Mapping With more Data - FEMA

Runqi Zhao, Danping Liu, Chenghao Meng

2020/11/11

Contents

Introduction

In this project, we are going to create our own interactive map to take a look at the public assistance program summary data from FEMA.

1 Data Cleaning

First, we will load the geolocation data of states and counties in United States from `maps` package. Meanwhile, we will also load the processed data based on FEMA database.

1.1 Load Data

We downloaded FEMA data from this link: https://www.fema.gov/openfema-data-page/public-assistance-funded-projects-details-v1

This .csv file include detailed public assistance funded projects information.

We selected the the Sever Storm(s), Hurricane and Coastal Storm incident-types in 2009-2018 for this analysis.

This dataset is prepared for county-level plot.

```
# Load the data of US geolocation
us_county <- map_data("county")
us_state <- map_data("state")
```

```r
# Load the data downloaded from FEMA
fema_dt <- read.csv(file="hrc_county.csv",header = T)
```

Here are some explanations of the key columns in the following analysis from FEMA website:

- `projectAmount`("Project Amount"): The estimated total cost of the Public Assistance grant project in dollars, without administrative costs. This amount is based on the damage survey.
- `federalShareObligated`("Federal Share Obligated"): The Public Assistance grant funding available to the grantee (State) in dollars, for sub-grantee's approved Project Worksheets.
- `totalObligated`("Total Obligated"): 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.

Since we would like to analyze the amount of damage brought by hurricane and other storm, we will drop columns containing irrelevent informations.

```r
# Drop the columns
hurri_storm_pre <- fema_dt %>%  select(-c("applicationTitle","damageCategoryCode","projectSize","stateNu
```

```r
# Change the capital letter to lower
hurri_storm_pre$state <- tolower(hurri_storm_pre$state)
hurri_storm_pre$county <- tolower(hurri_storm_pre$county)
```

1.2 Further Cleaning

Then, we will do further cleaning to get the total amount of the damage information in each county and state as a whole.

```r
hurri_storm_pre2 <-
hurri_storm_pre %>%
  group_by(Fips,incidentType) %>%
  summarise(state = unique(state),
            year = unique(declarationYear),
            county = unique(county),
            TotalAmount = sum(totalObligated)/1000,
            TotalProject = sum(projectAmount)/1000,
            TotalFederal =
              sum(federalShareObligated)/1000
            )
hurri_storm_pre2$Fips <- as.character(hurri_storm_pre2$Fips)
```

To get the fips of each county, we should load the dataset from `maps` package. Moreover, we will also merge the fips data with county data to get the longitude and latitude of each county.

```r
# Make adjustments on fips
county_fips <- county.fips
county_fips$fips <- as.character(county_fips$fips)

county_fips <- county_fips %>% separate(polyname, into= c("state","county"),sep= ",") %>% rename(Fips=fi
```

```r
# Merge the us_county with fips data
us_county <- us_county %>% rename(state=region,county=subregion)

county_info <- right_join(county_fips,us_county,by=c("state","county"))
```

We will also merge the county_info with hurri_storm_pre2 to get a full data frame for mapping.

```r
hurri_storm <- inner_join(county_info,hurri_storm_pre2,by=c("Fips","state","county"))
```

2 Explortary Data Analysis

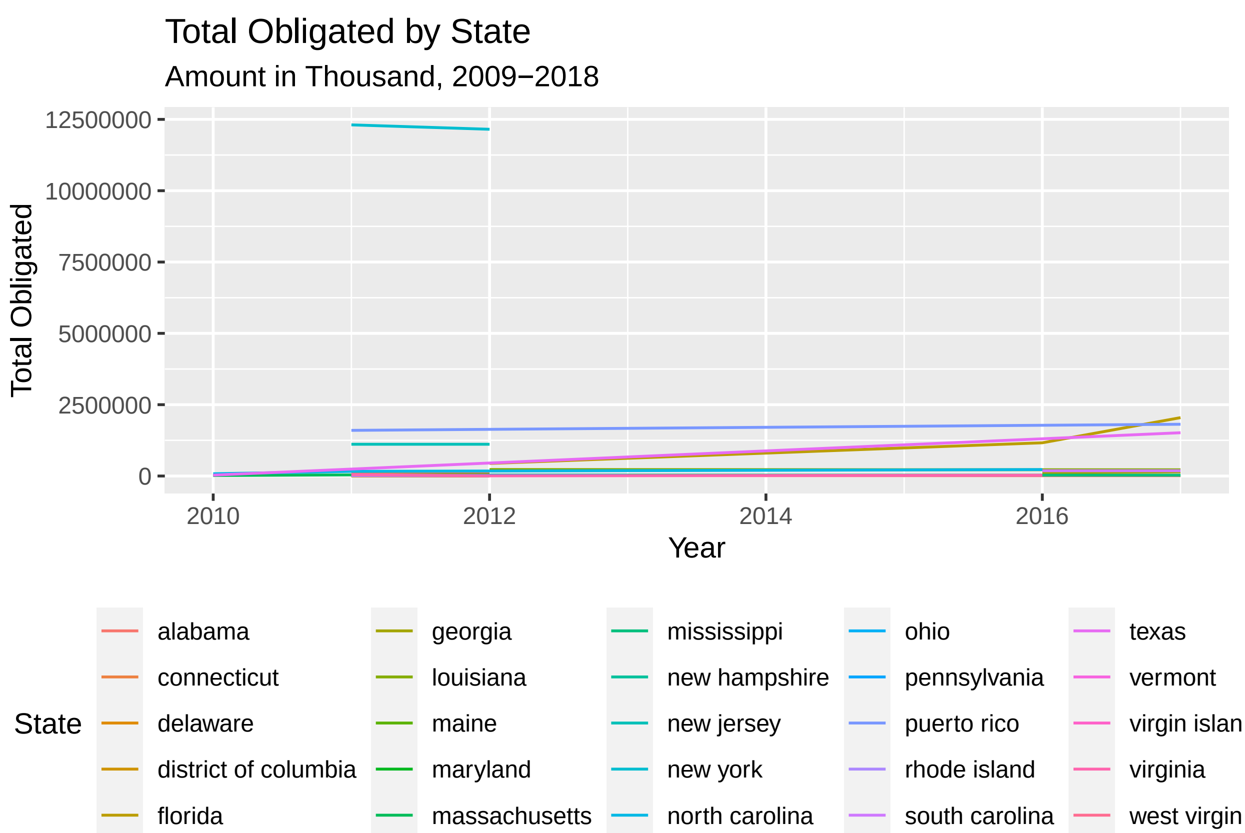
2.1 Damage Data by State

2.1.1 Total Obligated by State

First, we would like to explore the total obligated caused by all three kinds of disasters of interest by state by using line chart.

```r
hurri_storm_state <- hurri_storm_pre2 %>%
  filter(incidentType=="Hurricane") %>%
  group_by(state,year) %>%
  summarise(TotalAmount_state=sum(TotalAmount),
            TotalProject_state=sum(TotalProject),
            TotalFederal_state=sum(TotalFederal))
```

```r
# Draw the line chart-Total Obligated
ggplot(data=hurri_storm_state) + geom_line(aes(x=year,y=TotalAmount_state,col=state))+
  xlab("Year") + ylab("Total Obligated") +
  labs(title = "Total Obligated by State",subtitle = "Amount in Thousand, 2009-2018",col="State")+
   theme(legend.position = "bottom")
```

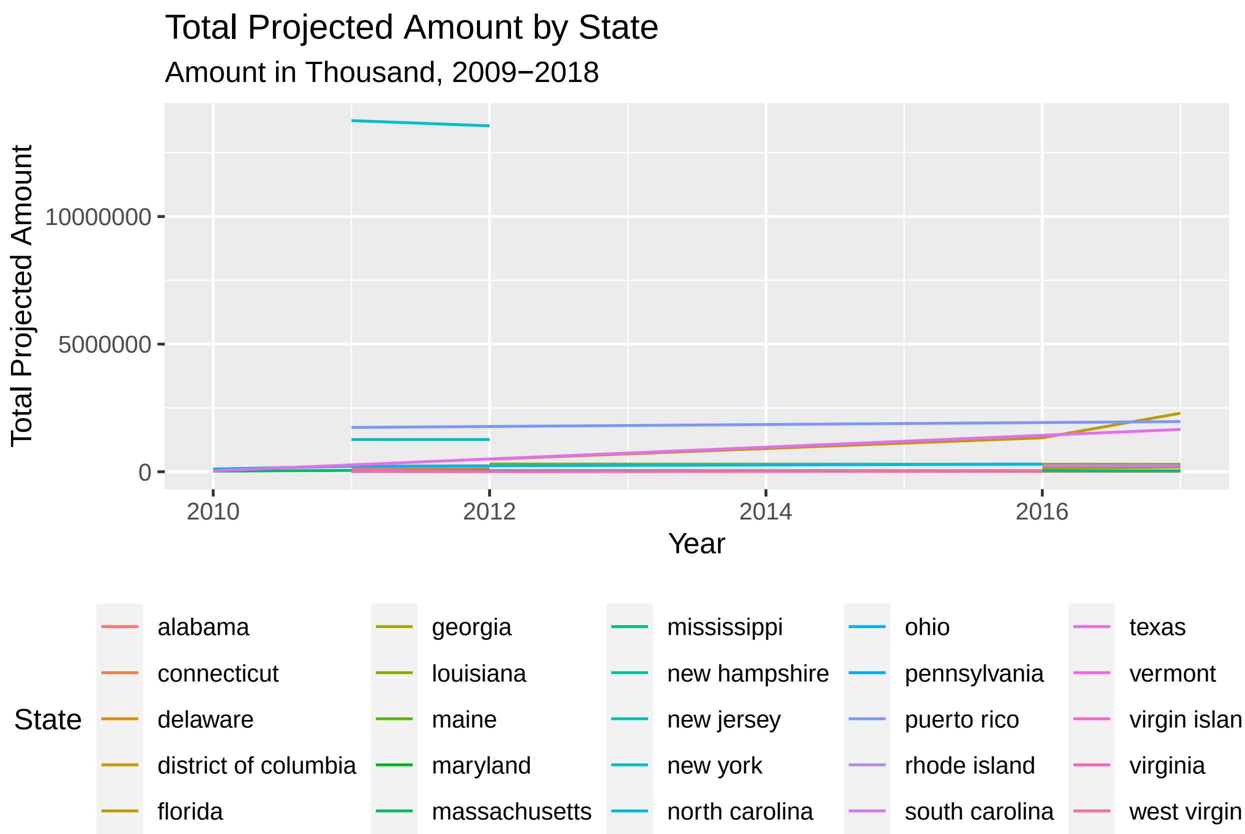


The line chart indicates that New York has the highest amount of **Total Obligated** in the year of 2011 and 2012.

2.1.2 Projected Amount by State

```
# Do not use scientific notation
options(scipen=200)

# Draw the line chart-Projected Amount
ggplot(data=hurri_storm_state) + geom_line(aes(x=year,y=TotalProject_state,col=state))+
  xlab("Year") + ylab("Total Projected Amount") +
  labs(title = "Total Projected Amount by State",subtitle = "Amount in Thousand, 2009-2018",col="State")
   theme(legend.position = "bottom")
```

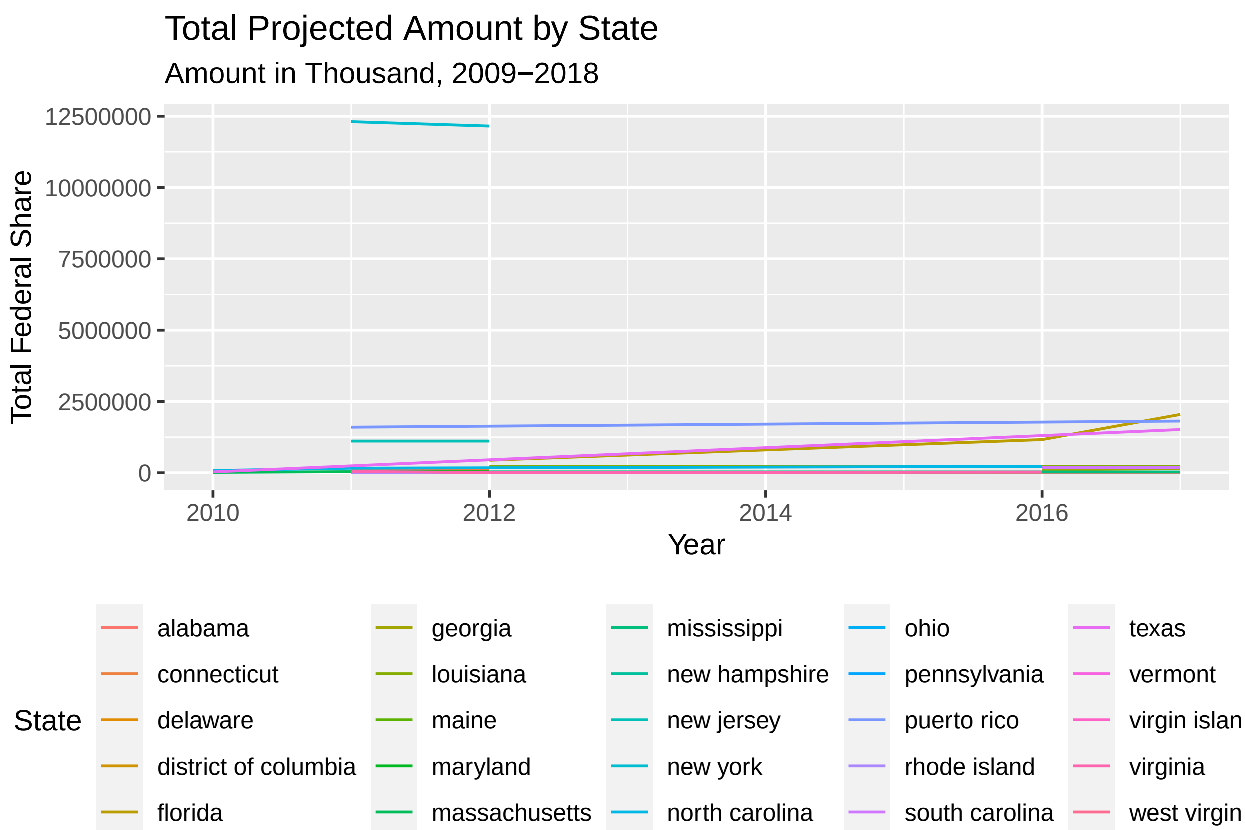


The line chart also indicates that the state of New York has the highest **Projected Amount** in the year of 2011.

2.1.3 Federal Share by State

Then we will look at Federal Share by state by using line chart.

```
# Draw the line chart-Projected Amount
ggplot(data=hurri_storm_state) + geom_line(aes(x=year,y=TotalFederal_state,col=state))+
  xlab("Year") + ylab("Total Federal Share") +
  labs(title = "Total Projected Amount by State",subtitle = "Amount in Thousand, 2009-2018",col="State")
   theme(legend.position = "bottom")
```

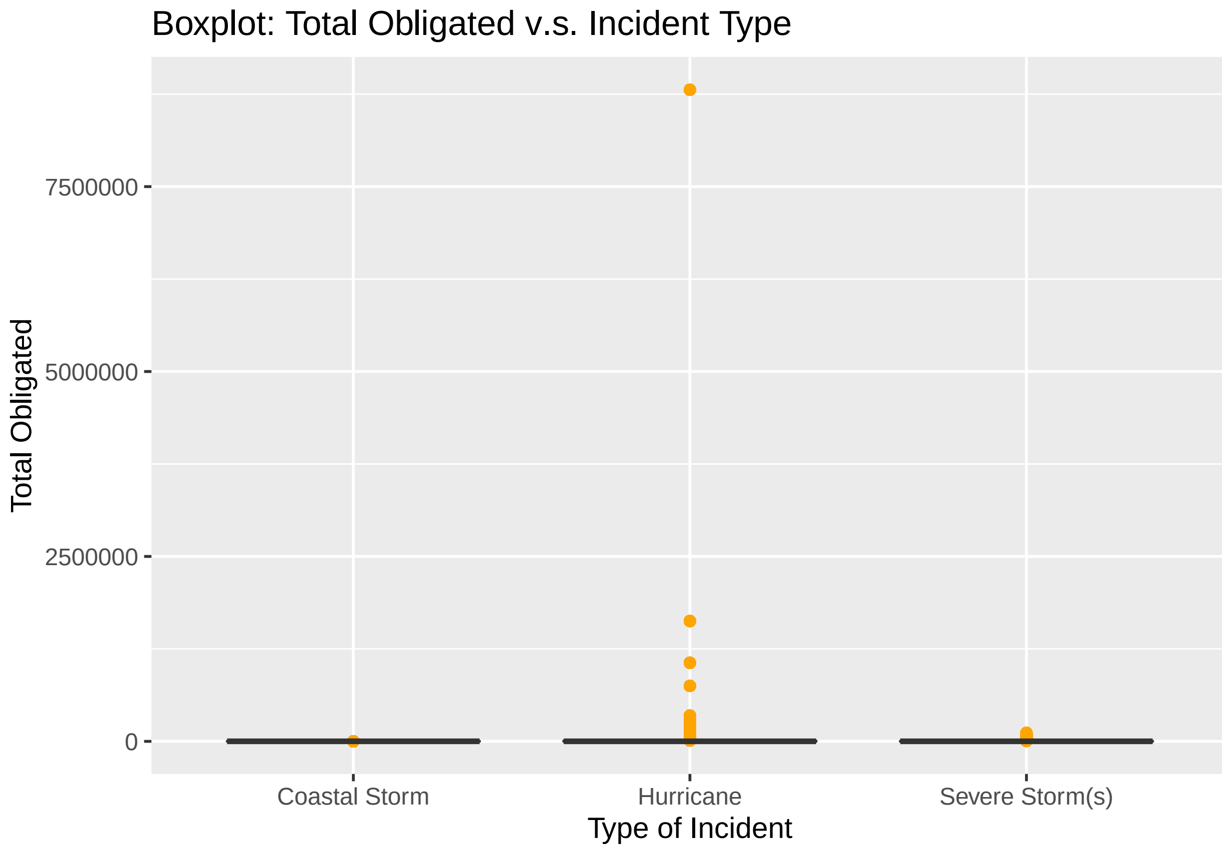


The line chart indicates that the state of New York has the highest **Federal Share** in the year of 2011 again.

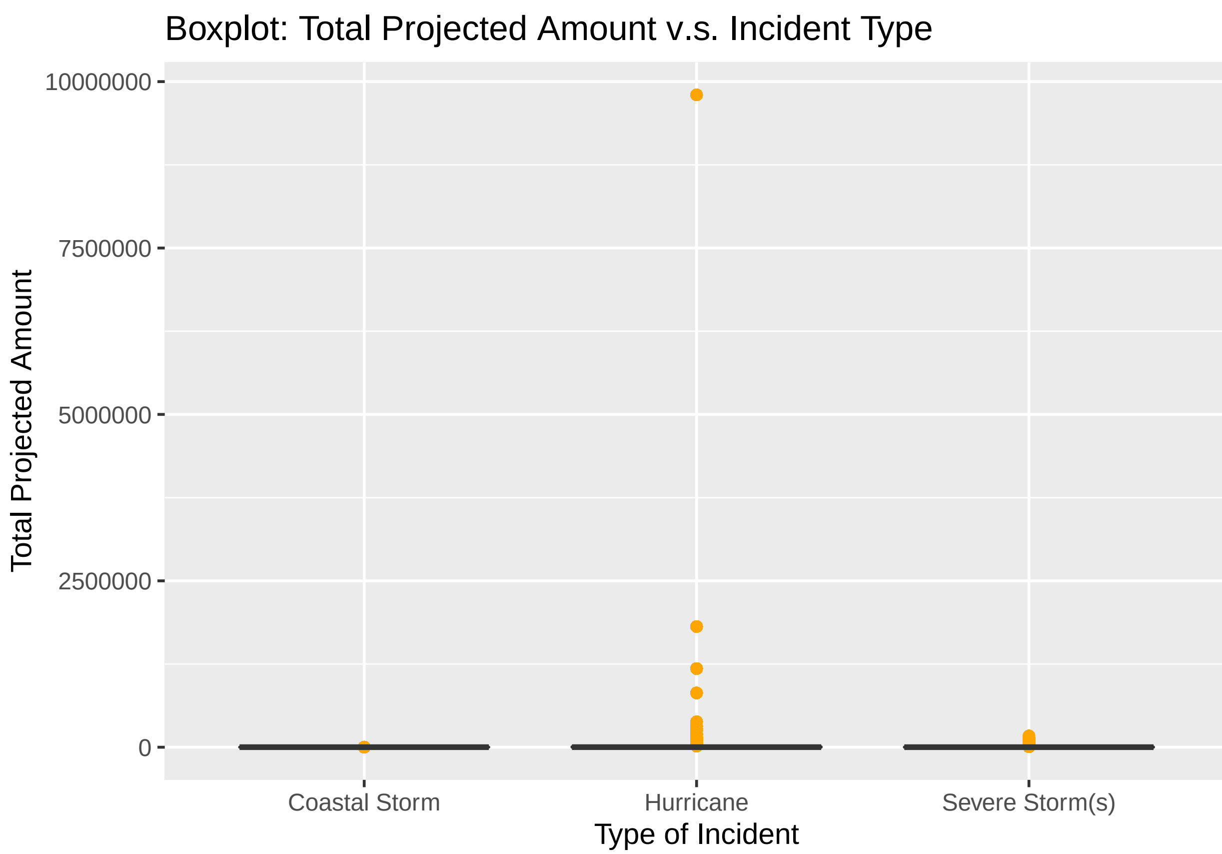
2.2 Outlier Exploration

From the line charts above, we can see that the damage data in the state of New York seems to be an outlier. In order to conduct further analysis of the outlier, we will draw a box plot based on three types of damage data: **Total Obligated**,**Projected Amount** and **Federal Share**.

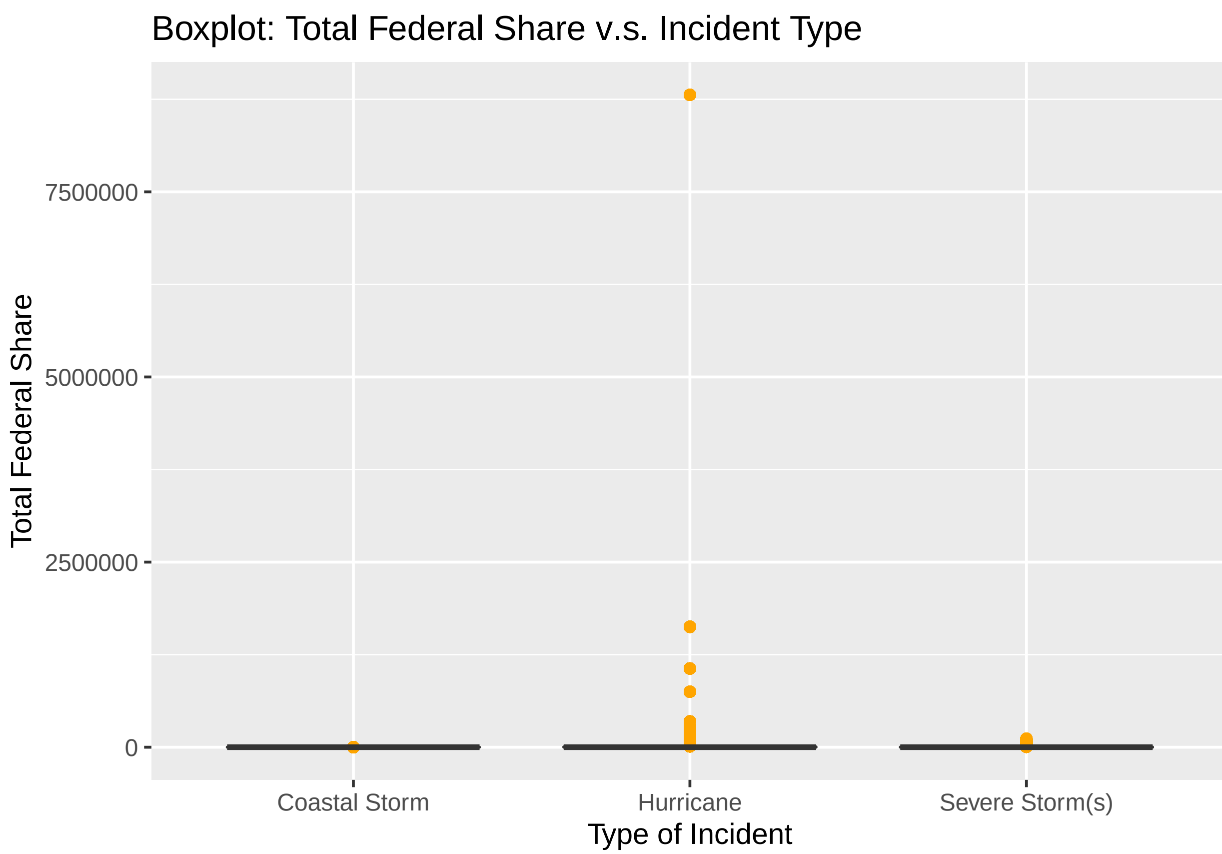
```r
# Boxplot to identify the outliers in TotalAmount
ggplot(data=hurri_storm)+
  geom_boxplot(aes(x=incidentType,y=TotalAmount),outlier.colour="orange")+
  xlab("Type of Incident") + ylab("Total Obligated")+
  ggtitle("Boxplot: Total Obligated v.s. Incident Type")
```




```
# Boxplot to identify the outliers in TotalProject
ggplot(data=hurri_storm)+
  geom_boxplot(aes(x=incidentType,y=TotalProject),outlier.colour="orange")+
  xlab("Type of Incident") + ylab("Total Projected Amount")+
  ggtitle("Boxplot: Total Projected Amount v.s. Incident Type")
```




```r
# Boxplot to identify the outliers in TotalProject
ggplot(data=hurri_storm)+
  geom_boxplot(aes(x=incidentType,y=TotalFederal),outlier.colour="orange")+
  xlab("Type of Incident") + ylab("Total Federal Share")+
  ggtitle("Boxplot: Total Federal Share v.s. Incident Type")
```



The three box plots above indicate that the damage data of all three tyoes of disasters has some outliers.

In order to make those maps convenient to review, we will move those outliers.

```
# Drop the outliers in TotalAmount
out_value_oblig <- boxplot.stats(hurri_storm$TotalAmount)$out
outliers_oblig <- hurri_storm %>% filter(TotalAmount %in% out_value_oblig)
hurri_storm_new <- hurri_storm %>% filter(!(TotalAmount %in% out_value_oblig))

# Drop the outliers in TotalProject
out_value_proj <- boxplot.stats(hurri_storm$TotalProject)$out
outliers_proj <- hurri_storm %>% filter(TotalProject %in% out_value_proj)
hurri_storm_new <- hurri_storm %>% filter(!(TotalProject %in% out_value_proj))

# Drop the outliers in TotalFederal
out_value_fed <- boxplot.stats(hurri_storm$TotalFederal)$out
outliers_fed <- hurri_storm %>% filter(TotalFederal %in% out_value_fed)
hurri_storm_new <- hurri_storm %>% filter(!(TotalFederal %in% out_value_fed))
```

3 Mapping the Damage

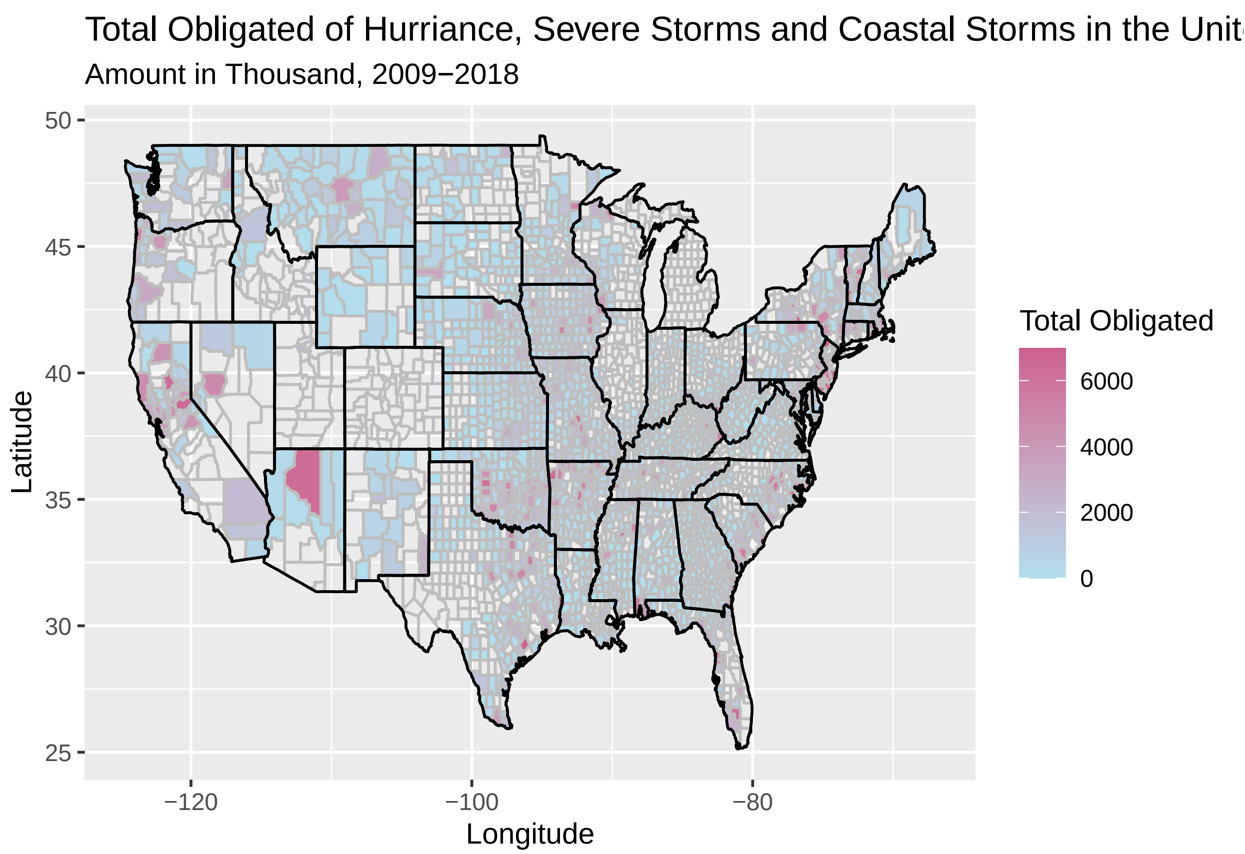
3.1 Total Obligated of Hurriance, Severe Storms and Coastal Storms

Before we take a look at the specific type of damage, we would like to take a overview of the damage data of all three types of damage.

```r
p <- ggplot()+
  geom_polygon(data=hurri_storm_new,
               aes(x=long,y=lat,group=group,fill=TotalAmount))+

  geom_path(data=county_info,mapping=aes(long,lat,group=group),color="grey")+
  geom_path(data=us_state,mapping=aes(long,lat,group=group))
```

```r
p + labs(title="Total Obligated of Hurriance, Severe Storms and Coastal Storms in the United States",
        subtitle="Amount in Thousand, 2009-2018",
        fill="Total Obligated" ) +
  xlab("Longitude") + ylab("Latitude") +
  scale_fill_continuous(low="lightblue2",high="hotpink3",space="Lab")
```



3.2 Damage Data of Hurricane

Then, we will take a look at the damage data of Hurricane. Since the code of mapping are very same, we will let `echo=FALSE` to hide those codes for the conciseness of the report.

In order to show the difference in the code of map for each type of disaster, we will keep the code in the first part of each following section.

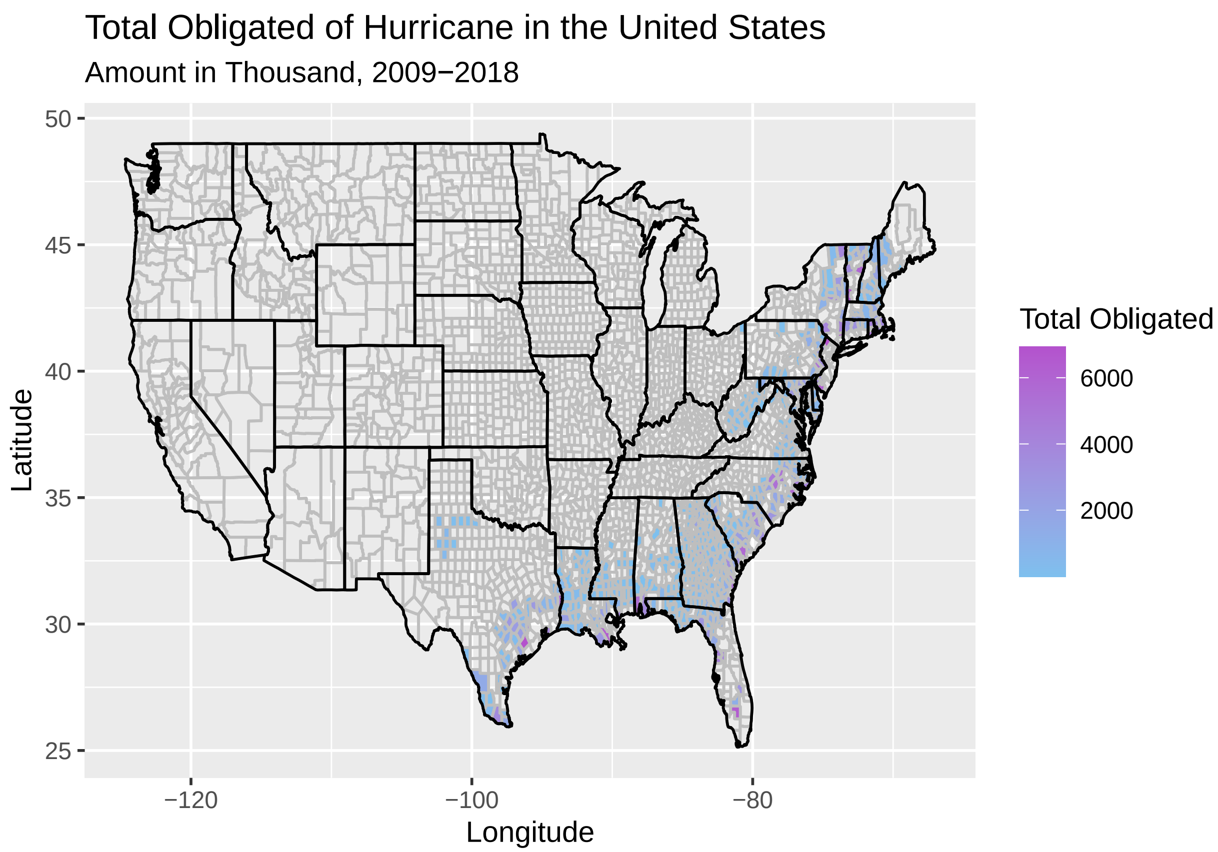
3.2.1 Hurricane: Total Obligated

```r
p1 <- ggplot()+
  geom_polygon(data=hurri_storm_new %>%
                 filter(incidentType=="Hurricane"),
```

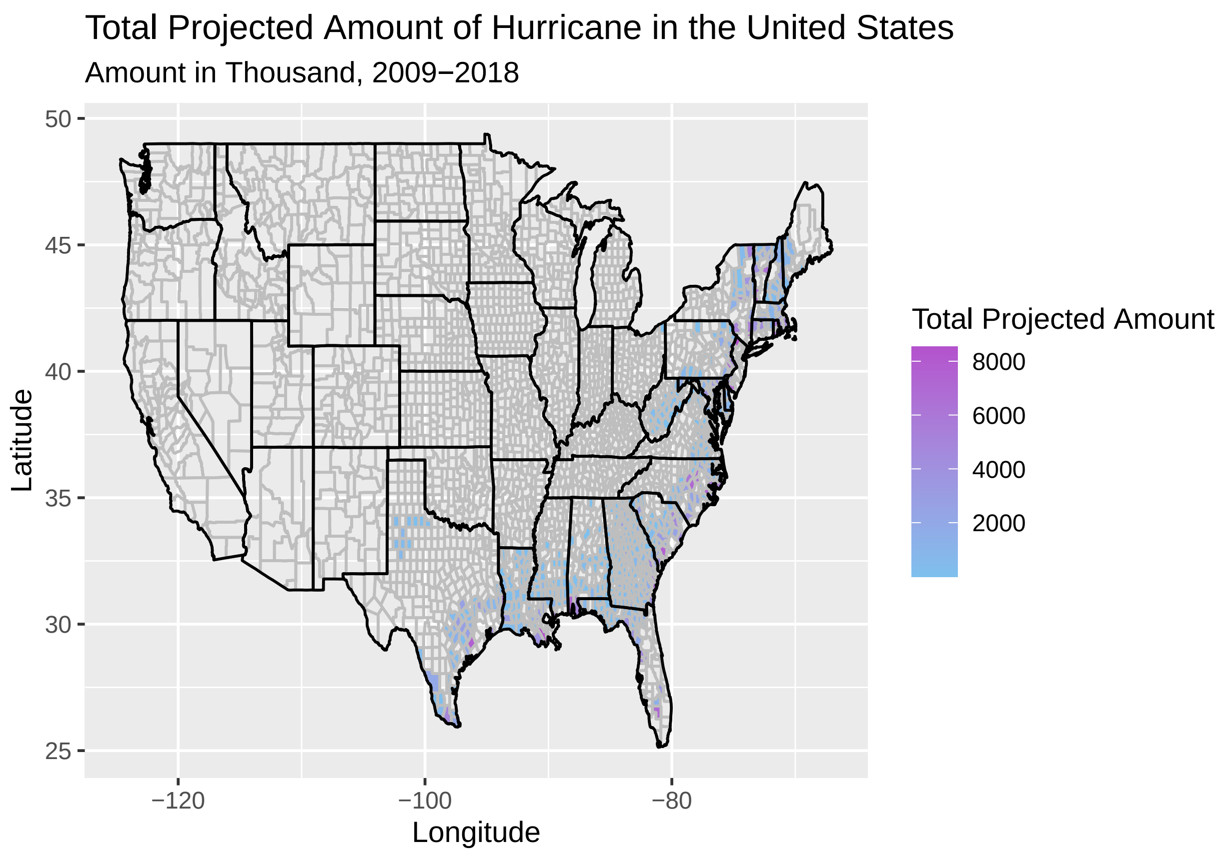
```r
              aes(x=long,y=lat,group=group,fill=TotalAmount))+

  geom_path(data=county_info,mapping=aes(long,lat,group=group),color="grey")+
  geom_path(data=us_state,mapping=aes(long,lat,group=group))
```

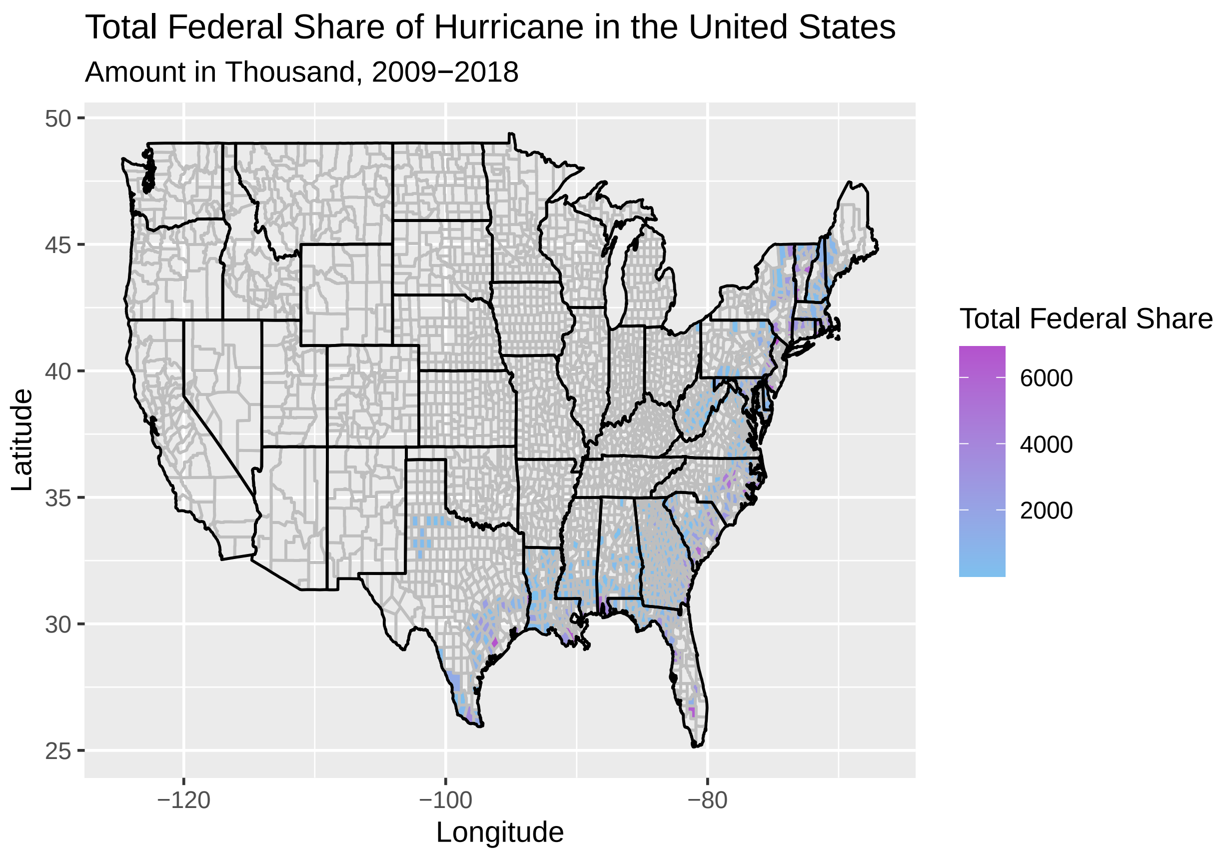
```r
p1 + labs(title="Total Obligated of Hurricane in the United States",
          subtitle="Amount in Thousand, 2009-2018",
          fill="Total Obligated" ) +
  xlab("Longitude") + ylab("Latitude") +
  scale_fill_continuous(low="skyblue2",high="mediumorchid3",space="Lab")
```



3.2.2 Hurricane: Total Projected Amount



3.2.3 Hurricane: Total Federal Share



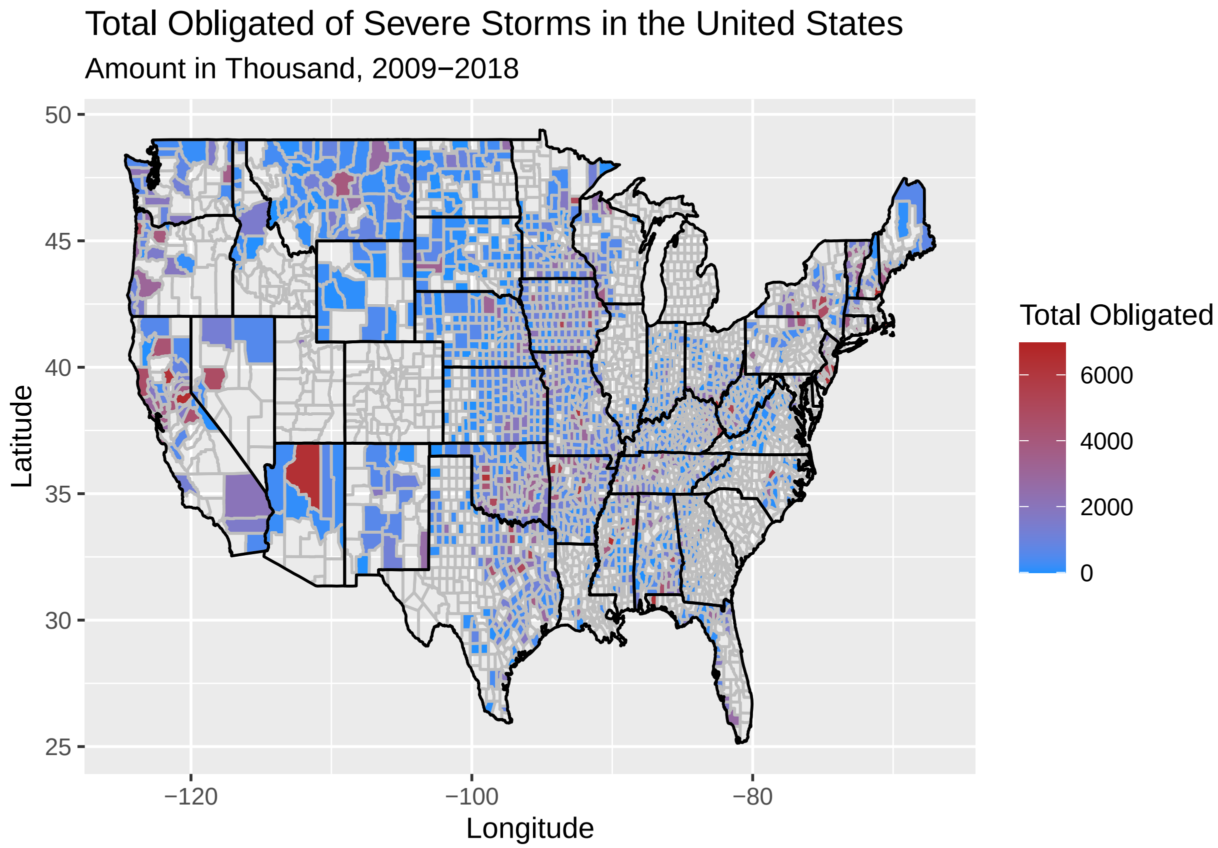
3.3 Damage Data of Severe Storms

3.3.1 Severe Storms: Total Obligated

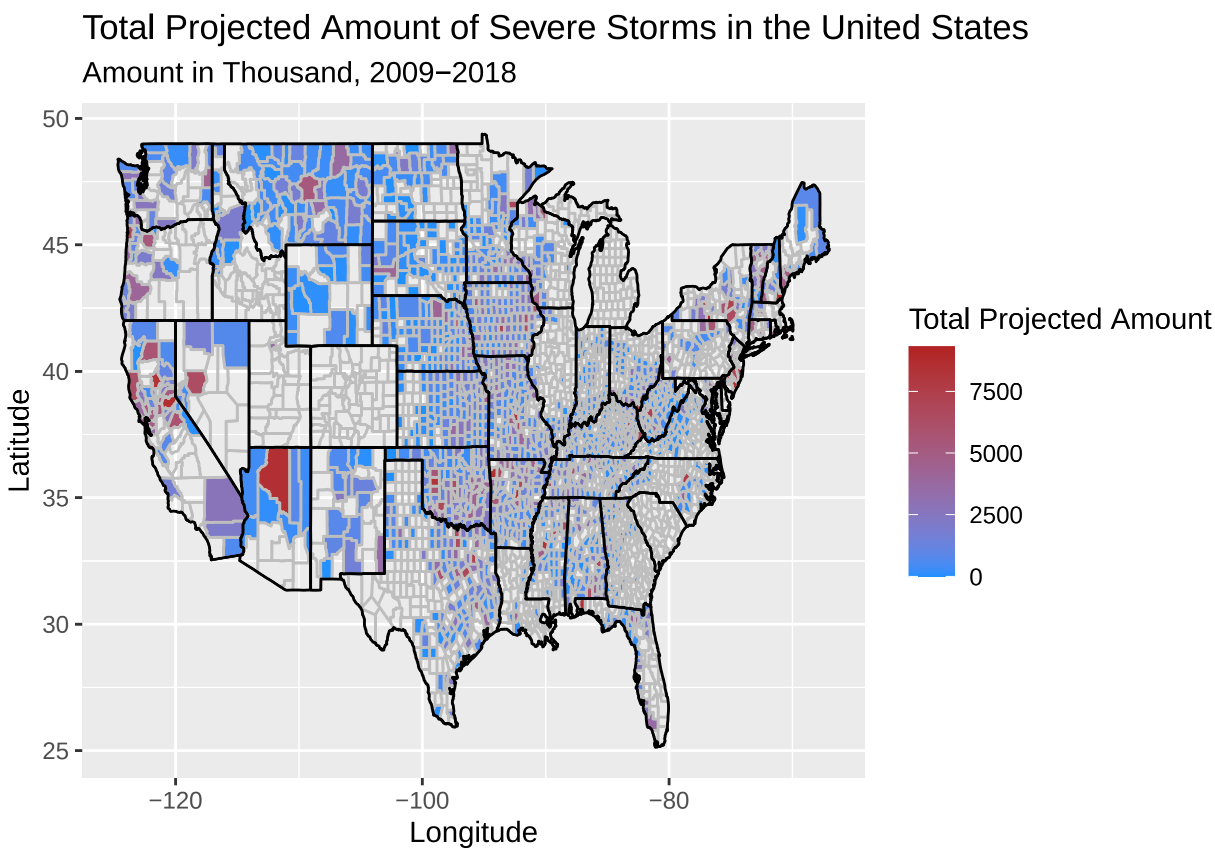
```
p4 <- ggplot()+
  geom_polygon(data=hurri_storm_new %>%
                 filter(incidentType=="Severe Storm(s)"),
               aes(x=long,y=lat,group=group,fill=TotalAmount))+

  geom_path(data=county_info,mapping=aes(long,lat,group=group),color="grey")+
  geom_path(data=us_state,mapping=aes(long,lat,group=group))
```

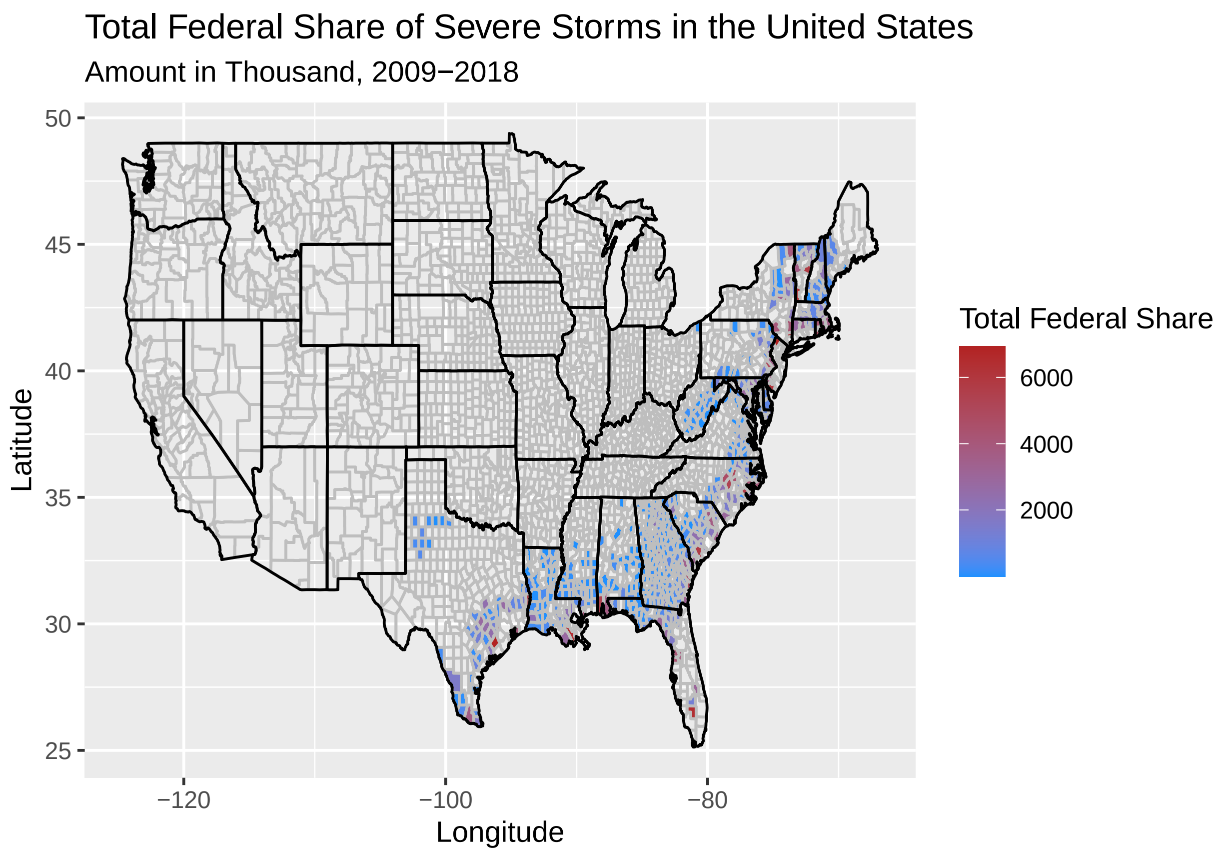
```
p4 + labs(title="Total Obligated of Severe Storms in the United States",
         subtitle="Amount in Thousand, 2009-2018",
         fill="Total Obligated" ) +
  xlab("Longitude") + ylab("Latitude") +
  scale_fill_continuous(low="dodgerblue1",high="firebrick",space="Lab")
```


Total Obligated of Severe Storms in the United States
Amount in Thousand, 2009−2018
Latitude
Longitude
50
45
40
35
30
25
−120
−100
−80
Total Obligated
6000
4000
2000
0

3.3.2 Severe Storms: Total Projected Amount



3.3.3 Severe Storms: Total Federal Share



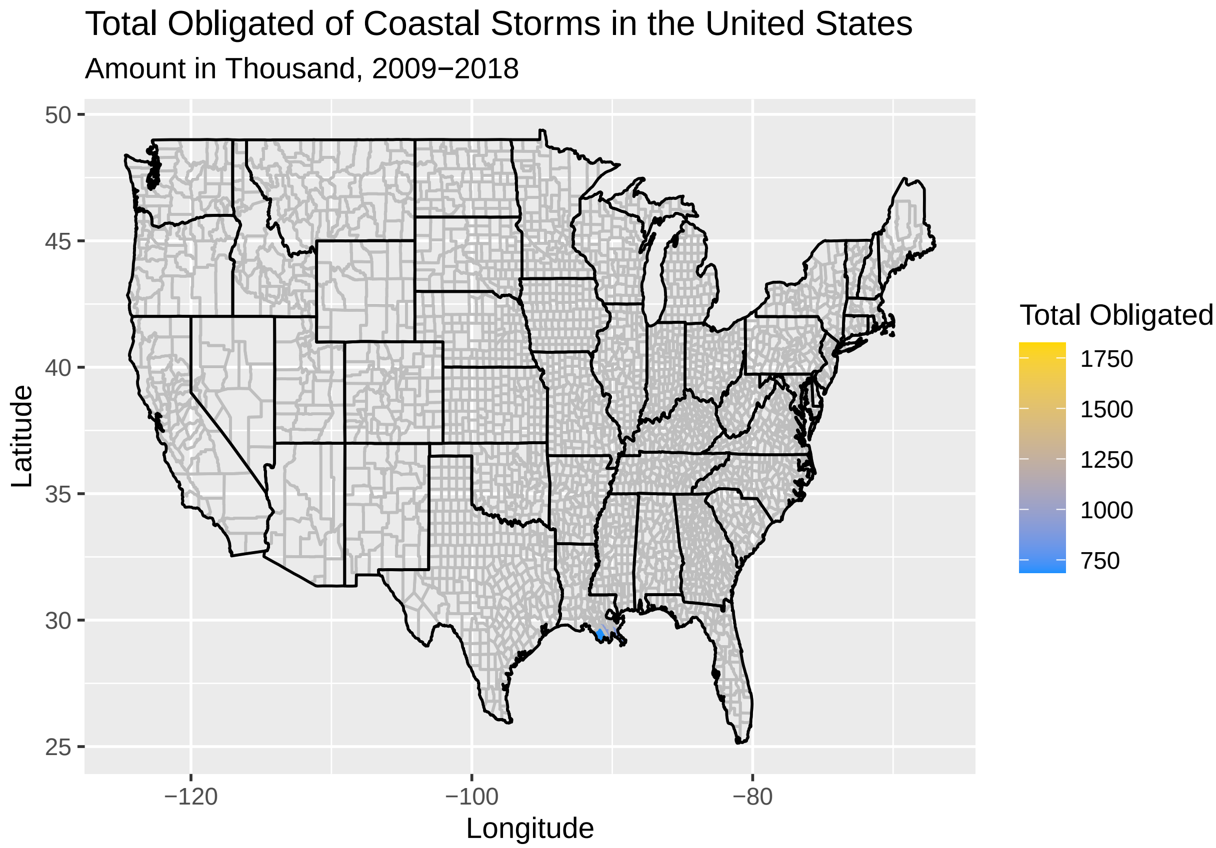
3.4 Damage Data of Coastal Storm

3.4.1 Coastal Storm: Total Obligated

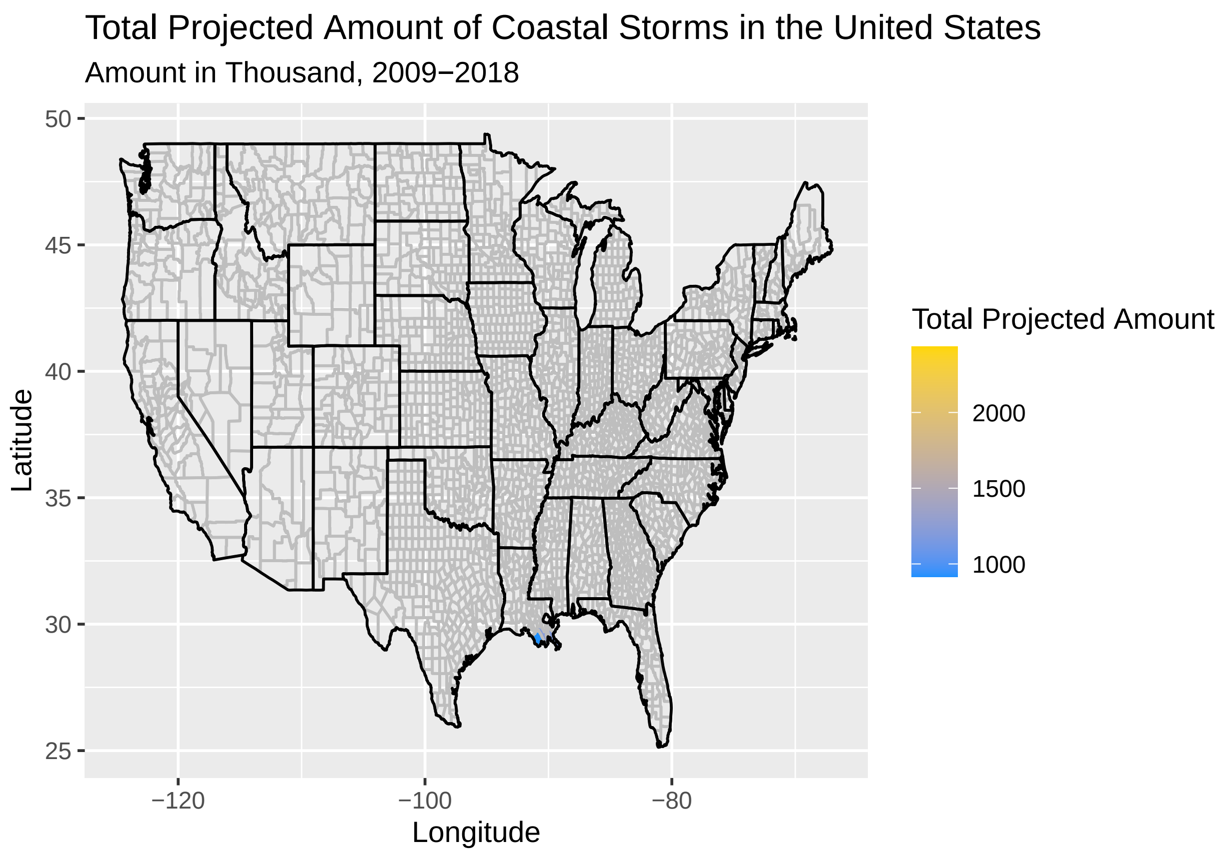
```
p7 <- ggplot()+
  geom_polygon(data=hurri_storm_new %>%
                 filter(incidentType=="Coastal Storm"),
               aes(x=long,y=lat,group=group,fill=TotalAmount))+

  geom_path(data=county_info,mapping=aes(long,lat,group=group),color="grey")+
  geom_path(data=us_state,mapping=aes(long,lat,group=group))
```

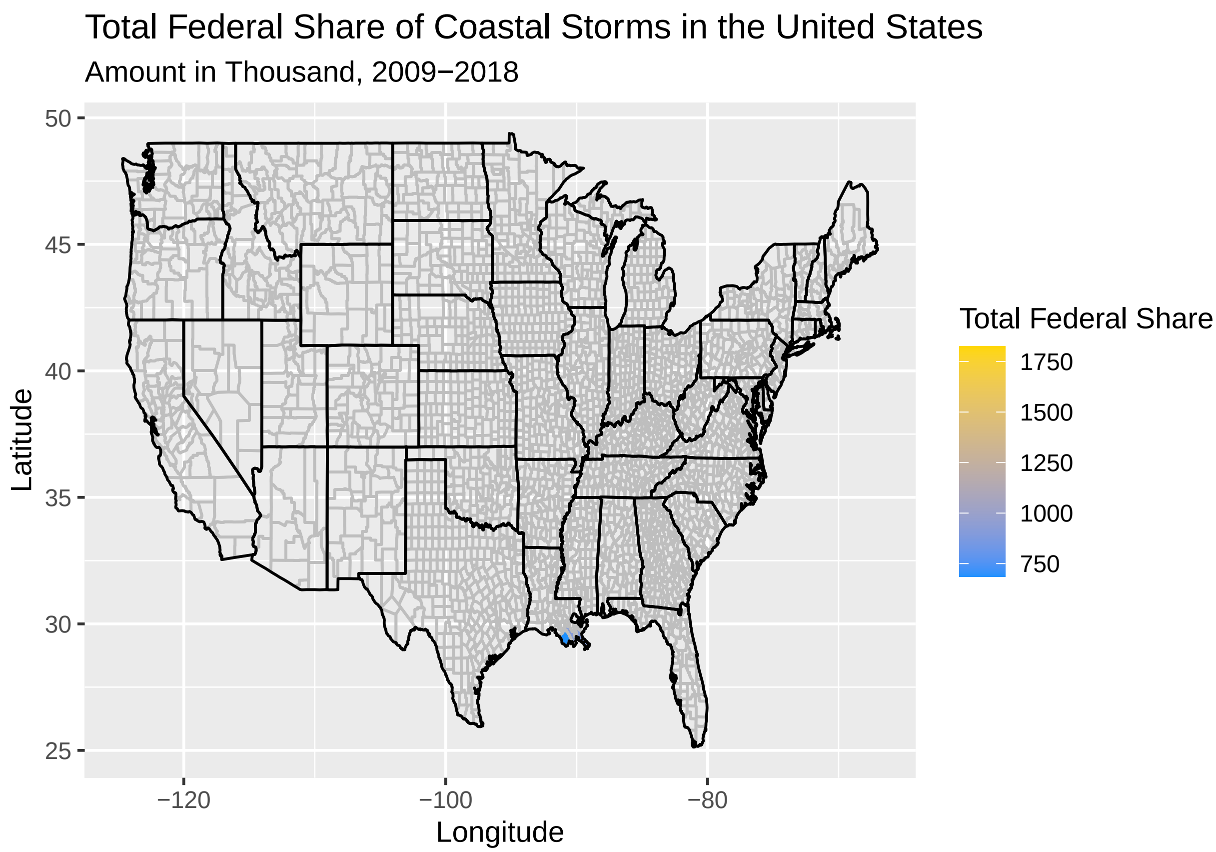
```
p7 + labs(title="Total Obligated of Coastal Storms in the United States",
         subtitle="Amount in Thousand, 2009-2018",
         fill="Total Obligated" ) +
  xlab("Longitude") + ylab("Latitude") +
  scale_fill_continuous(low="dodgerblue1",high="gold",space="Lab")
```


Total Obligated of Coastal Storms in the United States
Amount in Thousand, 2009−2018
50
45
40
35
30
25
Latitude
−120
−100
−80
Longitude
Total Obligated
1750
1500
1250
1000
750

3.4.2 Coastal Storm: Total Projected Amount



3.4.3 Coastal Storm: Total Federal Share



4 Reference

[1] Hadley Wickham, Romain Fran?ois, Lionel Henry and Kirill Müller (2020). dplyr: A Grammar of Data Manipulation. R package version 1.0.2. https://CRAN.R-project.org/package=dplyr

[2] Original S code by Richard A. Becker, Allan R. Wilks. R version by Ray Brownrigg. Enhancements by Thomas P Minka and Alex Deckmyn. (2018). maps: Draw Geographical Maps. R package version 3.3.0. https://CRAN.R-project.org/package=maps

[3] Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL http://www.jstatsoft.org/v40/i03/.

[4] H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

[5] Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." Journal of Open Source Software, 4(43), 1686. https://cran.r-project.org/web/packages/tidyverse/index.html.