Midterm project about Flood, Chenghan Wu

2023-10-31

```
library(readr)
library(tidyr)
library(dplyr)
library(ggplot2)
library(lubridate)
library(tidyverse)
```

Main Question

One interesting aspect of floods is that they can occur almost anywhere. How dangerous are floods? How expensive? Is there any pattern to the kinds of communities that suffer losses from floods?

Assemble a dataset to investigate floods in 2020-2021. Use the data resources below to assemble your data. Clean and organize the data. Write an EDA report.

Dataset explanation

The dataset I plan to use is from https://www.fema.gov/openfema-data-page/fema-web-disaster-summaries-v1 (https://www.fema.gov/openfema-data-page/fema-web-disaster-summaries-v1) https://www.fema.gov/openfema-data-page/disaster-declarations-summaries-v2 (https://www.fema.gov/openfema-data-page/disaster-declarations-summaries-v2) and https://www.ncei.noaa.gov/pub/data/swdi/stormevents/csvfiles/ (https://www.ncei.noaa.gov/pub/data/swdi/stormevents/csvfiles/) with year in 2020 and 2021

We can find detailed variable explanation in these links.

Plan

Outline the approach take to clean and organize the data.

1. Take a look at the data set I plan to use.
2. Explore data set of disaster_total, Try to filter the condition for require.
3. Find missing value percent in dataset.
4. Make a visualization about info about financial assistance average amount and interpret.
5. Explore storm dataset, plot the death and Property damage by state.
6. Plot The source info reporting the weather event.
7. Duraiton of flood.
8. Finally, we want to take a look at the distribution of the flood last.

1. Load the dataset we have

```
detail1 = read_csv("StormEvents_details-ftp_v1.0_d2020_c20230927.csv")
detail2 = read_csv("StormEvents_details-ftp_v1.0_d2021_c20231017.csv")
disaster_total = read_csv("DisasterDeclarationsSummaries.csv")
disaster_summary = read_csv("FemaWebDisasterSummaries (1).csv")
storm_events <- bind_rows(detail1, detail2)
```

2. Explroe dataset of disaster_total, Try to filter the condition we want.

```
unique(disaster_total$incidentType)
```

```
##  [1] "Fire"              "Flood"             "Hurricane"
##  [4] "Severe Storm"      "Winter Storm"      "Tornado"
##  [7] "Snowstorm"         "Earthquake"        "Biological"
## [10] "Mud/Landslide"     "Coastal Storm"     "Other"
## [13] "Severe Ice Storm"  "Dam/Levee Break"   "Tropical Storm"
## [16] "Tsunami"           "Typhoon"           "Volcanic Eruption"
## [19] "Freezing"          "Toxic Substances"  "Chemical"
## [22] "Terrorist"         "Drought"           "Human Cause"
## [25] "Fishing Losses"
```

```
# look only with severe storm related data.
disaster_flood <- disaster_total %>%
  mutate(beginDate = ymd(incidentBeginDate)) %>%
  filter(incidentType %in% c("Flood", "Severe Storm", "Winter Storm", "Severe Ice Stor
m", "Tsunami"),
         beginDate >= ymd('2020-01-01'),
         beginDate <= ymd('2021-12-31'))

# filter with same disasterNumber in summary data
fema_flood_summaries <- disaster_summary %>%
  filter(disasterNumber %in% disaster_flood$disasterNumber)
```

From here, we can see that there is 68 floods related events recorded in 2020 and 2021 in total. We want to see na percent.

3. Find missing value percent in dataset

```r
# Function to calculate NA percentages for each column in a dataframe
na_percentage <- function(df) {
  na_percents <- sapply(df, function(col) {
    sum(is.na(col)) / length(col) * 100
  })
  data.frame(
    Column = names(na_percents),
    NA_Percentage = na_percents
  )
}

na_percentages <- na_percentage(fema_flood_summaries)
print(na_percentages)
```

```
##                                                Column NA_Percentage
## disasterNumber                           disasterNumber      0.000000
## totalNumberIaApproved             totalNumberIaApproved     75.000000
## totalAmountIhpApproved           totalAmountIhpApproved     75.000000
## totalAmountHaApproved             totalAmountHaApproved     75.000000
## totalAmountOnaApproved           totalAmountOnaApproved     75.000000
## totalObligatedAmountPa           totalObligatedAmountPa      7.352941
## totalObligatedAmountCatAb     totalObligatedAmountCatAb      7.352941
## totalObligatedAmountCatC2g totalObligatedAmountCatC2g      8.823529
## paLoadDate                                   paLoadDate      7.352941
## iaLoadDate                                   iaLoadDate     75.000000
## totalObligatedAmountHmgp       totalObligatedAmountHmgp      5.882353
## hash                                               hash      0.000000
## lastRefresh                                 lastRefresh      0.000000
## id                                                   id      0.000000
```

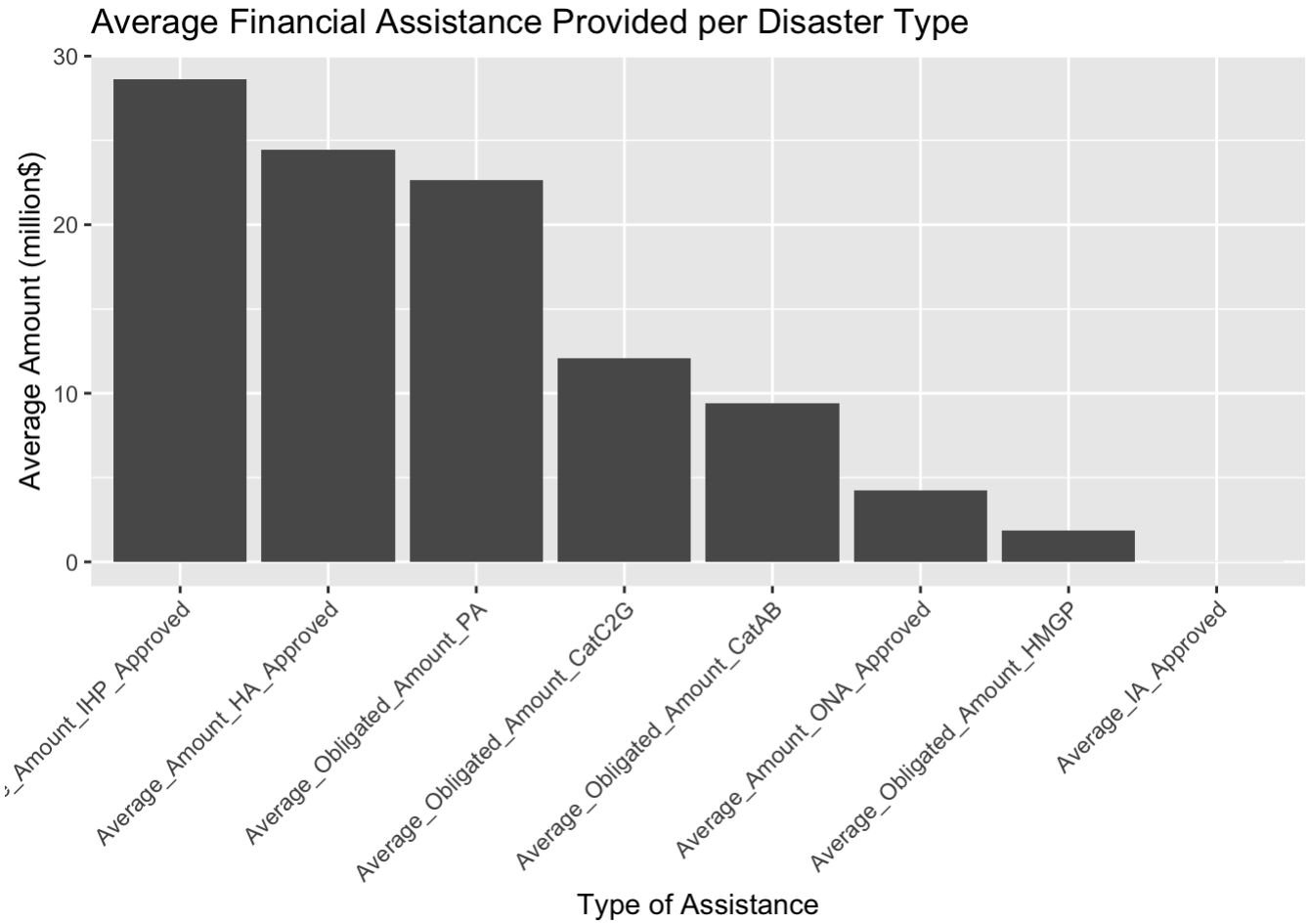
There are a lot of na here, maybe we can visualization some less missing values that is important to discover in the visualization part, about the finalical assitence in that part.

4. find the average of financial assistance values

```r
# Calculate averages for financial data, ignoring NA values
financial_averages <-  fema_flood_summaries  %>%
  summarise(
    Average_IA_Approved = mean(totalNumberIaApproved, na.rm = TRUE),
    Average_Amount_IHP_Approved = mean(totalAmountIhpApproved, na.rm = TRUE),
    Average_Amount_HA_Approved = mean(totalAmountHaApproved, na.rm = TRUE),
    Average_Amount_ONA_Approved = mean(totalAmountOnaApproved, na.rm = TRUE),
    Average_Obligated_Amount_PA = mean(totalObligatedAmountPa, na.rm = TRUE),
    Average_Obligated_Amount_CatAB = mean(totalObligatedAmountCatAb, na.rm = TRUE),
    Average_Obligated_Amount_CatC2G = mean(totalObligatedAmountCatC2g, na.rm = TRUE),
    Average_Obligated_Amount_HMGP = mean(totalObligatedAmountHmgp, na.rm = TRUE)
  ) %>%
  gather(key = "Assistance_Type", value = "Average_Amount")

# View the financial averages
print(financial_averages)
```

```
## # A tibble: 8 × 2
##   Assistance_Type                Average_Amount
##   <chr>                                   <dbl>
## 1 Average_IA_Approved                     7880.
## 2 Average_Amount_IHP_Approved         28655556.
## 3 Average_Amount_HA_Approved          24435844.
## 4 Average_Amount_ONA_Approved          4219711.
## 5 Average_Obligated_Amount_PA         22651625.
## 6 Average_Obligated_Amount_CatAB       9383370.
## 7 Average_Obligated_Amount_CatC2G     12100508.
## 8 Average_Obligated_Amount_HMGP        1884035.
```

```r
ggplot(financial_averages, aes(x = reorder(Assistance_Type,-Average_Amount), y = Average
_Amount/1000000)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Average Financial Assistance Provided per Disaster Type",
       x = "Type of Assistance",
       y = "Average Amount (million$)")
```



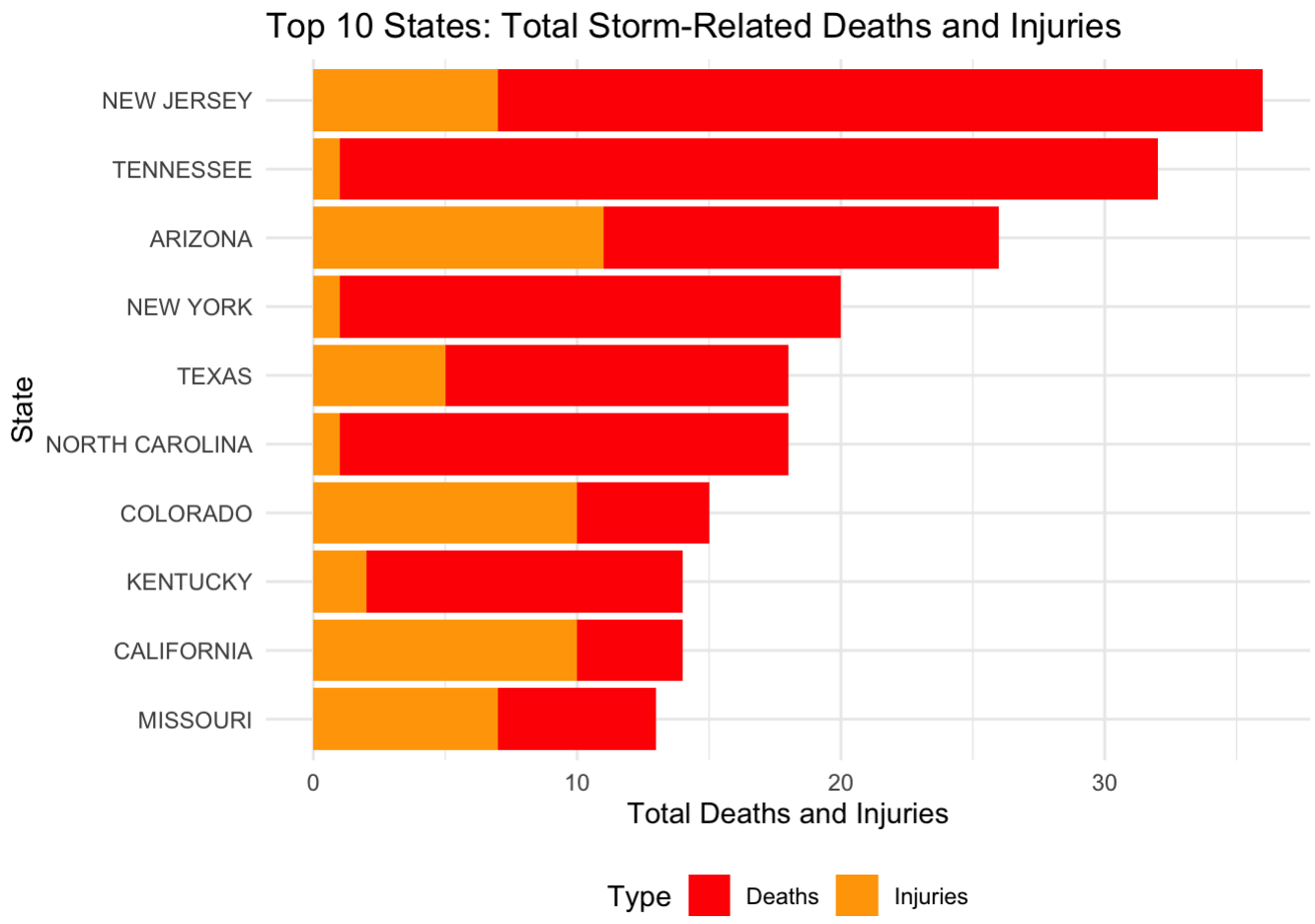
These are the average financial for each disaster and it looks that individual and Households Program and Housing Assistance (HA) and Public Assistance grant funding will take up the most the spending, which makes sense because the housing is destruct most in the flooding.

5. we want to take a look at the storm data to find the deaths

```
# change the data to date format, and filter the type = flood
storm_events <- storm_events %>%
  mutate(beginDate = as.Date(dmy_hms(BEGIN_DATE_TIME)))%>%
  filter(EVENT_TYPE %in% c("Flood","Flash Flood","Ice Storm"))

disaster_flood$fipsStateCode = as.numeric(disaster_flood$fipsStateCode)

# if we want to join the storm and flood with state and time, this is mutiple to mutiple
and not make any sense. Thus, i want show these separately
```

5. thus we want to see the deaths by each state first from storm dataset

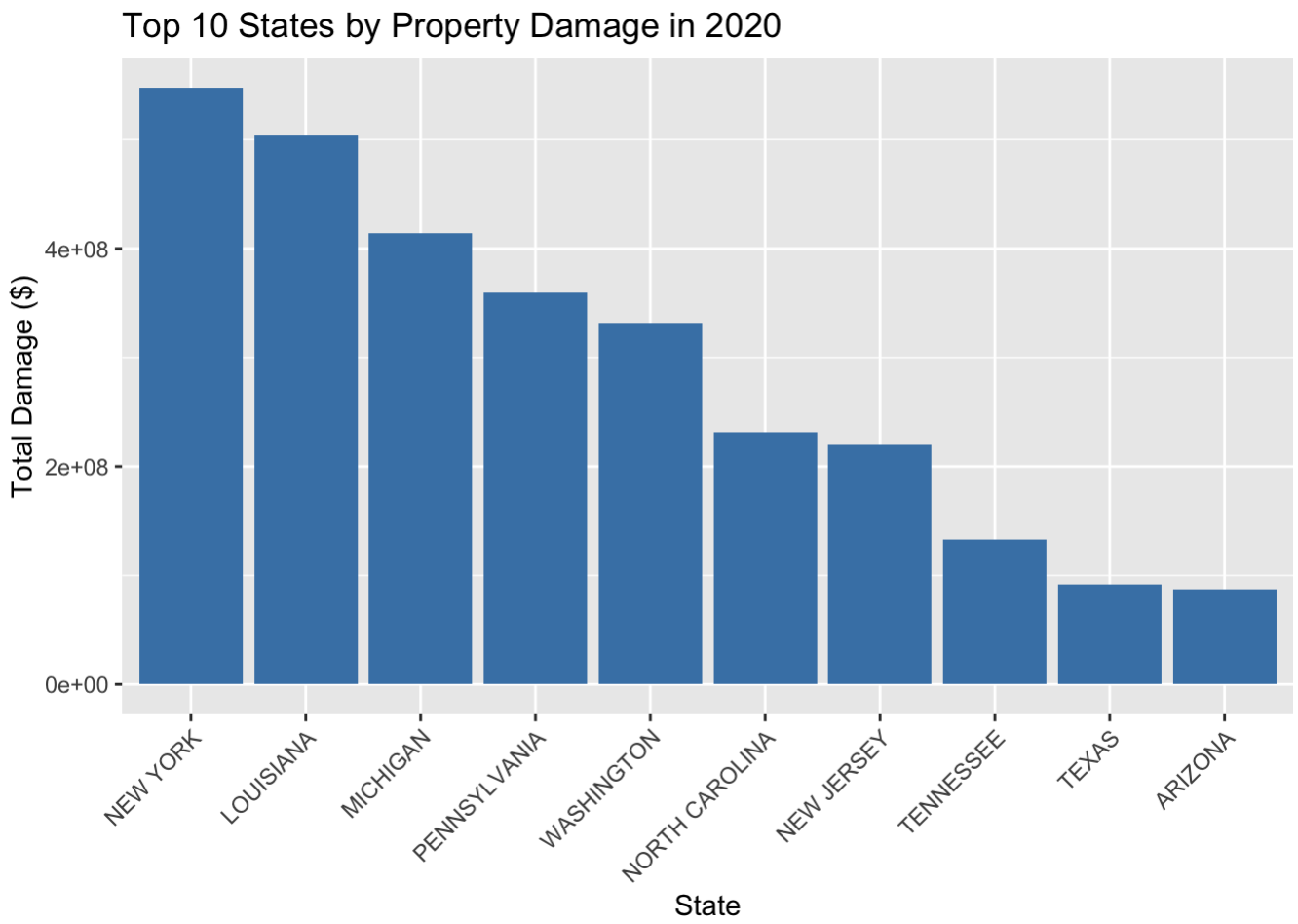
```
# Aggregate the total deaths and injuries by state
deaths_and_injuries_by_state <- storm_events %>%
  group_by(STATE) %>%
  summarise(
    TOTAL_DEATHS = sum(DEATHS_DIRECT, na.rm = TRUE) + sum(DEATHS_INDIRECT, na.rm = TRU
E),
    TOTAL_INJURIES = sum(INJURIES_DIRECT, na.rm = TRUE) + sum(INJURIES_INDIRECT, na.rm =
TRUE)
  ) %>%
  mutate(TOTAL = TOTAL_DEATHS + TOTAL_INJURIES) %>%
  arrange(desc(TOTAL)) %>%
  top_n(10, TOTAL)

# Create the plot
ggplot(deaths_and_injuries_by_state, aes(x=reorder(STATE, TOTAL), y=TOTAL, fill="Death
s")) +
  geom_bar(stat="identity") +
  geom_bar(aes(y=TOTAL_INJURIES, fill="Injuries"), stat="identity") +
  coord_flip() +  # Flip the coordinates to make it a horizontal bar plot
  labs(x="State", y="Total Deaths and Injuries", title="Top 10 States: Total Storm-Relat
ed Deaths and Injuries") +
  theme_minimal() +
  theme(legend.position="bottom") +
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = c("red", "orange"), labels = c("Deaths", "Injuries")) +
  guides(fill=guide_legend(title="Type"))
```



From this plot, we can see that these floods are not super dangerous, the overall injuery and death casued are around 35 highest in new jersey. and Tennessee and Arizona are the following.

Now , I want look into these damage properties

```r
# create function to change chr to numeric
convert_damage_value <- function(damage_str) {
  if (is.na(damage_str)) {
    return(0)
  }
  multipliers <- c(K = 1e3, M = 1e6, B = 1e9)
  if (grepl("[0-9]$", damage_str)) {
    return(as.numeric(damage_str))
  }
  numeric_part <- as.numeric(sub("[KMB]$", "", damage_str))
  suffix <- substring(damage_str, nchar(damage_str))
  return(numeric_part * multipliers[suffix])
}

storm_events$DAMAGE_PROPERTY_NUM <- sapply(storm_events$DAMAGE_PROPERTY, convert_damage_
value)
storm_events$DAMAGE_PROPERTY_NUM = as.numeric(storm_events$DAMAGE_PROPERTY_NUM)

# group by state

damage_by_state <- storm_events %>%
  group_by(STATE) %>%
  summarise(
    TOTAL_DAMAGE = sum(DAMAGE_PROPERTY_NUM, na.rm = TRUE),
  ) %>%
  arrange(desc(TOTAL_DAMAGE )) %>%
  top_n(10, TOTAL_DAMAGE)


##plot

ggplot(damage_by_state, aes(x = reorder(STATE, - TOTAL_DAMAGE), y =  TOTAL_DAMAGE)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  xlab("State") + ylab("Total Damage ($)") +
  ggtitle("Top 10 States by Property Damage in 2020") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

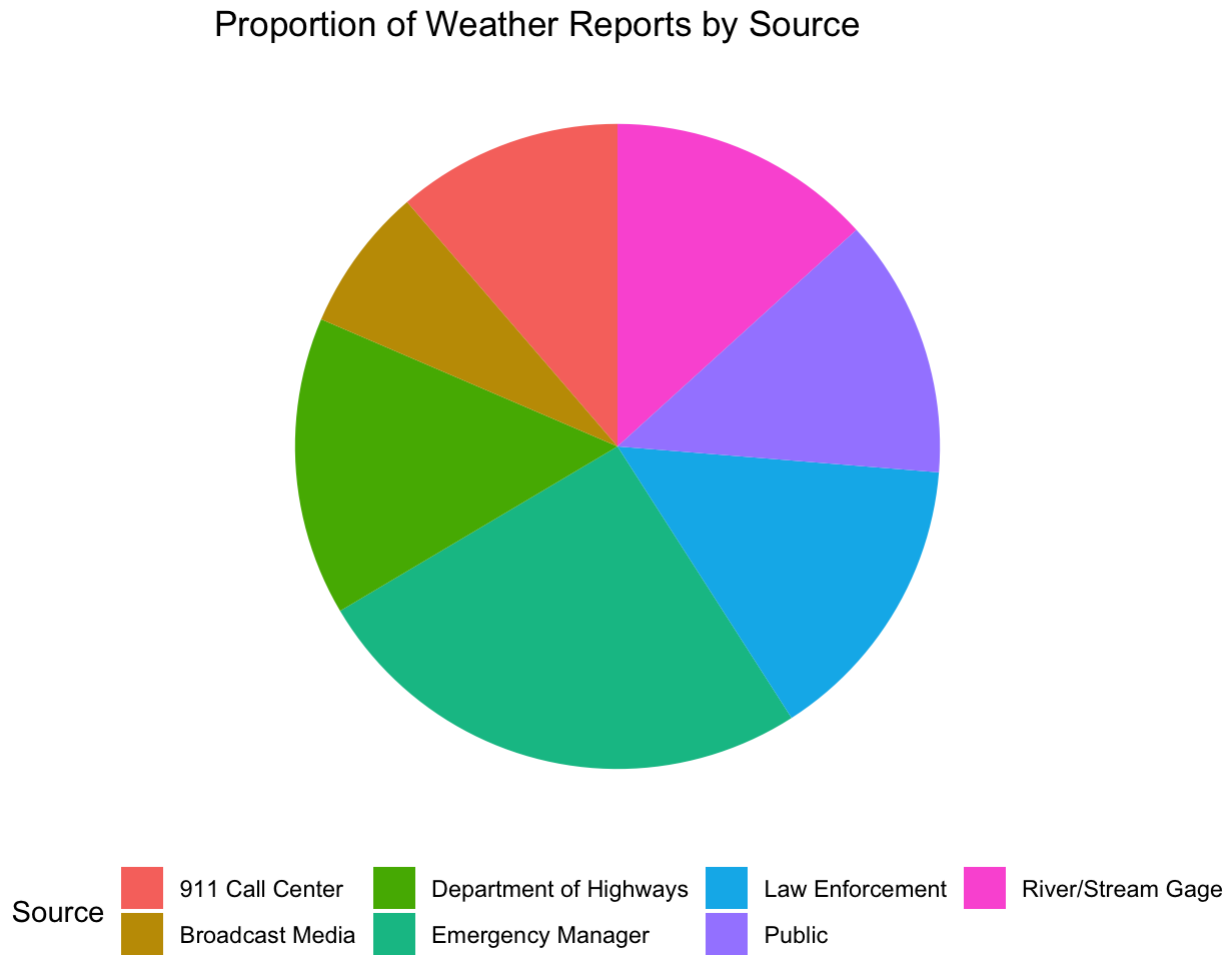


New York has highest number of losses which make sense because it has a lot of money, the damage will be high as well. Moreover, Luisa and Michigan has less value but also has high value of losses, It may induced by there a lot of floods in this area.

And there are several state that suffer most also has top injuries and death people.

6. so I may want to take a look at The source reporting the weather event.

```
source_counts <- storm_events %>%
  count(SOURCE) %>%
  arrange(desc(n))
top_n_sources <- head(source_counts, 7)

ggplot(top_n_sources, aes(x = "", y = n, fill = SOURCE)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y") +
  theme_void() +
  labs(fill = "Source", title = "Proportion of Weather Reports by Source") +
  theme(legend.position = "bottom")
```

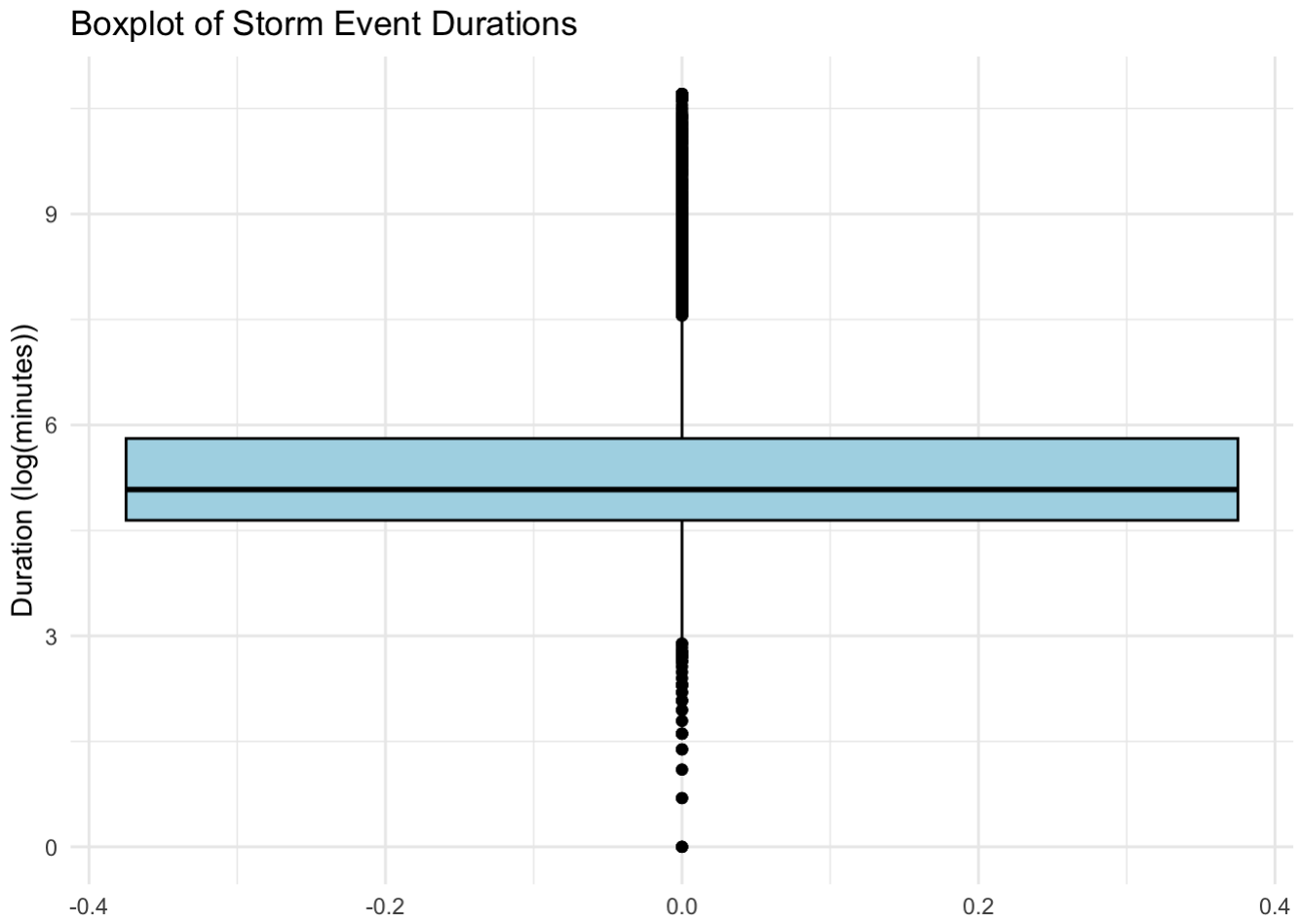


From this plot, we can see that The source reporting the weather event most comes from these 7 sources and next time if we want to know the flood info,we can go to Emergency Manager or Departmetns of high ways.

7. duration of the flood.

```
storm_events$BEGIN_DATE_TIME1 <- strptime(storm_events$BEGIN_DATE_TIME, format="%d-%b-%y
%H:%M:%S")
storm_events$END_DATE_TIME1 <- strptime(storm_events$END_DATE_TIME, format="%d-%b-%y %
H:%M:%S")
storm_events$duration <- difftime(storm_events$END_DATE_TIME1 , storm_events$BEGIN_DATE_
TIME1 , units="mins")

storm_events$duration = as.numeric(storm_events$duration)
# Assuming storm_events$duration is already converted to numeric
ggplot(storm_events, aes(y = log(duration))) +
  geom_boxplot(fill = "lightblue", colour = "black") +
  labs(title = "Boxplot of Storm Event Durations",
       y = "Duration (log(minutes))") +
  theme_minimal()
```

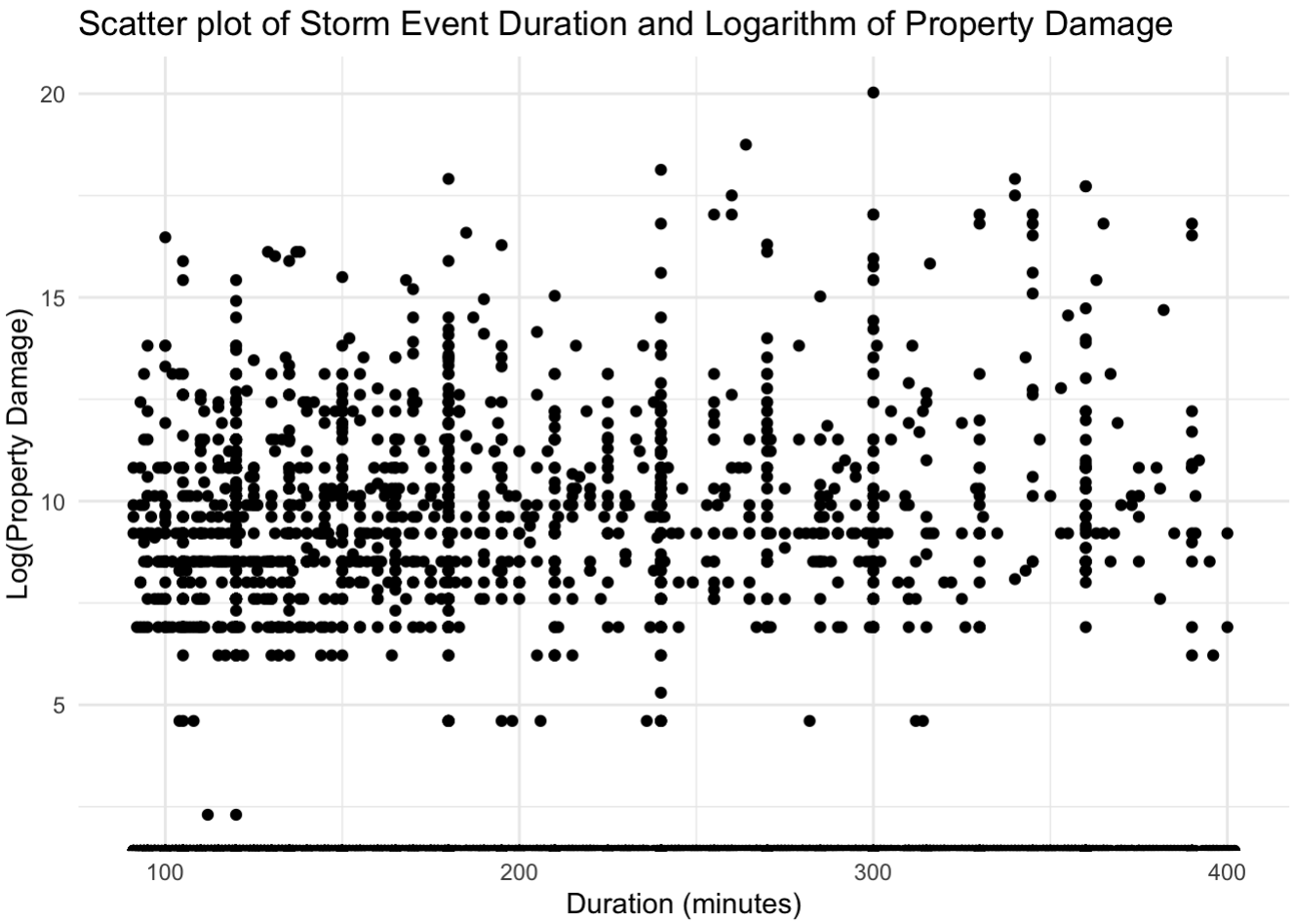
```
## Warning: Removed 786 rows containing non-finite values (`stat_boxplot()`).
```



As we can see from boxplot, we get tons of outliers so we have to using log transformation to see the box. Overall, I think median is around 5 and 75% of flood duration are around 4.5 to 6 og log value, which is 90 to 403 minute.

8. Now, I am trying to plot the relation between duration and DAMAGE_PROPERTY_NUM.

```
ggplot(storm_events[storm_events$duration < 403 &storm_events$duration >90, ],
       aes(x = duration, y = log(DAMAGE_PROPERTY_NUM))) +
  geom_point() +
  labs(title = "Scatter plot of Storm Event Duration and Logarithm of Property Damage",
       x = "Duration (minutes)",
       y = "Log(Property Damage)") +
  theme_minimal()
```



From this scatter plot, I do not see any trend that higher minutes will cause higher property loss for the most of floods. So it is hard to say that the longer the duration of flood,the higher damage value will cause.