

Tutorial

Chunlei Zheng

2019-03-02

```
library(earthquake)
library(ggplot2)
library(tidyr)
library(dplyr)
```

Introduction

National Oceanographic and Atmospheric Administration (NOAA) collects significant earthquakes around the world and make a dataset that contains information about 5,933 earthquakes over an approximately 4,000 year time span. This dataset has a substantial amount of information embedded in it that may not be immediately accessible to people without knowledge of the intimate details of the dataset.

The overall goal of the R package is to work with this dataset and provide the tools for processing and visualizing the data so that others may extract some use out of the information embedded within.

Clean data

```
fname <- system.file("extdata", "signif.txt.tsv", package="earthquake")
eq_dat_raw <- as.data.frame(readr::read_tsv(fname))
eq_dat_clean <- earthquake::eq_clean_data(eq_dat_raw)
class(eq_dat_clean$DATE)
#> [1] "Date"
```

Timeline visualization

Given countries, we can use `geom_timeline` to visualize three types of information: the times at which earthquakes occur within certain countries, the magnitudes (i.e. Richter scale value) and the number of deaths associated with each earthquake. For example,

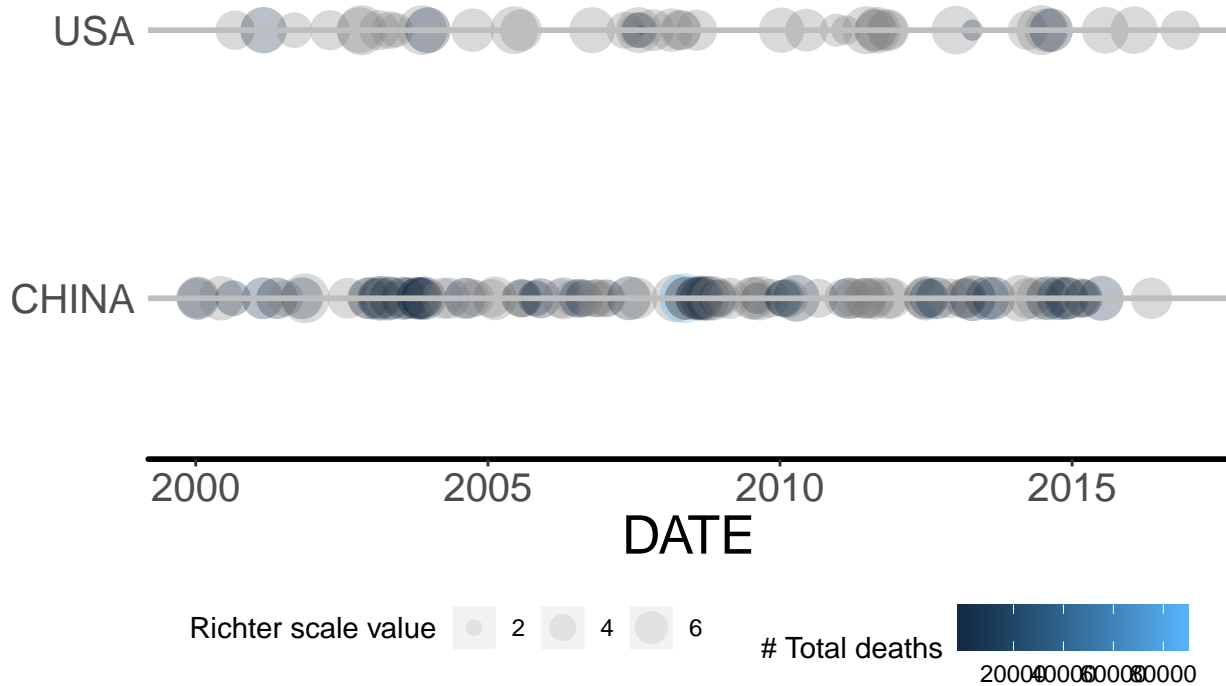
1. Choose the countries you want to look at

```
dat <- dplyr::select(eq_dat_clean, DATE, COUNTRY, LOCATION_NAME, LONGITUDE,
                    LATITUDE, EQ_MAG_MS, EQ_PRIMARY, TOTAL_DEATHS) %>%
  filter(COUNTRY %in% c("USA", "CHINA"))
```

2. Give time span parameters and draw a timeline

```
tl <- ggplot2::ggplot(data = dat, aes(x=DATE,
                                     size=as.numeric(EQ_PRIMARY),
                                     fill=as.numeric(TOTAL_DEATHS),
                                     y=COUNTRY)) +
  geom_timeline(xmin=as.Date("2000-01-01"), xmax=as.Date("2017-01-01"),
               stat="Timeline") +
  scale_fill_gradient(name="# Total deaths") +
  scale_size_continuous(name="Richter scale value") +
  theme_timeline
```

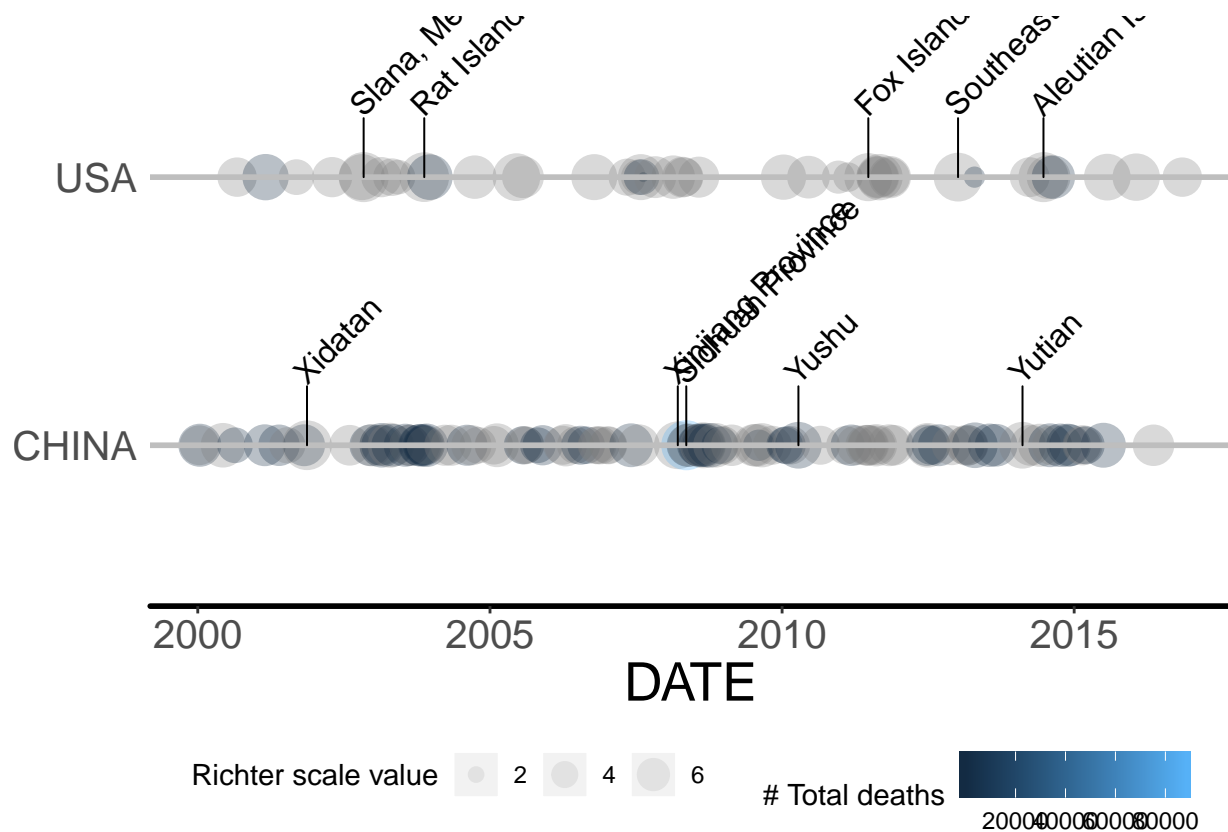
t1



3. If you want to know more specific locations for big earthquakes, `geom_timeline_label` can add this information. For example,

```
t1 <- ggplot2::ggplot(data = dat, aes(x=DATE,
                                     size=as.numeric(EQ_PRIMARY),
                                     fill=as.numeric(TOTAL_DEATHS),
                                     y=COUNTRY)) +
  geom_timeline(xmin=as.Date("2000-01-01"), xmax=as.Date("2017-01-01"),
               stat="Timeline") +
  geom_timeline_label(aes(label = LOCATION_NAME), n_max=5, xmin=as.Date("2000-01-01"),
                    xmax=as.Date("2017-01-01")) +
  scale_fill_gradient(name="# Total deaths") +
  scale_size_continuous(name="Richter scale value") +
  theme_timeline
```

t1



Interactive map visualization

In addition to building tools to visualize the earthquakes in time, it's important that we can visualize them in space too. In this module we will build some tools for mapping the earthquake epicenters and providing some annotations with the mapped data.

`eq_map()` function takes an argument `data` containing the filtered data frame with earthquakes to visualize. The function maps the epicenters (LATITUDE/LONGITUDE) and annotates each point with in pop up window containing annotation data stored in a column of the data frame. We can choose which column is used for the annotation in the pop-up with a function argument named `annot_col`. Each earthquake was shown with a circle, and the radius of the circle is proportional to the earthquake's magnitude (`EQ_PRIMARY`). For example,

```
eq_dat_clean %>%
  dplyr::filter(COUNTRY == "MEXICO" & lubridate::year(DATE) >= 2000) %>%
  eq_map(annot_col = "DATE")
```

In addition, it would be useful to have more interesting pop-ups for the interactive map created with the `eq_map()` function. `eq_create_label()` that takes the dataset as an argument and creates an HTML label that provides three types of information for each earthquake, including the location, the magnitude (`EQ_PRIMARY`), and the total number of deaths (`TOTAL_DEATHS`). For example,

```
eq_dat_clean %>%
  dplyr::filter(COUNTRY == "CHINA" & lubridate::year(DATE) >= 2000) %>%
  dplyr::mutate(popup_text = eq_create_label(.)) %>%
  eq_map(annot_col = "popup_text")
```