Original author:
Name: Brooke Anderson
Title: Assistant Professor
Department: Environmental & Radiological Health Sciences
University: Colorado State University

Open-source software to aggregate weather data for health studies

In environmental epidemiology, we often study the links between ambient exposures—like temperature, severe weather, air pollution, and floods—and human health risks. However, the signal of these risks can often be difficult to pick out from the noise of normal variation in health outcomes. Therefore, environmental epidemiologic studies can sometimes require a very large-scale analysis, including many cities over many years, to precisely estimate the risks related to these ambient environmental exposures. These multi-city, multi-scale environmental health studies require researchers to collect, clean, and aggregate large and complex exposure datasets.

We used the support of this grant to develop free, open-source tools to facilitate epidemiologic studies of water-related weather exposures, including exposures related to extreme rainfall, floods, and tropical storms. US agencies, like the National Oceanic and Atmospheric Administration (NOAA) and the United States Geological Survey (USGS), monitor exposure data, including precipitation and streamflow, and make historical datasets of these exposures available. However, collecting this data for a study has typically required identifying all station monitors near study sites, pulling data for all monitors, and aggregating monitor data to generate a study-site average (including removing monitors with too much missing data over the study period). This data collection has typically been done with point-and-click web interfaces that require several manual steps to collect data for each monitor, with later data cleaning and aggregating on a local computer. For a study of a hundred or more cities (which is not unusual for environmental epidemiology studies), the collection of environmental exposure data can require separately collecting the data for hundreds to thousands of monitors.

Recently, many US agencies have created or improved web services to allow their historical data to be pulled directly from their web-based databases, rather than requiring the use of a point-and-click interface. In essence, the agency gives researchers the rules for finding the right web address for the data they want, based, for example, on a monitor identification number and a date range. Because this process has consistent rules, a researcher can write software that will take a monitor identification number and date range as inputs and will pull the data from the agency's web database and save it to his or her local computer, rather than getting the data for each monitor one at a time. The idea behind this process is similar to web scraping, but most agencies now provide an API that formalizes the process, provides a greater guarantee that the rules for finding web-based data will remain fairly stable, and allows the agency more oversight in ensuring no one abuses this process.

The graduate student funded on this Water Center grant, Rachel Severson, led our efforts to develop open-source software to use these web services, so that a researcher can use simple code within the R programming language to collect county-level datasets of daily or hourly weather observations. The result is the open source R package `countyweather`, which is currently available on GitHub (see Figure 1). We developed this package so that a researcher only needs to input US county identifiers (FIPS codes) and desired date ranges, since health data is often available only aggregated to the county level, while surface observations of weather conditions are typically monitor-based. The software will automatically query the NOAA web-based databases to: (1) find all monitors within the county; (2) pull all available weather data from those monitors for any weather variables requested (e.g., precipitation, wind speed, temperature); (3) filter out any monitors with a coverage over the study period that is below a user-specified threshold (for example, a researcher could choose to filter out any monitors with less than 75% non-missing data over a study period); (4) average across monitors to generate a county-level average for the relevant time period (currently, the package can pull either hourly or daily measurements, so this step creates either an hourly or daily county-wide average); and (5) write out exposure datasets, datasets describing the stations included in that exposure data, and maps of the locations of the stations used to the researcher's local computer. Because this is all coded as a simple function, it is straightforward for a researcher to write a loop to collect and save data for many counties in an efficient way, and it is easy to later update and re-run the code to collect exposure data if the study size increases in the future.

In addition to this package, we also created a package (`countytimezones`, available on GitHub and CRAN) for converting weather data timestamps to local time. This package includes a dataset we created with Oleson time zones for every US county, and so it captures day light savings time, including changes over the years in day light savings practice in different counties. This package allows us to convert weather observations to a time zone appropriate for aggregating with health data, which is typically given based on local time. We also have begun developing software that can use USGS streamgage data to assess flood-based exposures for US counties. Further, undergraduate researcher Ziyu Chen, who was supported by an undergraduate research program and advised by PI Dr. Brooke Anderson, developed a package (`noaastormevents`, available on GitHub; see Figure 2) that can pull, sort, and map data from NOAA's Storm Events database and can pair this data with hurricane tracking data. This package allows researchers to identify events, including floods and flash floods, that occurred the same time and within a certain distance of a tropical storm path.

Several of these software projects are spin offs of ideas and code generated during our Spring 2016 Hackathon (described below) and involve students or researchers who attended our Fall 2015 Hydro-Epidemiology workshop series (also described below), two other activities supported by this grant.

Fall 2015 Hydro-epidemiology workshop series

PIs Brian Bledsoe and Brooke Anderson and researcher Joel Sholtes coordinated and led a workshop series funded by this grant on hydro-epidemiology. These workshops were held during four Fridays in October and November 2015 and brought together students and postdoctoral researchers in Epidemiology and Engineering to talk about water-related research topics spanning engineering and epidemiology and to help us explore potential areas of future collaboration. Other professors from engineering and epidemiology (Sybil Sharvelle and Sheryl Magzamen) also joined for parts of the workshops.

Sessions were divided into hour-long segments lead by students or professors and included a combination of research talks, describing current research projects, or in-depth analysis of journal articles. The topics we covered included: flood hazards and related health risks, watershed hydrology, Opportunistic Premise Plumbing Pathogens (OPPPs) including Legionella, gray water and health risks associated with its use, and disinfection by-products. Based on the interest from the students, we might explore expanding this idea in the future years into a 1-credit seminar course jointly listed for Engineering and Epidemiology students. Several of the people who attended these workshops were later involved in our Spring 2016 hackathon (described below) and in some of the open-source software development (described above).

Spring 2016 Weather Data Hackathon

On April 20 and 21, 2016, PI Brooke Anderson led a Weather Data Hackathon. Around 15 people participated in this hackathon, including undergraduate students, graduate students, postdocs, and professors. While some hackathons have teams compete against each other, this hackathon was a collaborative effort. With the help of Co-PI Sheryl Magzamen, we developed two weather data challenges, each focused on environmental health-related exposures, and hackathon participants worked together to develop code and find tools for these challenges. The first challenge was to find and explore as much weather exposure data as possible for three major hurricanes (Andrew in 1992, Cyclone Tracy in Australia in 1974, and Tropical Storm Bilis in China in 2006). Participants looked for existing R packages and online weather data APIs that would allow them to create datasets characterizing precipitation, wind, and flooding during these events, with an aim to creating code and tools that can generalize to any tropical storm. The second challenge had similar goals, but for wildfires in Alaska, and included collecting data on lightning strikes and weather conditions before and during fires.

One key challenge was to ensure that the code developed could serve as a seed for developing future software. Therefore, all the participants used GitHub to fork the same repository locally to their computers, and then pushed completed code to a central repository at the end of each hackathon session. This introduced many of the participants to GitHub (and version control more generally) as a tool for collaborative scientific research. Some of the ideas and code developed during this Hackathon we have since worked into some of our open source software projects described above.

Figure captions

Figure 1. This figure illustrates use of the `countyweather` open source R package developed by graduate student Rachel Severson and PI Brooke Anderson with the support of this grant. This package allows researchers to pull and aggregate county-level time series of weather exposure directly from NOAA web databases. This example shows how a function in this package can be used to pull and explore precipitation data from Miami-Dade County, FL, during Hurricane Andrew (1992). The package includes functions to give information on locations of weather monitors used and the number of monitors reporting for each observation (you can see that many monitors failed during Andrew!).

Figure 2. Following up on ideas and code generated through our spring Hackathon, undergraduate Ziyu Chen worked with PI Brooke Anderson to develop the open source R package `noaastormevents`, to pull data from NOAA's online Storm Events files and pair it with hurricane tracking data. This figure shows an example of using the `noaastormevents` R package to find, summarize, and map events that were near in time and location to the track of Hurricane Floyd (1999), which caused devastating flooding along the East Coast.

Figure 3. Around 15 undergraduate students, graduate students, postdoctoral scholars, and professors participated in the Spring 2016 Weather Data Hackathon that this grant helped fund. This figure shows some of the participants in action.