615 report

Ruiyi Lian

November 11, 2020

Input&Intro Data

The data is about the disaster distribution in the United State.

```
r <- read_csv("C:/Users/LXD/Documents/R/615 mapping/report/r(1).csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   disasterNumber = col_double(),
##   declarationDate = col_datetime(format = ""),
##   pwNumber = col_double(),
##   countyCode = col_double(),
##   stateNumberCode = col_double(),
##   projectAmount = col_double(),
##   federalShareObligated = col_double(),
##   totalObligated = col_double(),
##   obligatedDate = col_datetime(format = ""),
##   lastRefresh = col_datetime(format = "")
## )
```

```
## See spec(...) for full column specifications.
```

```
head(r)
```

```
## # A tibble: 6 x 22
##   disasterNumber declarationDate     incidentType pwNumber applicati
onTitle
##            <dbl> <dttm>              <chr>           <dbl> <chr>
## 1           1239 1998-08-26 04:00:00 Severe Stor~       41 Not Provi
ded
## 2           1239 1998-08-26 04:00:00 Severe Stor~       51 Not Provi
ded
## 3           1239 1998-08-26 04:00:00 Severe Stor~       43 Not Provi
ded
## 4           1239 1998-08-26 04:00:00 Severe Stor~        2 (L)
## 5           1239 1998-08-26 04:00:00 Severe Stor~       47 Not Provi
ded
## 6           1239 1998-08-26 04:00:00 Severe Stor~       35 Not Provi
ded
```

```
## # ... with 17 more variables: applicantId <chr>, damageCategoryCode
<chr>,
## #   dcc <chr>, damageCategory <chr>, projectSize <chr>, county <chr>,
## #   countyCode <dbl>, state <chr>, stateCode <chr>, stateNumberCode
<dbl>,
## #   projectAmount <dbl>, federalShareObligated <dbl>, totalObligated
 <dbl>,
## #   obligatedDate <dttm>, hash <chr>, lastRefresh <dttm>, id <chr>
```

For more information of the data: ### Data Cleaning

First of all, we need to select the variable we need. They are disasterNumber which helps us identify which disaster we are looking, county and state which help us locate where these disaster happened, incident type which helps us know what kinds of disaster it was, and state number code and county code which help us draw the map.

```
data<-r%>%group_by(disasterNumber,county,state)%>%summarise(projectAmou
nt=sum(projectAmount),countyCode=max(countyCode),stateNumberCode=max(st
ateNumberCode),incidentType=max(incidentType),.groups = 'drop')
data1<-data%>%group_by(county)%>%summarise(projectAmount=sum(projectAmo
unt),countyCode=max(countyCode),stateNumberCode=max(stateNumberCode),in
cidentType=max(incidentType),state=max(state),.groups = 'drop')
head(data1)
```

```
## # A tibble: 6 x 6
##   county    projectAmount countyCode stateNumberCode incidentType
state
##   <chr>             <dbl>      <dbl>           <dbl> <chr>
<chr>
## 1 Abbeville       588681.          1              45 Severe Storm(~
South Carol~
## 2 Acadia        16758715.          1              22 Severe Storm(~
Louisiana
## 3 Accomack       2613022.          1              51 Severe Storm(~
Virginia
## 4 Ada            3948655.          1              16 Flood
Idaho
## 5 Adair         14558158.          1              40 Snow
Oklahoma
## 6 Adams         58364474.          3              55 Tornado
Wisconsin
```

MAPPing

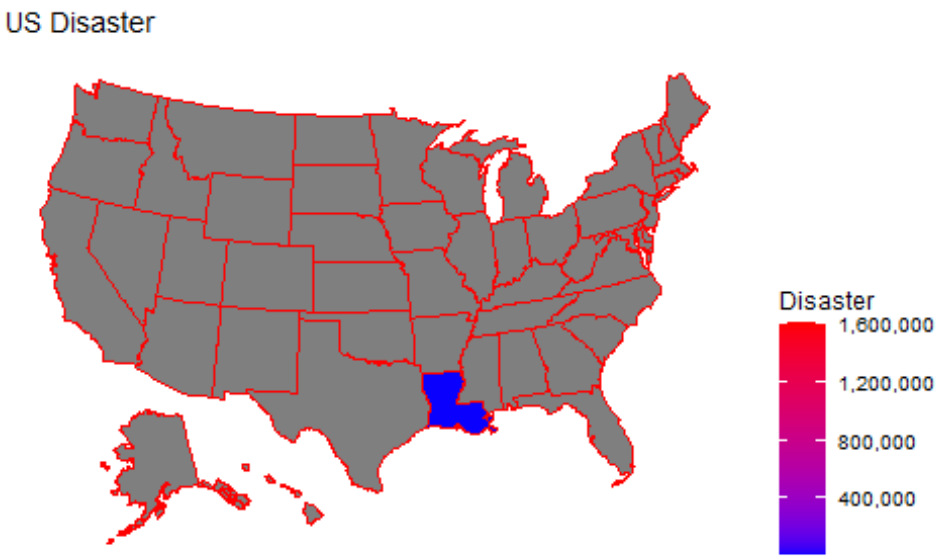
Here is examples for our data visualization on US Map.

Based on DisasterNumber

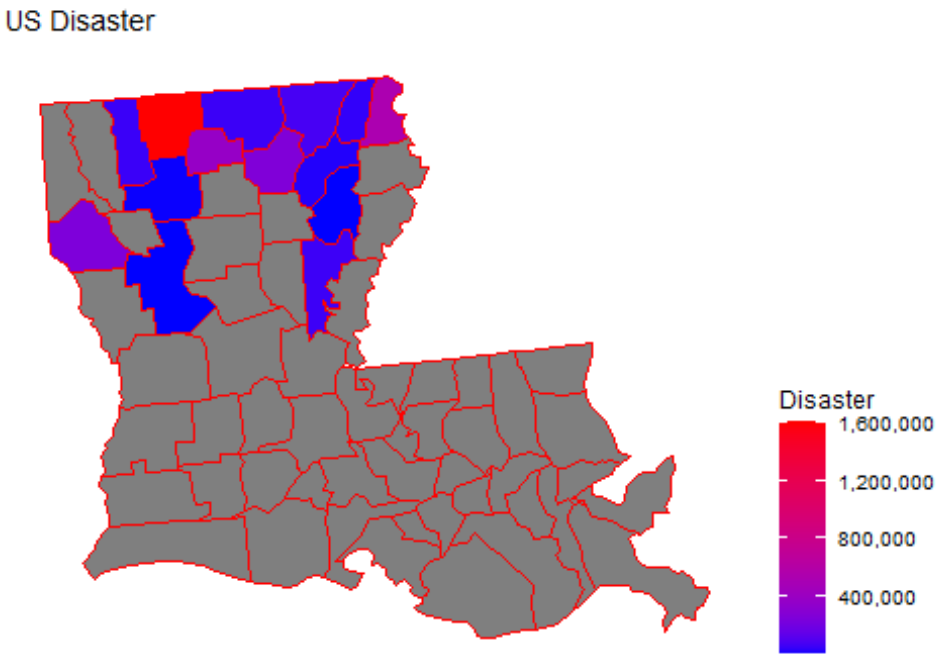
Let's take #1264 disaster for example.

```
data2<-data[data$disasterNumber == 1264,]

#based on us map
plot_usmap(data = data2, values = "projectAmount", color = "red") +
      scale_fill_continuous(
        low = "blue", high = "red", name = "Disaster", label = scales::
comma
      ) + labs(title = "US Disaster") + theme(legend.position = "right")
```




```
#based on State map
a<-data2$state
data2<-data2%>%filter(county != "Statewide")
    data2<-data2 %>% rowwise %>% mutate(fips = 1000*stateNumberCode+cou
ntyCode)
plot_usmap(regions = "county", data = data2, values = "projectAmount",i
nclude = a ,color = "red") +
      scale_fill_continuous(
        low = "blue", high = "red", name = "Disaster", label = scales::
comma
      ) + labs(title = "US Disaster") + theme(legend.position = "right")
```



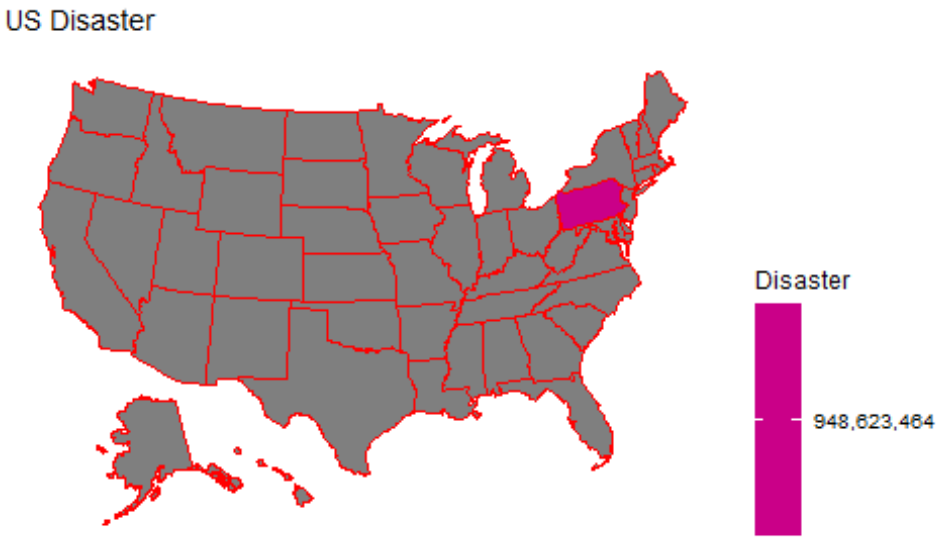
Based on State

Let's take PA as an example

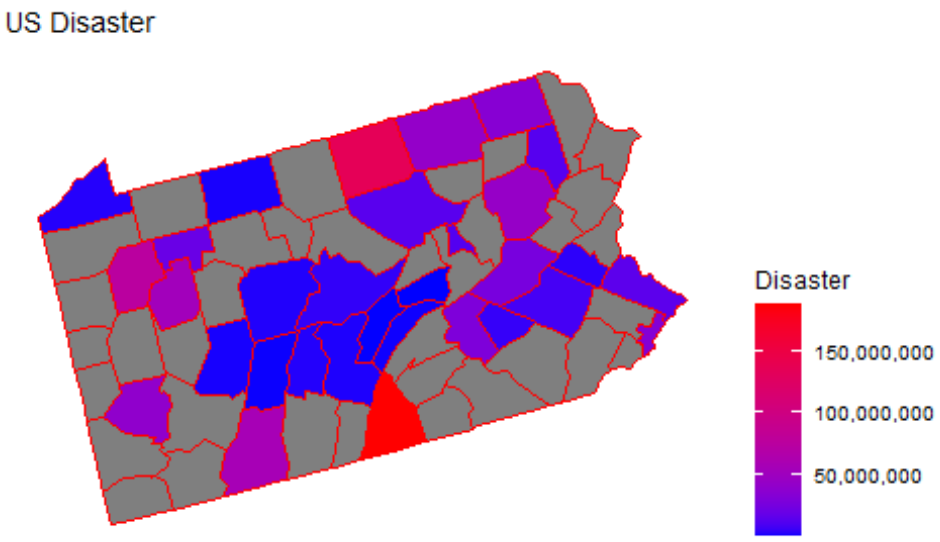
```
data3<-data1[data1$state == "Pennsylvania",]

#Based on the US MaP
data3_1<-data3%>%group_by(state)%>%summarise(projectAmount=sum(projectA
mount),countyCode=max(countyCode),stateNumberCode=max(stateNumberCode),
incidentType=max(incidentType),.groups = 'drop')

plot_usmap(data = data3_1, values = "projectAmount", color = "red") +
      scale_fill_continuous(
        low = "blue", high = "red", name = "Disaster", label = scales::
comma
      ) + labs(title = "US Disaster") + theme(legend.position = "right")
```




```
#Based on the state Map
a<-data3$state
data3_2<-data3%>%filter(county != "Statewide")
    data3_2<-data3_2 %>% rowwise %>% mutate(fips = 1000*stateNumberCode
+countyCode)
plot_usmap(regions = "county", data = data3_2, values = "projectAmount",
include = a ,color = "red") +
      scale_fill_continuous(
        low = "blue", high = "red", name = "Disaster", label = scales::
comma
      ) + labs(title = "US Disaster") + theme(legend.position = "right")
```

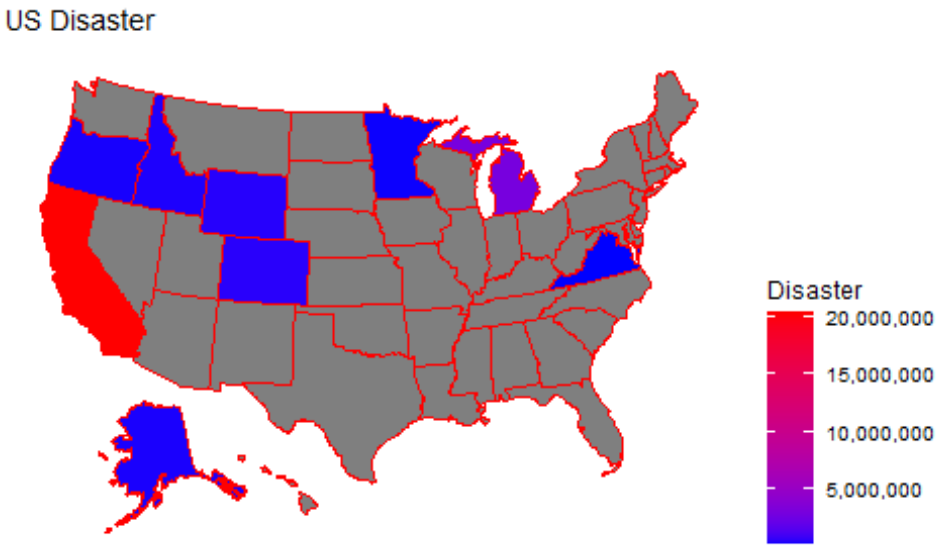


Based on Incident Type

Let's take Biological as an example

```
data4<-data1[data1$incidentType == "Biological",]

#Based on the US MaP
data4_1<-data4%>%group_by(state)%>%summarise(projectAmount=sum(projectA
mount),countyCode=max(countyCode),stateNumberCode=max(stateNumberCode),
incidentType=max(incidentType),.groups = 'drop')
plot_usmap(data = data4_1, values = "projectAmount", color = "red") +
      scale_fill_continuous(
        low = "blue", high = "red", name = "Disaster", label = scales::
comma
      ) + labs(title = "US Disaster") + theme(legend.position = "right")
```




```
#Based on the state Map
a<-data4$state
data4_2<-data4%>%filter(county != "Statewide")
    data4_2<-data4_2 %>% rowwise %>% mutate(fips = 1000*stateNumberCode
+countyCode)
plot_usmap(regions = "county", data = data4_2, values = "projectAmount",
include = a ,color = "red") +
      scale_fill_continuous(
        low = "blue", high = "red", name = "Disaster", label = scales::
comma
      ) + labs(title = "US Disaster") + theme(legend.position = "right")
```

US Disaster