

Mapping ex report

Jiaheng Li

2020/11/11

1 Project Description

In this project, I am using the OpenFEMA Dataset: Public Assistance Funded Projects from the FEMA website. There are total 22 variables in the original dataset. There are their descriptions.

disasterNumber: Sequentially assigned number used to designate an event or incident declared as a disaster. For more information on the disaster process visit <https://www.fema.gov> pwNumber: Sequentially assigned number used to identify a unique project applicationTitle: Standard, non-unique (free form text) application title applicantId: Unique Public Assistance applicant identification number county: The name of a U.S. county, parish, borough, independent city or other political subdivision of a U.S. state or territory countyCode: Unique identifier for a U.S. county; typically, this identifier makes up the last three digits of a five digit Federal Information Processing Standard (FIPS) Code stateNumberCode: Unique identifier for a U.S. state or territory; this identifier makes up the first two digits of a five digit Federal Information Processing Standard (FIPS) Code state: The name of a U.S. state or territory projectAmount: The estimated total cost of the Public Assistance grant project in dollars, without administrative costs. This amount is based on the damage survey stateCode: Two-character unique identifier for a U.S. state or territory totalObligated: The federal share of the Public Assistance grant eligible project amount in dollars, plus grantee (State) and sub-grantee (applicant) administrative costs. The federal share is typically 75% of the total cost of the project federalShareObligated: The Public Assistance grant funding available to the grantee (State) in dollars, for sub-grantee's approved Project Worksheets damageCategory: The category code of the damage location dcc: Damage category code lastRefresh: Date the record was last updated in the API data store id: Unique ID assigned to the record hash: MD5 Hash of the fields and values of the record projectSize: Projects are designated as Large or Small, which can affect a number of variables in case management (processing). Project size is determined by the eligible amount as set in the damage survey damageCategoryCode: Code representing the type of work eligible for reimbursement through a Public Assistance grant; for a full listing of each category please go to /government/grant/pa/re_categories.shtm declarationDate: Date the disaster was declared incidentType: Type of incident such as fire or flood. The incident type will affect the types of assistance available. For more information on incident types, please visit: <https://www.fema.gov> obligatedDate: Date the grant was obligated

My purpose on this project is to use the data I got to make a mapping tool of showing the total Obligated fund for the hurricane disaster in each state on the date when the disaster was declared.

2 Data Cleaning

```
# data <- read_excel("C:/Users/superavis/Documents/TEMPUSE/063-superzip-example/hurricane.xlsx")
# data2 <- data %>% select(state,stateCode,totalObligated,obligatedDate)
# data2$date <- as.Date(substr(data2$obligatedDate,1,10))
#
# data3 <- data2 %>% group_by(date, state) %>% summarise(total = sum(totalObligated,na.rm=T))
# save(data3, file = "hurricane.rdata")
```

I firstly use the excel filter to filter out the hurricane data in the original data because the size of the original

data was too large for R Studio to process. Then I convert the original date to the date variable of the R studio.

3 Using template on Shiny website

I found a teplate on the Shiny website which can fit in the project I planned to do of the mapping assignment. (<https://shiny.rstudio.com/gallery/superzip-example.html>)

3.1 Server

Then I switched out the varibales in the template and made a new interactive map with two parameters of 'date' and 'state' also set the output as the sum of totalObligated on that day. The codes are as following.

(<https://shiny.rstudio.com/gallery/superzip-example.html>)

```
library(geojsonio)
library(leaflet)
library(shiny)
library(RColorBrewer)
library(scales)
library(lattice)
library(dplyr)
library(lubridate)

states <-
  geojson_read(
    x = "https://raw.githubusercontent.com/PublicaMundi/MappingAPI/master/data/geojson/us-states.json"
    , what = "sp"
  )

shinyServer(
function(input, output, session) {

  labels <- reactive( {

    states2 <- states
    hurr2 <- hurr[hurr$date == input$date,]
    states2$hurricane <- hurr2$total[match(states$name, hurr2$state)]
    sprintf(
      "<strong>%s</strong><br/>%g sum of totalObligated",
      states2$name, states2$hurricane
    ) %>% lapply(htmltools::HTML) })

  statesdata <- reactive( {
    data.frame( hurr[hurr$date == input$date & hurr$state == input$state,3])
  })

  statesd <- reactive(
  {

    states2 <- states
    hurr2 <- hurr[hurr$date == input$date,]
```

```

states2$hurrican <- hurr2$total[match(states$name, hurr2$state)]
states2
})

output$map <- renderLeaflet( {
# states2 <- statesd()
states2 <- states
states2$hurrican <- hurr2$total[match(states2$name, hurr2$state)]
bins <- c(0, 10000, 50000, 1000000, Inf)

pal <- colorBin("YlOrRd", domain = states2$hurrican, bins = bins)

labels <-
  sprintf(
    "<strong>%s</strong><br/>%g sum of totalObligated",
    states2$name, states2$hurrican
  ) %>% lapply(htmltools::HTML)
leaflet( states2) %>%
  setView(-96, 37.8, 4) %>%
  addProviderTiles("MapBox", options = providerTileOptions(
    id = "mapbox.light",
    accessToken = Sys.getenv('MAPBOX_ACCESS_TOKEN')) ) %>%
  addPolygons(
    fillColor = ~pal(hurrican),
    weight = 2,
    opacity = 1,
    color = "white",
    dashArray = "3",
    fillOpacity = 0.7,
    highlight = highlightOptions(
      weight = 5,
      color = "#666",
      dashArray = "",
      fillOpacity = 0.7,
      bringToFront = TRUE),
    label = labels,
    labelOptions = labelOptions(
      style = list("font-weight" = "normal", padding = "3px 8px"),
      textsize = "15px",
      direction = "auto")) %>%
  addLegend(pal = pal, values = ~hurrican, opacity = 0.7, title = NULL,
    position = "bottomright")

})

output$tbl <- renderTable({
  statesdata()
})

}
)

```

3.2 UI

The UI file is used for designing the layout of the mapping tool.

```
load("C:/Users/aaron/OneDrive/Desktop/mapping/Mapping ex/hurricane.rdata")
hurr <- data3

# Choices for drop-downs
vars1 <- as.character(hurr$date)
vars2 <- as.character(unique(hurr$state))

shinyUI(
  navbarPage("Interactive map", id="nav",

    tabPanel("U.S states hurrican distribution shinyapp",

      sidebarLayout(
        sidebarPanel(
          h2("Hurrican distribution"),

          selectInput("date", "Date", vars1, selected = vars1[1]),
          selectInput("state", "State", vars2, selected = vars2[1]),
          h3("sum of total totalObligated"),
          tableOutput("tbl")),
        mainPanel(

          leafletOutput("map")
        )
      )
    )
  )
)
```