- ECCCTidy("O3_2018.csv", rows = 2500)

# Row outline of function, will need to clean up and expand to include other pollutants (i.e. SO2)
ECCCCombine <- function(O3, NO2){
  df <- O3 %>%
  inner_join(NO2) %>%
  na_if(-999) %>%
```

[3] Note this uses inner_join, so only O3 and NO2 values from stations found in BOTH datasets will be included.

```
  mutate(Ox = O3 + NO2)
}
```

The `ECCCCombine` function is pretty basic right now, but in the future I hope to expand it so that :

- Users can directly specify the ECCC files they want to use
- Multiple pollutants can be included such as SO_2, NO, and maybe stuff like $PM_{2.5}$; essentially any ECCC formatted `.csv`.
- Clean up the naming conventions to avoid duplication, etc.

The end result of this function will be saved as a `.csv` and *this will be the file uploaded to and used by the applet.*

Quickly plotting the combined ECCC data shows that the NAs are properly plotted as gaps in the timeseries, and that everything is kosher.

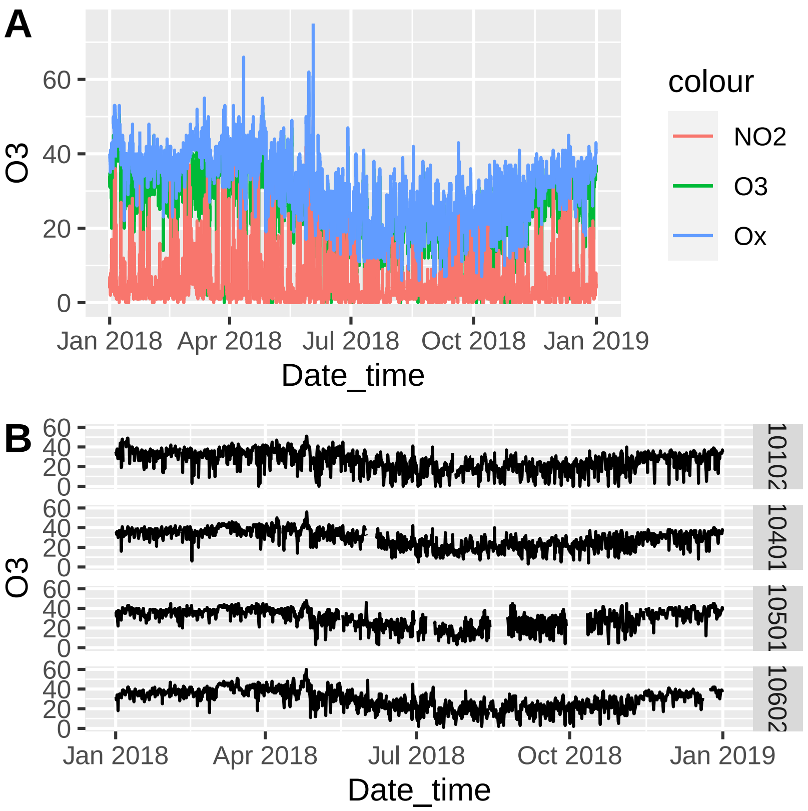

Figure 1: Example plots from datasets. (A) Plot at single NAPS station of O3, NO2, and Ox values. (B) O3 values for different stations. Note gaps in time series

Notes for the downstream stuff

- As it stands, combining all of the ECCC data into columns is easier than a massively long .csv (plus if I want to be able to export datasets to excel, this is the way (will need to fix date though…)). However., the 'long' format is easier w/ ggplot2, so it may be a good idea to convert to 'long' the subset of data to be plotted to take advantage of ggplot2 features…

- Will need to add a rolling average feature to applet