Extended Mapping Report

Ruxin Liu, Jinzhe Zhang

11/9/2020

Introduction

In this report, we used the data from the FEMA website and this data is about the summary of Public Assistance Funded Projects. We are specifically interested in the hurricane disasters happened from the year 2009 to the year 2018 and there are no hurricanes been declared as disasters in year 2014 and year 2015. Therefore, we only used a small subset of the data and this data can be found here: https://www.fema.gov/openfema-data-page/public-assistance-funded-projects-details-v1 (https://www.fema.gov/openfema-data-page/public-assistance-funded-projects-details-v1)

We will produce one report, one published shiny app and one presentation slide using revealjs to explore any interesting trends or patterns from the hurricane data. All the codes, projects and data being used can be found in the GitHub Repo.

Date Preparation

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Read in the data
funding <- read.csv("PublicAssistanceFundedProjectsDetails.csv")
```

```
# Select the data related to Hurricane
funding <- funding %>%
  filter(incidentType == "Hurricane")
```

```
# Separate the date to get year information
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.3     ✓ stringr 1.4.0
## ✓ tidyr   1.1.2     ✓ forcats 0.5.0
## ✓ readr   1.3.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
funding <- funding %>%
  separate(declarationDate, c("year", "month", "day"), sep = "-")
# Change the format from character to integer
funding$year <- as.integer(funding$year)
```

```r
# Subset the data from year 2009 to year 2018
funding <- funding %>%
  filter(year >= 2009 & year <= 2018)
# The sub-data funding is stored and uploaded in GitHub Repo
# Remove some columns
funding <- funding %>%
  select(-c(day, incidentType, hash, id, lastRefresh, obligatedDate))
```

```r
# Create fips for county
library(maps)
# head(county.fips)
county_fip <- county.fips %>%
  separate(polyname, c("state", "county"), sep = ",")
```

```r
# Capitalize the state and county name for consistency
county_fip$state <- toupper(county_fip$state)
funding$state <- toupper(funding$state)
county_fip$county <- toupper(county_fip$county)
funding$county <- toupper(funding$county)
# Add the fips into our funding data set
funding <- left_join(funding, county_fip, by = c("county", "state"))
```

```r
funding_sum <- funding %>%
  group_by(fips, state, county, year) %>%
  summarize(project_amount = sum(projectAmount))
```

```r
# Get the longitude and latitude information for each county
county <- (map_data("county"))
colnames(county)[5] <- "state"
colnames(county)[6] <- "county"
# Capitalize the state and county name for consistency
county$state <- toupper(county$state)
county$county <- toupper(county$county)
state <- map_data("state")
```

```r
# Add the longitude and latitude information into our funding data
funding_sum <- right_join(funding_sum, county, by = c("county", "state"))
```

```r
# Find the range cut points
# summary(funding_sum$project_amount)
range_amount <- cut(funding_sum$project_amount, breaks = c(0, 1.5e+05, 1e+06, 6e+06, 2e+10),
                     include.lowest = TRUE)
funding_sum$range <- range_amount
```

EDA

In the table below, the number of hurricanes being declared as disasters in each year is shown.

```r
library(kableExtra)
# Number of hurricanes in each year
hurr <- funding %>%
  group_by(year) %>%
  summarize(number = length(unique(disasterNumber)))
kable(hurr)
```

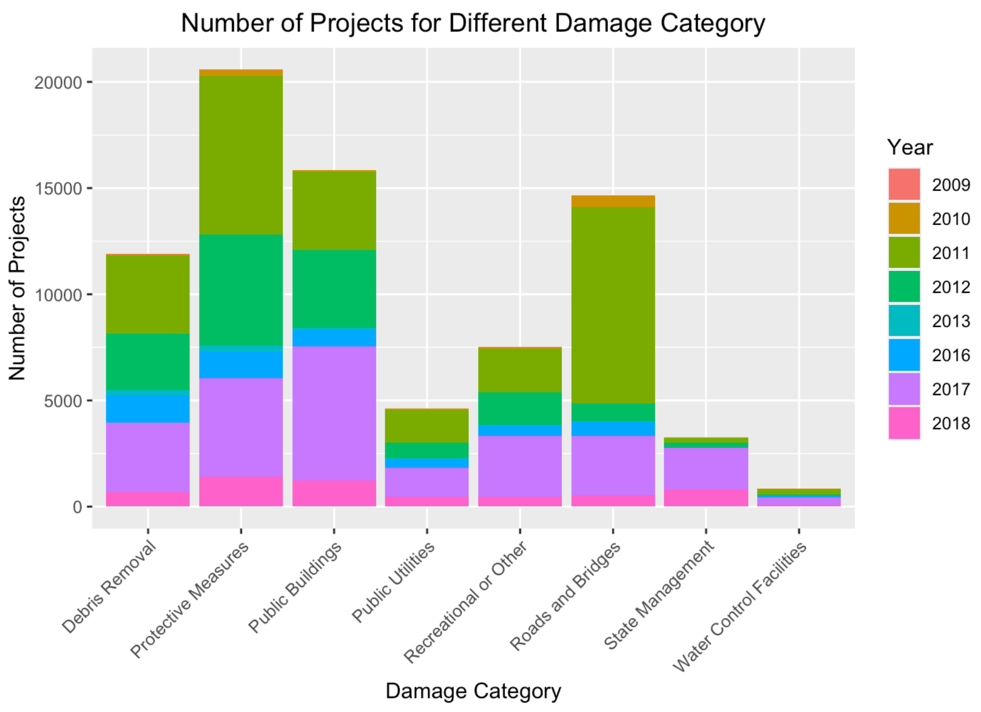
year	number
2009	1
2010	4
2011	22
2012	17
2013	2
2016	6
2017	16
2018	9

```
damage <- funding %>%
  group_by(damageCategory, year) %>%
  summarize(number = length(damageCategory))
```

```
## `summarise()` regrouping output by 'damageCategory' (override with `.groups` argument)
```

In the plot below, we explored the number of projects due to different damage categories from year 2009 ro year 2018. And it is cler that the distribution is quite different.

```
ggplot(damage) +
      geom_col(aes(x = damageCategory, y = number, fill = factor(year))) +
      ggtitle("Number of Projects for Different Damage Category") +
      labs(x = "Damage Category", y = "Number of Projects", fill = "Year") +
      theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
      theme(plot.title = element_text(hjust = 0.5))
```



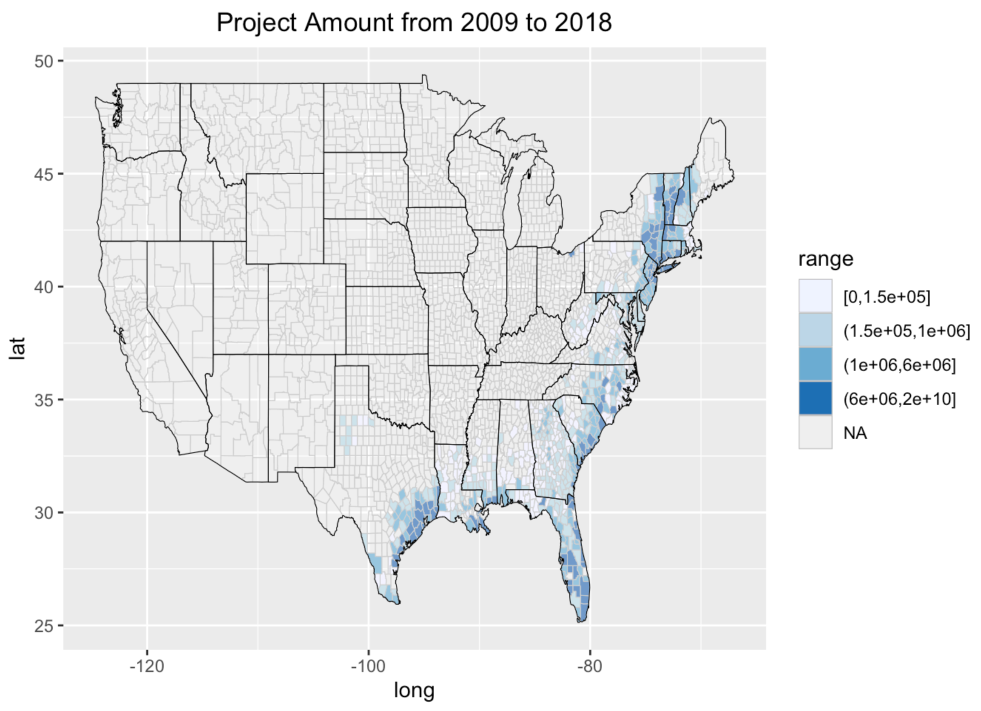
```
obligate <- funding %>%
  group_by(state, disasterNumber, year) %>%
  summarize(aaverage_federal_share_obligated = mean(federalShareObligated),
            average_total_obligated = mean(totalObligated))
```

```
## `summarise()` regrouping output by 'state', 'disasterNumber' (override with `.groups` argument)
```

Mapping with ggplot

The project amount variable is the estimated total cost of the Public Assistance grant project in dollars. And we plot these amounts on the map to see the difference in total amount among different counties from the year 2009 to the year 2018.

```
ggplot() + geom_polygon(data = funding_sum, aes(x = long, y = lat, group = group,
                                                fill = range),
                        color = "grey", size = 0.2, alpha = 1.6) +
  geom_polygon(data = state, aes(x = long, y = lat, group = group),
               color = "black", fill = "white", size = 0.2, alpha = 0.3) +
      scale_fill_brewer(palette = "Blues") +
  ggtitle("Project Amount from 2009 to 2018") +
  # Center the title
  theme(plot.title = element_text(hjust = 0.5))
```



```
# Calculate the total number of projects in each state
funding_year <- funding %>%
  group_by(stateCode, state) %>%
  summarize(number = length(stateCode))
```

```
## `summarise()` regrouping output by 'stateCode' (override with `.groups` argument)
```

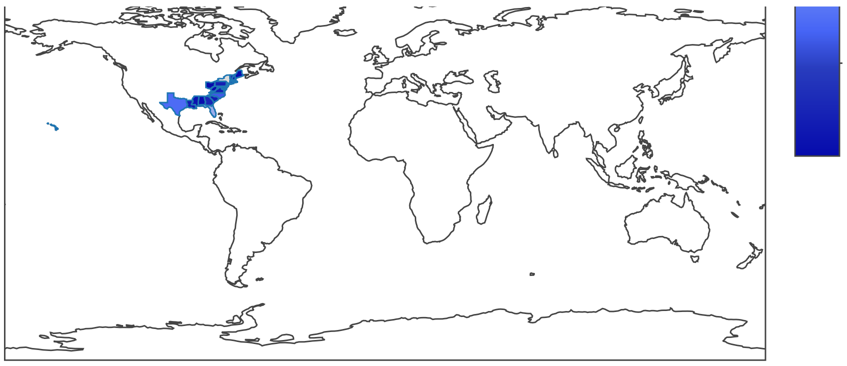
Interactive map with plotly

In this interactive map, we calculated and showed the total number of funded projects in each state from the year 2009 to the year 2018. Since this map is an interactive one, the report can only be shown in an HTML file.

```
library(plotly)
# Create hover on the map
funding_year$hover <- with(funding_year, paste("State:", state,"<br>","Project Number:",number,"<br>"))
project_number <- plot_geo(funding_year, locationmode = 'USA-states')
project_number <- project_number %>% add_trace(
  locations = ~stateCode,
  type = 'choropleth',
  z = ~number,
  text = ~hover,
  colorscale = "Blues"
)
# Add title
project_number <- project_number %>% layout(
  title = 'Number of Hurricane Project from 2009 - 2018'
)

project_number
```

```
## Warning: `arrange_()` is deprecated as of dplyr 0.7.0.
## Please use `arrange()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```



Adding more data from FEMA

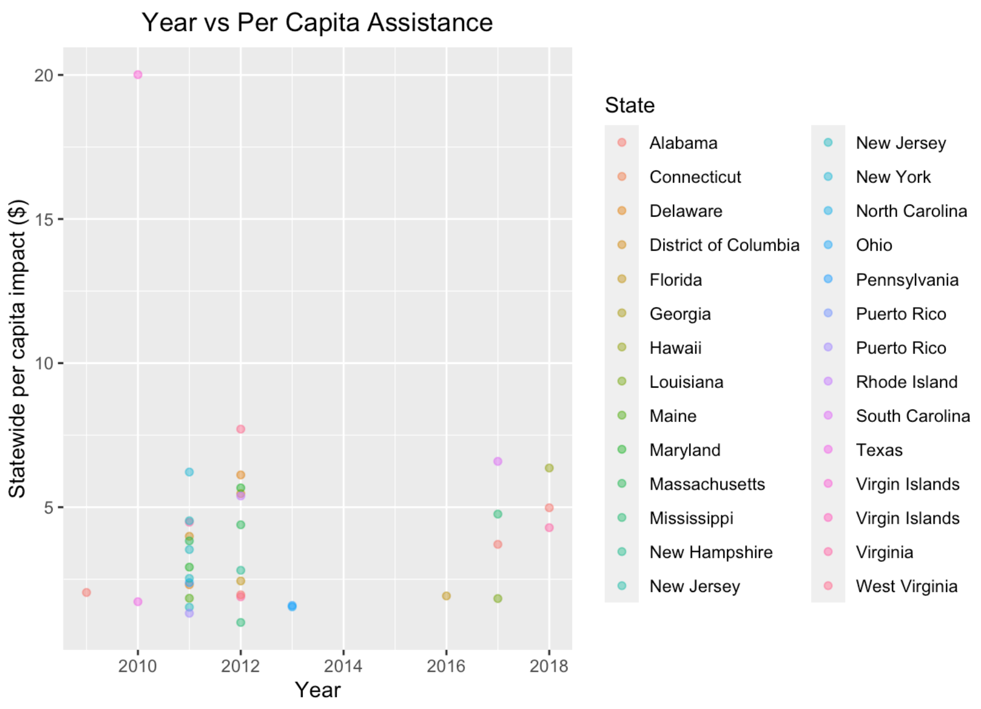
On the FEMA website, we could search for the declared disasters from here:https://www.fema.gov/disasters/disaster-declarations?field_dv2_state_territory_tribal_value=All&field_year_value=All&field_dv2_declaration_type_value=All&field_dv2_incident_type_target_id_selective=49124 (https://www.fema.gov/disasters/disaster-declarations?field_dv2_state_territory_tribal_value=All&field_year_value=All&field_dv2_declaration_type_value=All&field_dv2_incident_type_target_id_selective=49124)

We selected all the declared hurricanes year by year from 2009 to 2018 and collected the statewide per capita assistance in dollars. These information are recorded by FEMA under Preliminary Damage Assessment in pdf files, so these data are manually collected and stored in the FEMA.csv data file. This information allows us to compare the public assistance between different states (2009 - 2017： Democratic Party, 2017 - 2018： Republican Party). The actual declared disasters are more than the ones showing in the data set, but some of the hurricanes are missing this damage assessment information on the FEMA website. Even for some declared hurricanes having the Preliminary Damage Assessment, the Statewide per capita impact value is missing, probably because there is no assistance requested in that case.

```
# Load the data in
assistance <- read.csv("FEMA.csv")
```

```
ggplot(data = assistance, aes(Year, Statewide_per_capita_impact))+
  geom_point(aes(color = State), alpha=0.5)+
  labs(title = "Year vs Per Capita Assistance",
       y = "Statewide per capita impact ($)", x = "Year") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Removed 22 rows containing missing values (geom_point).
```



Preparation for Shiny app

```
# Create a separate table for shiny to make shiny app run faster
funding <- read.csv("funding.csv")
Mapping_table <- funding
Mapping_table <- bind_cols(funding$year, funding$countyCode, funding$stateCode,
                           funding$stateNumberCode, funding$projectAmount)
```

```
## New names:
## * NA -> ...1
## * NA -> ...2
## * NA -> ...3
## * NA -> ...4
## * NA -> ...5
```

```
names(Mapping_table) <- c("year", "countyCode", "stateCode", "stateNumberCode", "projectAmount")
Mapping_table$countyrealcode <- str_pad(Mapping_table$countyCode,3,side = "left",pad = "0")

Mapping_table$statecode <- str_pad(Mapping_table$stateNumberCode,2,side="left",pad="0")

Mapping_table$fips <- str_c(Mapping_table$statecode,Mapping_table$countyrealcode)

Mapping_tables_2010 <- Mapping_table %>% subset(Mapping_table$year=="2010")%>%
  group_by(fips,stateCode,year) %>%
  summarize(total = sum(projectAmount))
```

```
## `summarise()` regrouping output by 'fips', 'stateCode' (override with `.groups` argument)
```

```
Mapping_tables_2011 <- Mapping_table %>% subset(Mapping_table$year=="2011")%>%
  group_by(fips,stateCode,year) %>%
  summarize(total = sum(projectAmount))
```

```
## `summarise()` regrouping output by 'fips', 'stateCode' (override with `.groups` argument)
```

```
Mapping_tables_2012 <- Mapping_table %>% subset(Mapping_table$year=="2012")%>%
  group_by(fips,stateCode,year) %>%
  summarize(total = sum(projectAmount))
```

```
## `summarise()` regrouping output by 'fips', 'stateCode' (override with `.groups` argument)
```

```
Mapping_tables_2013 <- Mapping_table %>% subset(Mapping_table$year=="2013")%>%
  group_by(fips,stateCode,year) %>%
  summarize(total = sum(projectAmount))
```

```
## `summarise()` regrouping output by 'fips', 'stateCode' (override with `.groups` argument)
```

```
Mapping_tables_2009 <- Mapping_table %>% subset(Mapping_table$year=="2009")%>%
  group_by(fips,stateCode,year) %>%
  summarize(total = sum(projectAmount))
```

```
## `summarise()` regrouping output by 'fips', 'stateCode' (override with `.groups` argument)
```

```
Mapping_tables_2016 <- Mapping_table %>% subset(Mapping_table$year=="2016")%>%
  group_by(fips,stateCode,year) %>%
  summarize(total = sum(projectAmount))
```

```
## `summarise()` regrouping output by 'fips', 'stateCode' (override with `.groups` argument)
```

```
Mapping_tables_2017 <- Mapping_table %>% subset(Mapping_table$year=="2017")%>%
  group_by(fips,stateCode,year) %>%
  summarize(total = sum(projectAmount))
```

```
## `summarise()` regrouping output by 'fips', 'stateCode' (override with `.groups` argument)
```

```
Mapping_tables_2018 <- Mapping_table %>% subset(Mapping_table$year=="2018")%>%
  group_by(fips,stateCode,year) %>%
  summarize(total = sum(projectAmount))
```

```
## `summarise()` regrouping output by 'fips', 'stateCode' (override with `.groups` argument)
```

```
Mapping_table_total <- bind_rows(Mapping_tables_2009, Mapping_tables_2010,
                                 Mapping_tables_2011, Mapping_tables_2012,
                                 Mapping_tables_2013, Mapping_tables_2016,
                                 Mapping_tables_2017, Mapping_tables_2018 )
```

Reference

1. Hadley Wickham, Romain François, 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 (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 (https://CRAN.R-project.org/package=maps)
3. Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686 (https://doi.org/10.21105/joss.01686)
4. C. Sievert. Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC Florida, 2020.
5. FEMA: https://www.fema.gov/openfema-data-page/public-assistance-funded-projects-details-v1 (https://www.fema.gov/openfema-data-page/public-assistance-funded-projects-details-v1)
6. FEMA: https://www.fema.gov/disasters/disaster-declarations?field_dv2_state_territory_tribal_value=All&field_year_value=All&field_dv2_declaration_type_value=All&field_dv2_incident_type_target_id_selective=4 (https://www.fema.gov/disasters/disaster-declarations?field_dv2_state_territory_tribal_value=All&field_year_value=All&field_dv2_declaration_type_value=All&field_dv2_incident_type_target_id_selective=4
7. Hao Zhu (2019). kableExtra: Construct Complex Table with 'kable' and Pipe Syntax. R package version 1.1.0. https://CRAN.R-project.org/package=kableExtra (https://CRAN.R-project.org/package=kableExtra)