MSDR Package Documentation

AH Uyekita

2019-02-21

Contents

Prerequisites **5**

1 eq_clean_data **7**

1.1 Loading the data 7

1.2 Creating new features 8

1.3 Conversion Process 8

1.4 Cleaning Process 9

1.5 Example 1 9

1.6 Example 2 9

2 eq_location_clean **11**

2.1 Introduction 11

2.2 Example 11

3 geom_timeline **15**

3.1 Example 1 15

3.2 Example 2 16

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.name/tinytex/.

Chapter 1

eq_clean_data

This function has two behaviours:

1) When you assign a file to load, and;

```
# Loading the 'signif.txt' file.
eq_clean_data(file_name = system.file("extdata", "signif.txt", package = "msdr"))
```

2) When you pipe a dataset already loaded.

```
# Pipe.
readr::read_delim("signif.txt",
                  delim = "\t") %>% eq_clean_data()
```

1.1 Loading the data

This function also loads the Earthquake database from NOAA.

I_D

YEAR

LOCATION_NAME

EQ_PRIMARY

TOTAL_DEATHS

1

-2150

JORDAN: BAB-A-DARAA,AL-KARAK

7.3

NA

3

-2000

TURKMENISTAN: W

7.1

1

2

-2000

SYRIA: UGARIT

NA

NA

5877

-1610

GREECE: THERA ISLAND (SANTORINI)

NA

NA

8

-1566

ISRAEL: ARIHA (JERICHO)

NA

NA

11

-1450

ITALY: LACUS CIMINI

NA

NA

As you can see, there are several observations with NA values.

1.2 Creating new features

The `eq_clean_data` creates the DATE variable binding the columns YEAR, MONTH, and DAY. All this using the Lubridate package.

```
# Creating a new feature.
df <- df %>%
       mutate(DATE = lubridate::ymd(paste(df$YEAR,       # YEAR column
                                          df$MONTH,      # MONTH column
                                          df$DAY,        # DAY column
                                          sep = "/")))   # YYYY/MM/DD
```

1.3 Conversion Process

I have converted the class of some features:

- `TOTAL_DEATHS` to numeric;
- `EQ_PRIMARY` to numeric;
- All `NA`'s of `TOTAL_DEATHS` in zeros.

1.4 Cleaning Process

I have removed:

- All observations flagged as Tsunami, and;
- All observations with no Date.

1.5 Example 1

How to load a `txt` file.

```
# Load the package
library(msdr)

# Define as file_name the txt file.
df <- eq_clean_data(file_name = raw_data_path)

# Dimensions of the loaded dataframe.
dim(df)
#> [1] 2840   49
```

1.6 Example 2

Piping a dataset to the `eq_clean_data`.

```
# Load the package
library(msdr)

# Piping a read_delim with eq_clean_data.
readr::read_delim(raw_data_path,
                  delim = "\t") %>%

              eq_clean_data() -> df

# Dimensions of the loaded dataframe.
dim(df)
#> [1] 2840   49
```

Chapter 2

eq_location_clean

2.1 Introduction

This function creates a new column with the earthquake `LOCATION`. The function `eq_clean_data` uses it behind the scenes, so it is not necessary to call this function after call `eq_clean_data`.

2.2 Example

Piping a raw data to creates a LOCATION column.

```
# Path to the raw data.
raw_data_path <- system.file("extdata", "signif.txt", package = "msdr")

# Loading the dataset of Earthquake.
df <- readr::read_delim(file = raw_data_path,
                        delim = '\t',
                        col_names = TRUE,
                        progress = FALSE,
                        col_types = readr::cols())

# Printing some columns.
df %>%
       eq_location_clean() %>%
              # Selecting some features.
              select(YEAR,
                     COUNTRY,
                     LOCATION,
                     EQ_PRIMARY,
                     TOTAL_DEATHS) %>%
                     # Filtering.
                     filter(YEAR > 1990 &
                            YEAR < 2019) %>%
                            # Show the first 10 rows.
                            head(10) %>%
                                   # Enhance table visualization.
                                   kable()
```

YEAR

COUNTRY

LOCATION

EQ_PRIMARY

TOTAL_DEATHS

1991

MYANMAR (BURMA)

Thabeikkyin, Mandalay

7.1

NA

1991

AFGHANISTAN

Badakhstan, Baghlan, Laghman, Nagarhar

6.4

848

1991

SOLOMON ISLANDS

Solomon Islands

6.9

NA

1991

FRANCE

France

3.8

9

1991

RUSSIA

Kuril Islands

5.7

NA

1991

BERING SEA

Bering Sea

6.7

NA

1991

CHINA

Kalpin

6.1

NA

1991

CHINA

Ne, Datong

5.5

NA

1991

PERU

Rioja, Neuva Cajamarca

6.4

NA

1991

PERU

Rioja, Moyobamba, Nueva Cajamarca

6.7

53

As you can see, the `LOCATION` column has only cities in Title Case mode.

Chapter 3

geom_timeline

The `geom_timeline` is a new `geom_*` of `ggplot2` package that aims to enhance the visualization of earthquake. This Geom has some configuration:

- size: The earthquakes as displayed as circles with different radius (according to the `EQ_PRIMARY`);
- color: This is based on the `TOTAL_DEATHS`;
- x axis: This is the temporal axis.
- y axis: Each county has your own line, it is not possible to mix countries in a single y axis.

3.1 Example 1

Let's plot the earthquake from 1000 to 2000, which occured in `JAPAN`.

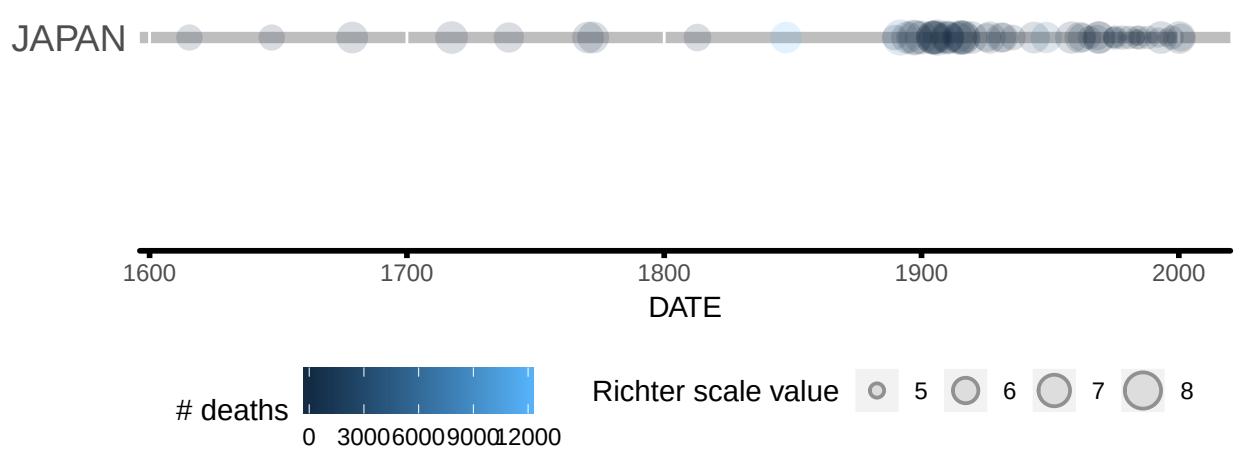
```
# Path to the raw data.
raw_data_path <- system.file("extdata", "signif.txt", package = "msdr")

# Loading the dataset of Earthquake.
df <- readr::read_delim(file = raw_data_path,
                        delim = '\t',
                        col_names = TRUE,
                        progress = FALSE,
                        col_types = readr::cols())

# Cleaning the data and filtering.
df %>%
       eq_clean_data() %>%

              filter(COUNTRY %in% 'JAPAN',
                     YEAR >= 1000 &
                     YEAR <= 2000) %>%
              # Creating a ggplot object
              ggplot() +
                     # Using the new Geom
                     geom_timeline(aes(x     = DATE,
                                       y     = COUNTRY,
                                       size  = EQ_PRIMARY,
                                       color = TOTAL_DEATHS)) +
```

```
                        # Adding theme.
                        theme_msdr() +
                               # Editing the legends' titles
                               labs(size = "Richter scale value",
                                    color = "# deaths")
```



Most of earthquake records in Japan are concentrated between 1900 and 2000.

3.2 Example 2

The earthquake record of 2018. Simple comparison.

```
# List of countries in Europe and "West Asia". This is not an exhaustive list.
eurasia <- c('SPAIN','GREECE','TURKEY','PORTUGAL','RUSSIA','FRANCE','MACEDONIA','BULGARIA',
             'ALBANIA','GEORGIA','ITALY','SLOVENIA','UK','CYPRUS','UKRAINE','CROATIA','AUSTRIA')

# Cleaninig data and filtering.
df %>%
       eq_clean_data() %>%
              # Creating a new feature.
              mutate(CONTINENT = case_when(COUNTRY %in% eurasia ~ "EURASIA",
                                           !(COUNTRY %in% eurasia) ~ "WORLD")) %>%
                     # Filtering.
                     filter(YEAR >= 2018 &
                            YEAR <= 2019) %>%

                            # Creating a ggplot object
                            ggplot() +
                                   # Using the new Geom
                                   geom_timeline(aes(x      = DATE,
```

```
                    y     = CONTINENT,
                    size  = EQ_PRIMARY,
                    color = TOTAL_DEATHS)) +
  # Adding theme.
  theme_msdr() +
        # Editing the legends' titles
        labs(size = "Richter scale value",
             color = "# deaths")
```