Geocoding The EMDAT Dataset

Ram Mukund Kripa

7/15/2021

## Abstract

EM-DAT is a very popular disaster dataset among researchers in the field of Earth and Environmental science. However, it comes with challenges, in its format and otherwise, that make it unsuitable to a tidy data science workflow in popular environments like R. The Wrangler for Emergency Events Database, weed for short, aims to provide researchers with the right tools to easily prepare EM-DAT data for downstream analysis, as well as some exploration of the same. The sample workflow here demonstrates the capabilities of weed through its functions.

## Keywords

## Introduction

WEED (Wrangler for Emergency Events Database) is an R Package designed to help researches in the field of climatological disaster studies better analyse EM-DAT and other related datasets. As will be explained in later sections of this paper, EM-DAT contains a lot of useful climatoligical and sociological data pertaining to natural disasters. However, it has a few drawbacks in its presentation of geolocation data as well as its unsuitability to a tidy data science workflow.

WEED attempts to solve and eradicate these issues to help researchers glean valuable insights from their EM-DAT dataset!

## Structure of The Package

The structure of the Wrangler for Emergency Events Database is best explained through a sample workflow.

### Loading

First, one may download the subset of EM-DAT required from the public EM-DAT query tool located at [this link](https://public.emdat.be/). From here, researchers can exploit the loading functionality of WEED. EM-DAT files tend to be excel sheets with a few lines of metadata, followed by the subsection of the dataset downloaded. The read\_emdat function allows researchers to load the dataset as a List in one of two ways: the dataset, as well as its metadata, or just the dataset.

sample\_data <- read\_emdat(here("data", "emdat\_public\_2021\_01\_12\_full.xlsx"))  
summary(sample\_data)

| Sample Data | | | |
| --- | --- | --- | --- |
| Dis.No | Country | Location | Year |
| 1900-9002-CPV | Cabo Verde | Countrywide | 1900 |
| 1900-9001-IND | India | Bengal | 1900 |
| 1901-0003-BEL | Belgium | Hornu | 1901 |
| 1902-0012-GTM | Guatemala | Quezaltenango, San Marcos | 1902 |
| 1902-0003-GTM | Guatemala |  | 1902 |
| 1902-0010-GTM | Guatemala |  | 1902 |

### Exploration

The next step might be to explore the given data. One of the most pressing issues with EM-DAT at the moment is its geocoding data. The fact that a very small fraction of disasters have usable geocoding data, in terms of Latitude and Longitude, severely hampers location analysis. Another issue is the presence of multiple locations per disaster.

| Sample Data: Problems | | | | | |
| --- | --- | --- | --- | --- | --- |
| Dis.No | Country | Disaster.Type | Location | Latitude | Longitude |
| 2000-0919-USA | United States of America (the) | Storm | Alabama, Georgia, Louisiana, North Carolina, South Carolina, Tennessee, Virginia, New York, Pennsylvania, Massachussetts provinces |  |  |
| 1964-0092-USA | United States of America (the) | Flood |  |  |  |
| 1928-0024-CAN | Canada | Earthquake | Burin Peninsula, Newfoundland | 48.60 N | 58.00 W |
| 1987-0492-USA | United States of America (the) | Transport accident | Texas |  |  |
| 1998-0212-USA | United States of America (the) | Extreme temperature | Arizona, Florida, Colorado, Texas, Oklahoma, LA |  |  |

Having a single latitude and longitude refer to multiple locations also makes analysis significantly more challenging, and nigh on impossible. To counteract these problems, the recommended Weed workflow is to change the unit of analysis of the data frame from “one row per disaster” to “one row per disaster-location pair”. This process will henceforth be referred to as “locationizing”.

locationized\_df <- sample\_df %>%  
 split\_locations(column\_name = "Location")

| Locationized Sample Data | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Dis No | Country | Disaster Type | Latitude | Longitude | location\_word | uncertain\_location\_specificity |
| 2000-0919-USA | United States of America (the) | Storm |  |  | alabama | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | georgia | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | louisiana | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | north carolina | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | south carolina | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | tennessee | FALSE |

The split\_locations function allows users to execute the process of “locationizing”. It comes with a default method of splitting, defined by its parameters ‘dummy\_words’, which indicate which words to altogether remove from the location strings, and ‘joiner\_regex’, which indicates how the locations have been concatenated to form the location strings.

A few notes on the kinds of locations we get:

Collections of locations are represented in a wide variety of ways in EM-DAT, owing to the fact that this is a dataset that is manually filled row-by-row. Since there is only one location column, a solitary string is associated with each disaster, and there is a need for a delimiter to distinguish between separate locations affected by the disaster. Common methods of representation of abritrary locations (here called A, B and C) include “A, B, and C”, “(1) A (2) B (3) C” and “A (B and C).” Such delimiters are handled by the location extraction algorithm, but others may be accounted for using the joiner\_regex parameter.

When parentheses are present in location strings, like “Berkeley (California)” or “California (Berkeley, Emeryville, Alameda)”, we see that problems can arise in regard to differing levels of specificity. We are unsure of the specificity of each these locations just by their position relative to their associated parentheses. Hence, we flag these with an uncertainty column called “uncertain\_location\_specificity”.

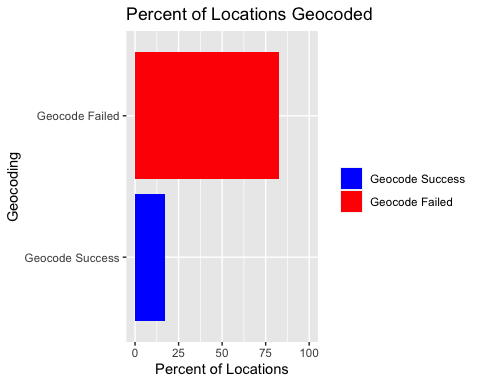
locationized\_sample\_data <- sample\_data[['disaster\_data']] %>%  
 split\_locations(column\_name = "Location")

The locationized Data frame is compatible with exploratory functions like percent\_located\_locations and percent\_located\_disasters which allow for easy visualization of the coverage provided in the data, with respect to latitudes and longitudes.

### Percent Located Location-Disaster Pairings

This function displays what proportion of the locations (obtained by split\_locations) have been geocoded in the input dataset.

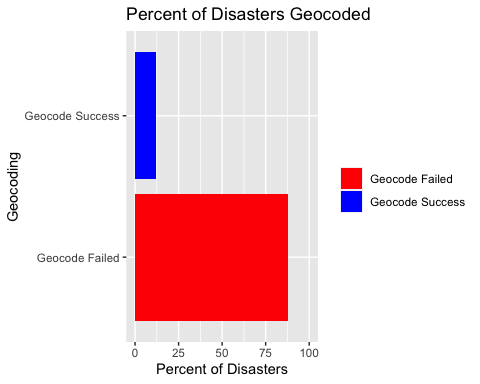
locationized\_sample\_data %>%  
 percent\_located\_locations(lat\_column = "Latitude",  
 lng\_column = "Longitude")



### Percent Located Disasters

This is quite self explanatory, in the sense that it displays what proportion of the disasters have been geocoded in the input dataset.

locationized\_sample\_data %>%  
 percent\_located\_disasters(lat\_column = "Latitude",  
 lng\_column = "Longitude")



As we can see, the coverage is very sparse. Certainly not enough for proper analysis.

Once the data has been locationized, it is ready to be geocoded.

### Geocoding

Weed uses the free [geonames API](https://www.geonames.org/) to geocode each location. To use this functionality, one must first create a free account and then supply their username to the geocode function in Weed. This function comes with a few options, depending on the kind of analysis that is being performed. Researchers can utilize the n\_results parameter to get the n closest matches to the input location and decide which one to use. The unwrap parameter also allows researchers to keep the geocoded data in a nested Data frame structure, possibly good for exporting, or in unwrapped from, where each lat and long gets a separate column (lat1, lng1, lat2, lng2, etc.)

geocoded\_df <- locationized\_df %>%  
 geocode(unwrap = FALSE, geonames\_username = sample\_username)

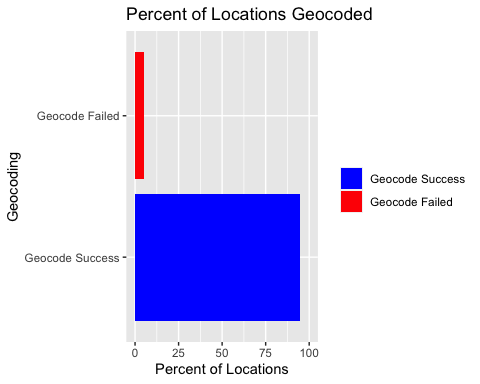
| Geocoded Data | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dis No | Country | Disaster Type | Latitude | Longitude | location\_word | uncertain\_location\_specificity | lat | lng |
| 2000-0919-USA | United States of America (the) | Storm |  |  | alabama | FALSE | 34.60739 | -86.97977 |
| 2000-0919-USA | United States of America (the) | Storm |  |  | georgia | FALSE | 33.69277 | -84.39957 |
| 2000-0919-USA | United States of America (the) | Storm |  |  | louisiana | FALSE | 30.12595 | -92.00939 |
| 2000-0919-USA | United States of America (the) | Storm |  |  | north carolina | FALSE | 34.00071 | -81.03481 |
| 2000-0919-USA | United States of America (the) | Storm |  |  | south carolina | FALSE | 34.00071 | -81.03481 |
| 2000-0919-USA | United States of America (the) | Storm |  |  | tennessee | FALSE | 35.80000 | -86.50000 |

If your dataset has more than 1000 records after locationizing it is advisable to use our function geocode\_batches which includes a cooldown time between batches so that issues with limits on the number of queries one can make to the geonames API can be avoided.

Further exploration with the percent\_located\_locations and percent\_located\_disasters functions is advisable, to visualize the success of our geocoding. As the data is indeed locationized, a choice must be made as to how to decide if a disaster has been “located”. Two popular choices are any and all. “Any” considers a disaster located if any one of its constituent locations has valid lat-long data, while “all” requires every constituent location to be geocoded. These can be set by the how parameter of percent\_located\_disasters, which also allows user defined functions!

### Percent Located Locations

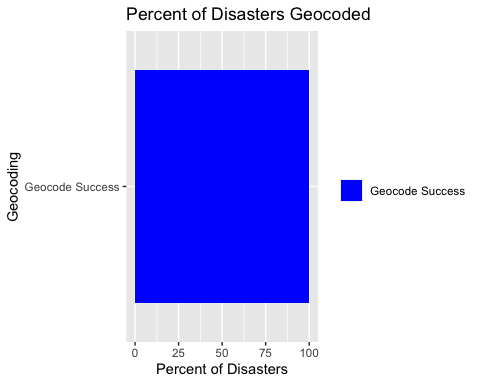
geocoded\_df %>%  
 percent\_located\_locations()



### Percent Located Disasters

A slight change from before as, now, we have different geocoding for each location within a disaster. We can decide whether to consider a disaster to be geocoded if any of the locations have been geocoded, or all, or some specific percentage or combination. “How” and “any” are default options but we support user-defined aggregation functions as well.

geocoded\_df %>%  
 percent\_located\_disasters(how = "any")



The goal with the geocoding and subsequent exploration was to provide as much flexibility and modularity to this step of the workflow as possible, to allow for diverse analyses and use cases.

### Elementary Analysis

One of the most common uses of lat-long data is checking whether a point lies in some defined region. Weed allows for regions to be defined either as a lat-long box, or as a shapefile. For increased modularity, the shapefile may be defined as either a shape object or even as a string containing the file name.

#### Lat-Long Box

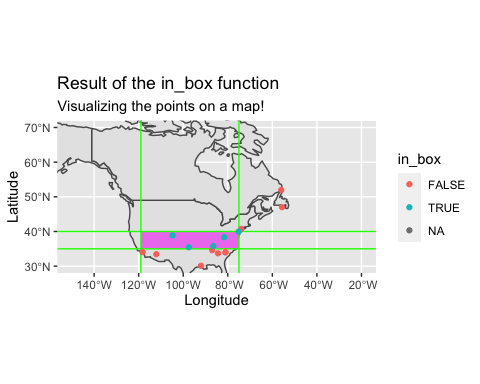
Assume the required box is given as follows

tllat = 40 # top left latitude  
tllng = -119 # top left longitude  
brlat = 35 # bottom right latitude  
brlng = -75 # bottom right longitude

We want to find out whether each of the disaster-location pairings in our dataset falls within this latitiude and longitude bounding box. This easy-to-use function performs the simple task, and can easily be piped with elementary visual analysis for verification and ease of understanding.

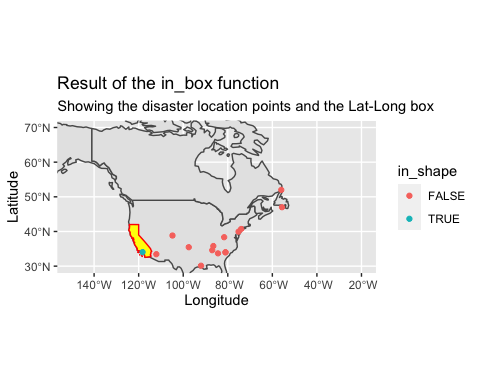
inbox\_df <- geocoded\_df %>%  
 located\_in\_box(top\_left\_lat = tllat, top\_left\_lng = tllng, bottom\_right\_lat = brlat, bottom\_right\_lng = brlng)

| Lat Long Box Data | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dis No | Country | Disaster Type | Latitude | Longitude | location\_word | uncertain\_location\_specificity | lat | lng | in\_box |
| 2000-0919-USA | United States of America (the) | Storm |  |  | alabama | FALSE | 34.60739 | -86.97977 | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | georgia | FALSE | 33.69277 | -84.39957 | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | louisiana | FALSE | 30.12595 | -92.00939 | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | north carolina | FALSE | 34.00071 | -81.03481 | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | south carolina | FALSE | 34.00071 | -81.03481 | FALSE |
| 2000-0919-USA | United States of America (the) | Storm |  |  | tennessee | FALSE | 35.80000 | -86.50000 | TRUE |



#### Shapefile

Likewise, we can check whether points lie in a given shapefile. Here we see which points lie in the region of California.



These functionalities should help with most common geocoding use cases and serve as a starting point for further analysis.

## Impact and Conclusion

The Wrangler for Emergency Events Database package can be used by researchers for analysis of disaster patterns in many regions. While it does not directly deal with analysis functions, the data cleaning and preparation tools offered by weed have proven useful in conducting further research. Geocoded disaster-location pairs are invaluable, and help connect the various locations affected by each calamity. Other efforts in this sphere of geocoding disaster data seem to be static, in that net new data is not geocoded, and seem to offer only one location for each disaster, even though common sense would suggest that many types of disasters affect a number of places in different ways. By changing the unit of analysis, providing geocoding and elemntary exploratory analysis tools, we hope that weed is eventually the first step of choice in emergency event analysis.

## Acknowledgements

## References