NOAA Storm Database Analysis

Michael Harrison

February 22, 2017

Introduction

Storms and other severe weather events can cause both public health and economic problems for communities and municipalities. Many severe events can result in fatalities, injuries, and property damage, and preventing such outcomes to the extent possible is a key concern.

This project involves exploring the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database. This database tracks characteristics of major storms and weather events in the United States, including when and where they occur, as well as estimates of any fatalities, injuries, and property damage.

Your data analysis must address the following questions:

1. Across the United States, which types of events (as indicated in the variable) are most harmful with respect to population health?
2. Across the United States, which types of events have the greatest economic consequences?

Data Processing

Cleaning

Data cleaning and organization will depend upon the following packages: data.table, dplyr, lubridate, and ggplot2(which is loaded later in the analysis to avoid masking issues with dplyr). Data from csv file read in using fread function of the data.table package for speed. All column names are set to lower case for ease in scripting; state columns renamed for clarity. Dates are transformed using the lubridate package function to set column to unambiguous date format. Data is then subset by date, starting on the first of January 1996, as NOAA began recording all event types.

```
knitr::opts_chunk$set(error = TRUE)
library(data.table); library(dplyr);
library(lubridate);
options(scipen = 999)

stormdt_raw <- fread("StormData.csv")
stormdt <- stormdt_raw

colnames(stormdt) <- tolower(colnames(stormdt))
colnames(stormdt)[1] <- "fips"
colnames(stormdt)[7] <- "state"
```

```
#change date formats
stormdt$bgn_date <- mdy_hms(stormdt$bgn_date)
stormdt$end_date <- mdy_hms(stormdt$end_date)
stormdt <- stormdt[bgn_date >= "1996-01-01"]
stormdt$bgn_datetime <- as.POSIXct(paste(stormdt$bgn_date, stormdt$bgn_time),
                                   format = "%Y-%m-%d %H:%M:%S")
stormdt$end_datetime <- as.POSIXct(paste(stormdt$end_date, stormdt$end_time),
                                   format = "%Y-%m-%d %H:%M:%S")
```

The following script takes the factor of property and crop damage and applies it to the damage standardize the value.

```
factordmg <- function(DT){
        DT[propdmgexp == "H" | propdmgexp == "h", propdmg := propdmg * 100]
        DT[propdmgexp == "K" | propdmgexp == "k", propdmg := propdmg * 1000]
        DT[propdmgexp == "M" | propdmgexp == "m", propdmg := propdmg * 1E6]
        DT[propdmgexp == "B" | propdmgexp == "b", propdmg := propdmg * 1E9]
        for(i in 1:9){stormdt[propdmgexp == i, propdmg := propdmg * 10^i]}

        DT[cropdmgexp == "H" | cropdmgexp == "h", cropdmg := cropdmg * 100]
        DT[cropdmgexp == "K" | cropdmgexp == "k", cropdmg := cropdmg * 1000]
        DT[cropdmgexp == "M" | cropdmgexp == "m", cropdmg := cropdmg * 1E6]
        DT[cropdmgexp == 0, cropdmg := cropdmg * 10^0]
        DT[cropdmgexp == 2, cropdmg := cropdmg * 10^2]
}

stormdt <- factordmg(stormdt)
```

Create new datetime columns for event begining and event ending.

```
stormdt <- stormdt %>%
        select(fips, state, county, countyname, evtype,
               time_zone, bgn_datetime, end_datetime, bgn_date, end_date,
               length, width, f, mag,
               fatalities, injuries, propdmg, cropdmg) %>%
        filter(fatalities > 0 | injuries > 0 | propdmg > 0 | cropdmg > 0)
```

```
monthly_cpi <-
        read.table("https://research.stlouisfed.org/fred2/data/CPIAUCSL.csv",
                   header = TRUE, sep = ",")
colnames(monthly_cpi) <- tolower(colnames(monthly_cpi))
monthly_cpi$cpi_year <- year(monthly_cpi$date)

yearly_cpi <- monthly_cpi %>%
        group_by(cpi_year) %>%
        summarise(cpi = mean(value)) %>%
                mutate(adjustment =
                               yearly_cpi$cpi /
                               yearly_cpi$cpi[yearly_cpi$cpi_year ==2011])
```

```
## Error in eval(expr, envir, enclos): object 'yearly_cpi' not found
```

```
stormdt$adjyear = year(stormdt$bgn_datetime)
stormdt$adjustment <- yearly_cpi$adjustment[match(stormdt$adjyear,
                                                  yearly_cpi$cpi_year)]
```

```
## Error in eval(expr, envir, enclos): object 'yearly_cpi' not found
```

```
stormdt$adjpropdmg <- stormdt$propdmg / stormdt$adjustment
```

```
## Error in `$<-.data.frame`(`*tmp*`, "adjpropdmg", value = numeric(0)): replacement has 0 rows, data has 20
```

```
stormdt$adjcropdmg <- stormdt$cropdmg / stormdt$adjustment
```

```
## Error in `$<-.data.frame`(`*tmp*`, "adjcropdmg", value = numeric(0)): replacement has 0 rows, data has 20
```

The data is reshaped to eliminate column variables unneeded for the purposes of the analysis using the select function of dplyr. The data is then filtered to remove records where fatalities, injuries, property damage, or crop damage amounted to zero.

Examining the unique values in the evtype variable returns nearly 1000 types of events, though has a set list of 48 event types for recording purposes. A variety of non-standard event types recorded and typo make up wide range of event types. The values stored in the evtype column were corrected to fit the standard 48 through cross referencing the NOAA storm database analysis handbook.

```
stormdt$evtype <- tolower(stormdt$evtype)
stormdt$evtype <- trimws(stormdt$evtype, which = "both")
stormdt$evtype <- gsub("  ", " ", stormdt$evtype)
stormdt$evtype <- gsub("^tstm wind.*", "thunderstorm wind", stormdt$evtype)
stormdt$evtype <- gsub("^thunderstorm.*", "thunderstorm wind",
                       stormdt$evtype)
stormdt$evtype <- gsub("^hurricane.*", "hurricane(typhoon)",
                       stormdt$evtype)
stormdt$evtype <- gsub("^typhoon.*", "hurricane(typhoon)", stormdt$evtype)
stormdt$evtype <- gsub("^high wind.*", "high wind", stormdt$evtype)
stormdt$evtype <- gsub("^gust.*", "marine high wind", stormdt$evtype)
stormdt$evtype <- gsub("^non.*", "high wind", stormdt$evtype)
stormdt$evtype <- gsub(".*fire.*", "wildfire", stormdt$evtype)
stormdt$evtype <- gsub(".*surf.*", "high surf", stormdt$evtype)
stormdt$evtype <- gsub(".*astronomical high tide.*", "high surf",
                       stormdt$evtype)
stormdt$evtype <- gsub(".*coastal flooding.*", "coastal flood",
                       stormdt$evtype)
stormdt$evtype <- gsub("gradient wind", "tropcial depression",
                       stormdt$evtype)
stormdt$evtype <- gsub("landspout", "dust devil", stormdt$evtype)
stormdt$evtype <- gsub("lake effect snow", "lake-effect snow",
                       stormdt$evtype)
stormdt$evtype <- gsub("tropcial depression", "tropical depression",
                       stormdt$evtype)
stormdt$evtype <- gsub("marine tstm wind", "marine thunderstorm wind",
                       stormdt$evtype)
stormdt$evtype <- gsub("glaze", "freezing fog", stormdt$evtype)
stormdt$evtype <- gsub("^tropical storm wind.*", "tropical storm",
                       stormdt$evtype)
stormdt$evtype <- gsub(".*rain.*", "heavy rain", stormdt$evtype)
stormdt$evtype <- gsub(".*microburst.*", "thunderstorm wind",
                       stormdt$evtype)
stormdt$evtype <- gsub(".*whirlwind.*", "thunderstorm wind",
                       stormdt$evtype)
stormdt$evtype <- gsub(".*downburst.*", "thunderstorm wind",
                       stormdt$evtype)
stormdt$evtype <- gsub(".*small hail.*", "hail", stormdt$evtype)
stormdt$evtype <- gsub(".*blowing dust.*", "dust storm", stormdt$evtype)
stormdt$evtype <- gsub(".*fog.*", "dense fog", stormdt$evtype)
stormdt$evtype <- gsub(".*coastalstorm.*", "tropical storm",
                       stormdt$evtype)
stormdt$evtype <- gsub(".*coastal storm.*", "tropical storm",
                       stormdt$evtype)
stormdt$evtype <- gsub(".*strong winds.*", "strong wind", stormdt$evtype)
stormdt$evtype <- gsub(".*heavy snow shower.*", "heavy snow",
                       stormdt$evtype)
```

```
stormdt$evtype <- gsub(".*storm surge.*", "storm surge/tide",
                       stormdt$evtype)
stormdt$evtype <- gsub(".*warm weather.*", "heat", stormdt$evtype)
stormdt$evtype <- gsub(".*winds.*", "high wind", stormdt$evtype)
stormdt$evtype <- gsub("^wind$", "high wind", stormdt$evtype)
stormdt$evtype <- gsub(".*excessive snow*", "blizzard", stormdt$evtype)
stormdt$evtype <- gsub("^snow$", "heavy snow", stormdt$evtype)
stormdt$evtype <- gsub(".*late season snow.*", "heavy snow",
                       stormdt$evtype)

flood <- c("flood", "dam", "ice jam", "fld", "river flood",
           "lakeshore flood", "river flooding", "high water")
for(f in flood){stormdt$evtype <- gsub(paste(".*", f, ".*", sep=""),
                                       "flood", stormdt$evtype)}

frost <- c("freeze", "hard freeze", "agricultural freeze", "frost")
for(f in frost){stormdt$evtype <- gsub(paste(".*", f, ".*", sep=""),
                                    "frost/freeze", stormdt$evtype)}

winter <- c("freezing", "black ice", "icy", "ice roads",
                   "ice on road", "light snow", "snow squall", "wintry",
                   "winter", "snow and ice", "rain/snow", "cold and snow",
                   "mixed precip", "falling snow/ice", "blowing snow")

for(w in winter){stormdt$evtype <- gsub(paste(".*", w, ".*", sep=""),
                                               "winter weather", stormdt$evtype)}

heat <- c("heat wave", "record heat", "record excessive heat",
          "hyperthermia", "unseasonably warm")
for(h in heat){stormdt$evtype <- gsub(paste(".*", h, ".*", sep = ""),
                                      "excessive heat", stormdt$evtype)}

xcold <- c("extreme cold", "unseasonable cold", "extended cold",
           "unseasonably cold", "hypothermia", "extreme windchill")
for(c in xcold){stormdt$evtype <- gsub(paste(".*", c, ".*", sep = ""),
                                      "extreme cold/wind chill",
                                      stormdt$evtype)}

cold <- c("cold", "cold temperature", "cold weather")
for(c in cold){stormdt$evtype <- gsub(paste(".*", c, ".*", sep=""),
                                      "cold/wind chill", stormdt$evtype)}

seas <- c("heavy seas", "high seas", "rough seas", "rip currents",
          "wind and wave", "rogue wave", "high swells", "marine accident")
for(s in seas){stormdt$evtype <- gsub(paste(".*", s, ".*", sep=""),
                                      "high surf",
                                      stormdt$evtype)}

debris <- c("mudslide", "mud slide", "mudslides", "rock slide",
            "landslide","landslides", "erosion","landslump", "debris")
for(d in debris){stormdt$evtype <- gsub(paste(".*", d, ".*", sep = ""),
                                        "debris flow", stormdt$evtype)}
```

Decision was made to eliminate evtype values that did not have clear relationship to standard 48; these events("other" amd "drowning") were only present in 1 record respectively.

```
stormdt <- stormdt[stormdt$evtype != "other" & stormdt$evtype != "drowning",]
```

For presentation's sake, names of event types are capitalized.

```
evtype_capitalize <- function(x) {
        s <- strsplit(x, " ")[[1]]
        paste(toupper(substring(s, 1,1)), substring(s, 2),
              sep="", collapse=" ")
}

stormdt$evtype <- sapply(stormdt$evtype, evtype_capitalize)
```

Analysis

All visualizations for storm data will be generated using the ggplot2 package from Hadley Wickman, begining with a histogram showing the event type counts for the time frame of January 1, 1996 to November 30, 2011.

Table reflecting the values of top 6 events by count:

```
library(ggplot2);
evtype_hist <- qplot(stormdt$evtype,
                     main = "Event Type Frequency accross USA",
                     xlab = "Event Type",
                     ylab = "Count")

knitr::kable(head(count(stormdt, evtype) %>% arrange(desc(n)), n=10),
             caption = "Event Counts 1996-2011", col.names = c("Event",
                                                                "Count"))
```

Table 1: Event Counts 1996-2011

Event	Count
Thunderstorm Wind	105453
Flood	29527
Hail	22690
Tornado	12365
Lightning	11151
High Wind	5476
Strong Wind	3414
Winter Weather	2215
Wildfire	1229
Heavy Rain	1086

Examining impact on public health

For the purposes of calculation, a timeframe variable is created using the difftime function with units set to "weeks", the value of function then divded by 52.25 (accounting for the impact of leap years) to set the unit it years. The returned value is then coerced to a numeric value for computation.

A new data.table is created by grouping the data by event type, creating summary statistics for total fatalities, mortatity by event occurence, number of occurences, and finally occurences of a given event type by year (calculated with timeframe variable). ggplot() is employed to visualize total fatalities by event.

```r
timeframe <- as.numeric(difftime(max(stormdt$bgn_date),
                                 min(stormdt$bgn_date),
                                 units = "days") / 365.25)

event_fatalities <- stormdt %>%
        group_by(evtype) %>%
        summarise(totalfatalities = sum(fatalities),
                  peryear = totalfatalities / timeframe,
                  occurences = n(),
                  fatalitiesperoccurence = totalfatalities / n()) %>%
        arrange(desc(totalfatalities))

ggplot(event_fatalities, aes(evtype, peryear)) +
        geom_bar(stat = "sum") +
        labs(title = "Fatalities per Year by Event Type",
             x = "Event Type", y = "Fatalities") +
        theme_minimal() +
        theme(axis.text.x = element_text(angle = 45, hjust = 1),
              plot.title = element_text(hjust = 0.5),
              legend.position = "none")
```

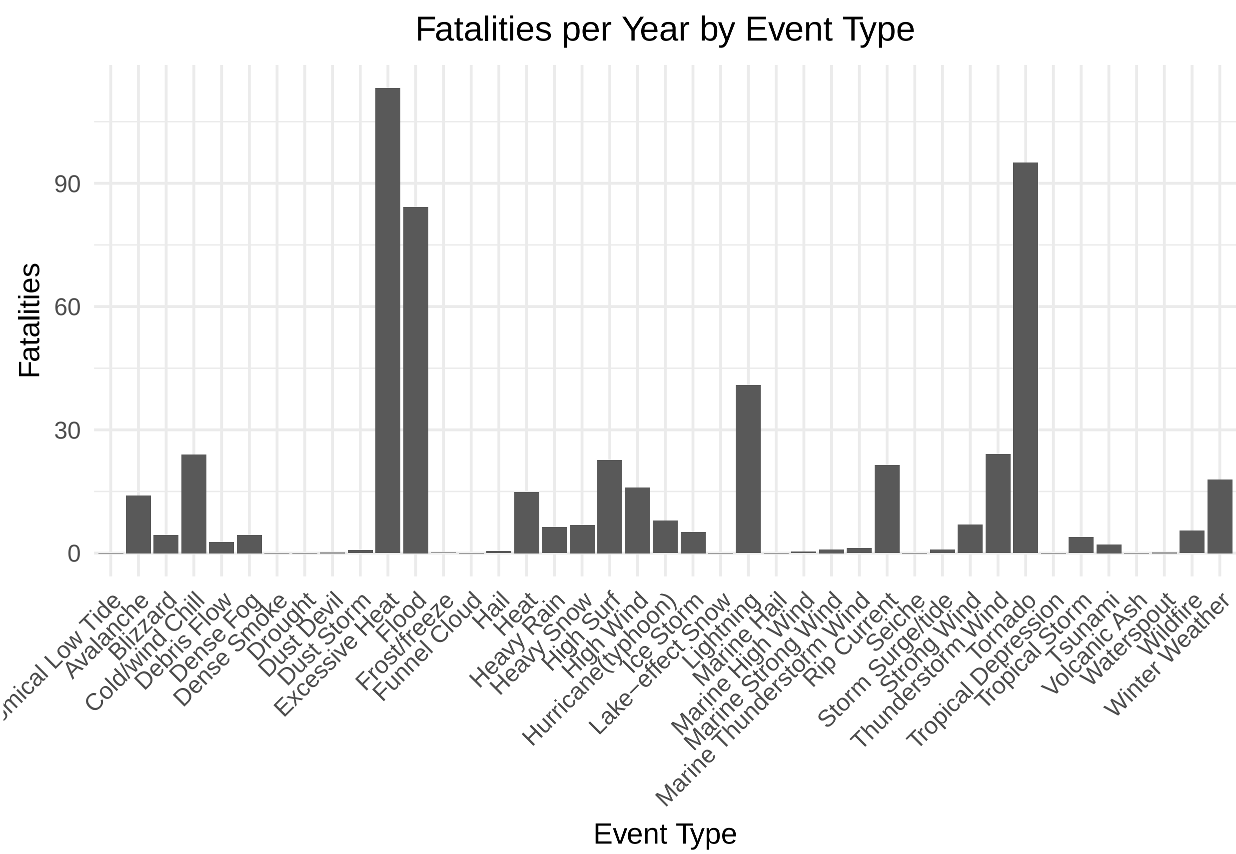


Table below expands on the data visualized above:

```r
knitr::kable(head(event_fatalities, n=10),
             caption = "Total Fatalities by Event",
             col.names = c("Event", "Total Fatalities",
                          "Fatalities per Year", "Occurences (since 1996)",
                          "Fatalities per Occurence"))
```

Table 2: Total Fatalities by Event

Event	Total Fatalities	Fatalities per Year	Occurences (since 1996)	Fatalities per Occurence
Excessive Heat	1800	113.13887	693	2.5974026
Tornado	1511	94.97380	12365	0.1221998
Flood	1340	84.22561	29527	0.0453822
Lightning	650	40.85570	11151	0.0582907
Thunderstorm Wind	383	24.07344	105453	0.0036319
Cold/wind Chill	382	24.01058	420	0.9095238
High Surf	361	22.69063	475	0.7600000
Rip Current	340	21.37068	364	0.9340659
Winter Weather	285	17.91366	2215	0.1286682
High Wind	254	15.96515	5476	0.0463842

```
event_fatalities <- stormdt %>%
        group_by(evtype) %>%
        summarise(totalfatalities = sum(fatalities),
                  peryear = totalfatalities / timeframe,
                  occurences = n(),
                  fatalitiesperoccurence = totalfatalities / n()) %>%
        arrange(desc(fatalitiesperoccurence))

knitr::kable(head(event_fatalities, n=7),
             caption = "Total Fatalities by Event",
             col.names = c("Event", "Total Fatalities",
                           "Fatalities per Year", "Occurences (since 1996)",
                           "Fatalities per Occurence"))
```

Table 3: Total Fatalities by Event

Event	Total Fatalities	Fatalities per Year	Occurences (since 1996)	Fatalities per Occurence
Excessive Heat	1800	113.138875	693	2.5974026
Tsunami	33	2.074213	14	2.3571429
Heat	237	14.896619	165	1.4363636
Rip Current	340	21.370676	364	0.9340659
Cold/wind Chill	382	24.010583	420	0.9095238
Avalanche	223	14.016649	264	0.8446970
High Surf	361	22.690630	475	0.7600000

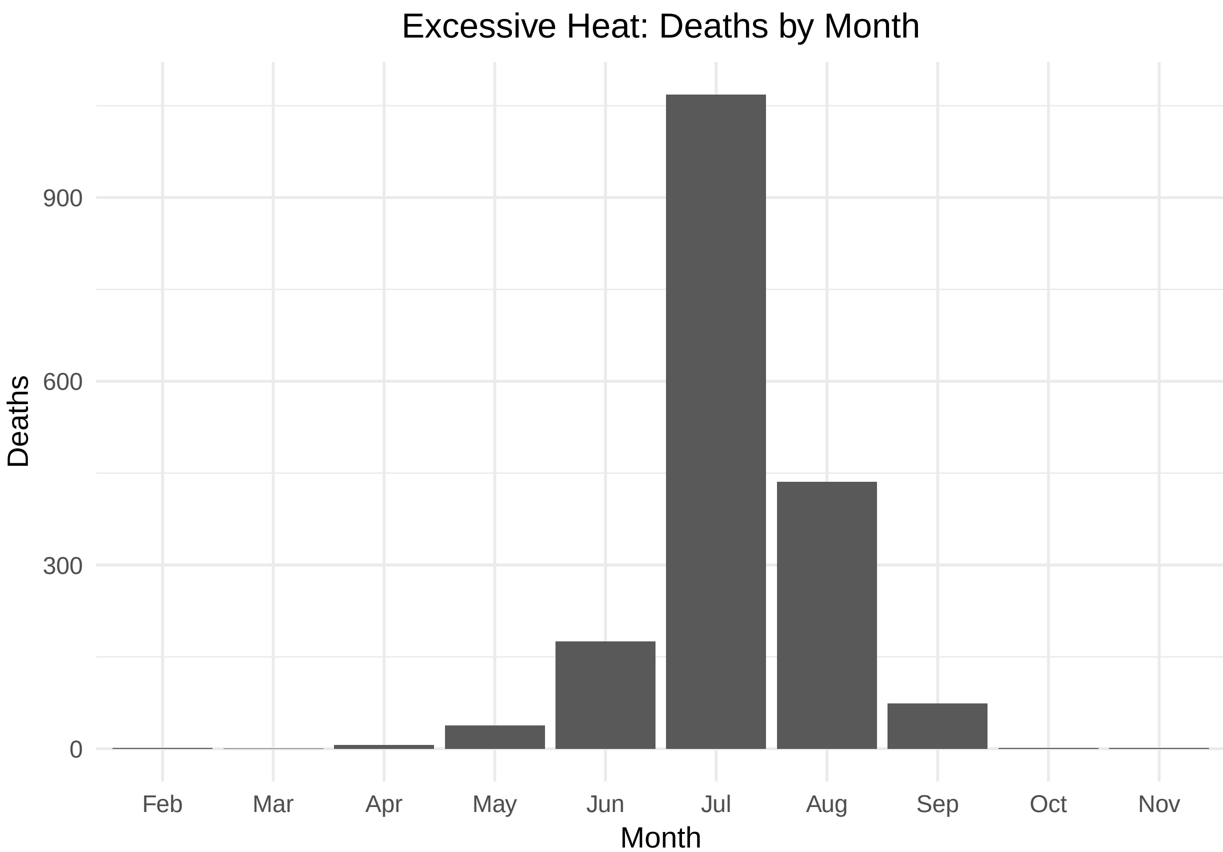
By the visualization and table above, the 3 event types which have the greatest impact of mortality are: Excessive Heat, Tornado, and Flood. The impact of these events will be further examined.

PublicHealth 1)Excessive Heat

- most impacted states
- count by month

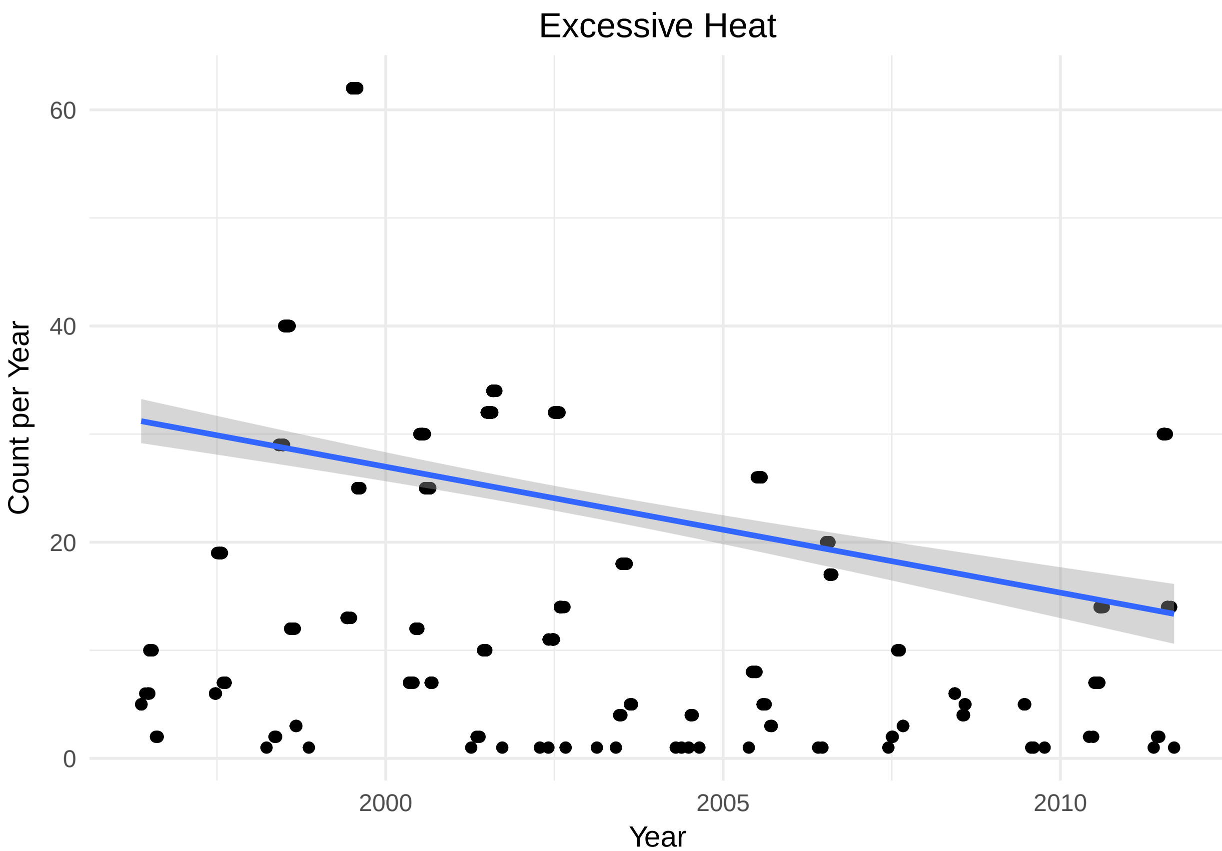
```
excessive_heat <- stormdt[which(stormdt$evtype == "Excessive Heat"),]
```

```r
excessive_heat_vis <- excessive_heat %>%
        mutate(month = format(bgn_datetime, "%m")) %>%
        group_by(month) %>%
        summarise(totalfatalities = sum(fatalities))

# excessive_heat_vis$month <- month.abb[as.numeric(excessive_heat_vis$month)]

ggplot(excessive_heat_vis,aes(month, totalfatalities),
       color = aes(time_zone)) +
        geom_col() +
        labs(title = "Excessive Heat: Deaths by Month",
             x = "Month", y = "Deaths") +
        scale_x_discrete(labels = month.abb[as.numeric(
                                  excessive_heat_vis$month)]) +
        theme_minimal() +
        theme(plot.title = element_text(hjust = 0.5))
```




```r
excessive_heat_count <- excessive_heat %>%
        mutate(month = format(bgn_datetime, "%m")) %>%
        group_by(adjyear, month) %>%
        mutate(count = n())

ggplot(excessive_heat_count, aes(bgn_datetime, count)) +
        geom_point() +
        geom_smooth(method = lm) +
        labs(title = "Excessive Heat",
             x = "Year", y = "Count per Year") +
```

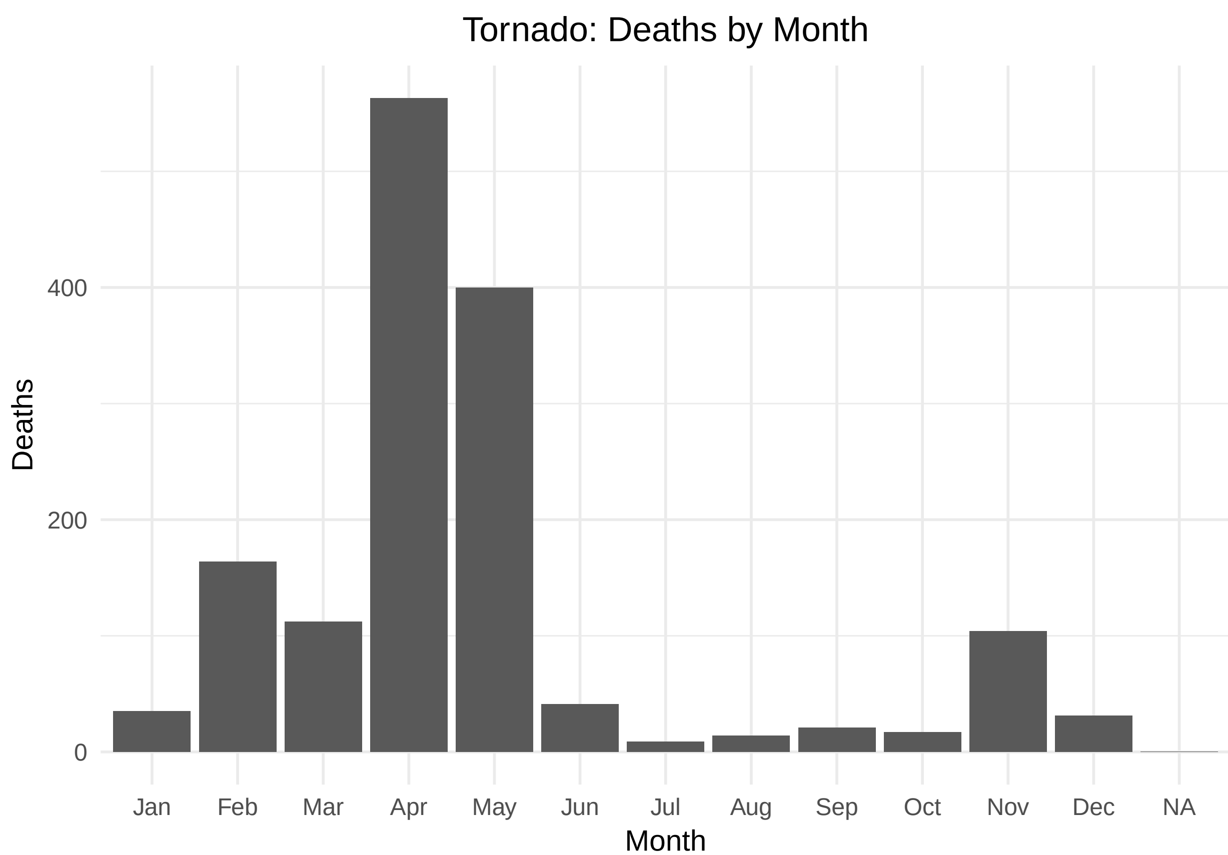
```
        theme_minimal() +
        theme(plot.title = element_text(hjust = .5))
```



Public Health 2)Tornado

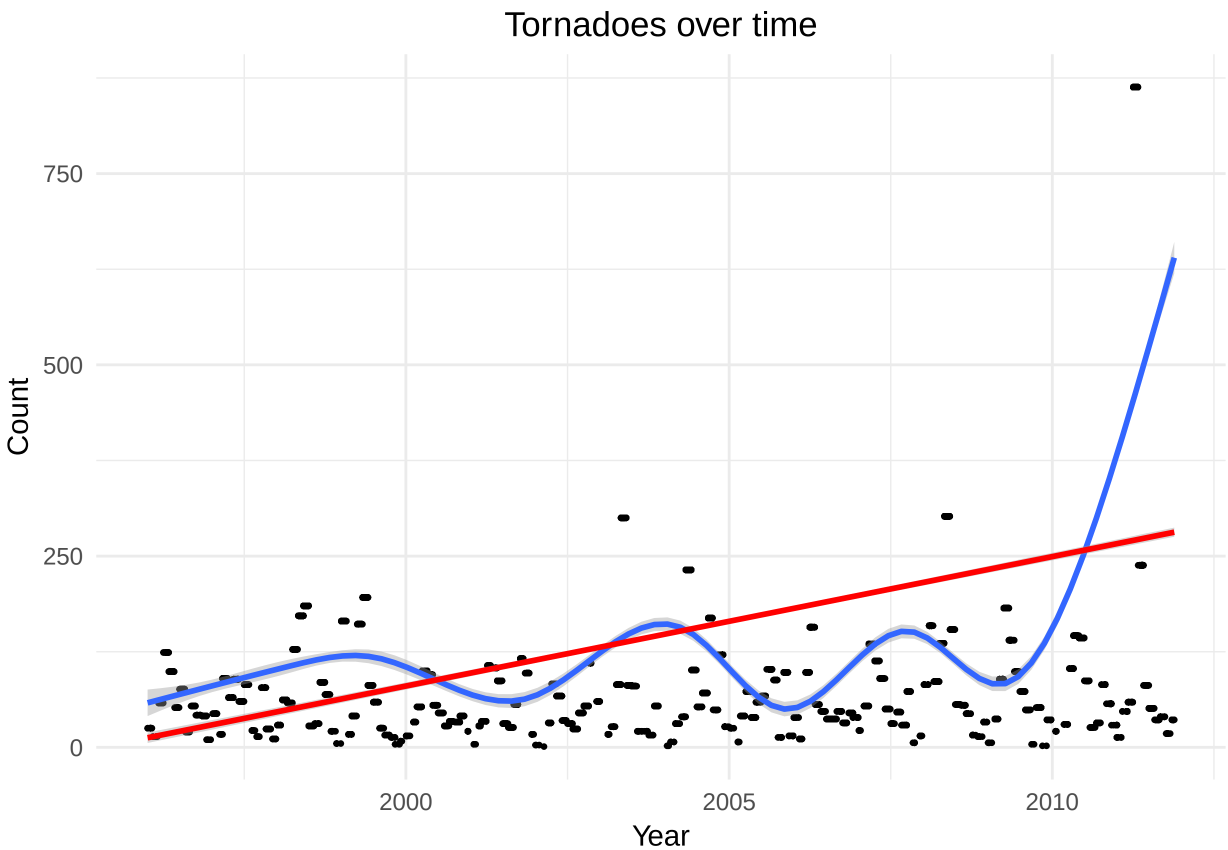
-summary stats: duration, widtch, avg damage by F type, avg damage by width, avg damage by length, state with most torandos(damage by year) -visualizaitons:

```
tornado <- stormdt[which(stormdt$evtype == "Tornado"),]

tornado_vis <-
        tornado %>%
        mutate(month = format(bgn_datetime, "%m")) %>%
        group_by(month) %>%
        summarise(monthfatalities = sum(fatalities))

ggplot(tornado_vis,aes(month, monthfatalities)) +
        geom_col() +
        labs(title = "Tornado: Deaths by Month",
             x = "Month", y = "Deaths") +
        scale_x_discrete(labels = month.abb[as.numeric(tornado_vis$month)]) +
        theme_minimal() +
        theme(plot.title = element_text(hjust = 0.5))
```




```
tornado_counts <-
        tornado %>%
        group_by(format(bgn_datetime, "%Y%m")) %>%
        mutate(count = n())

ggplot(tornado_counts, aes(bgn_datetime, count)) +
        geom_point(size = .5) +
        geom_smooth() +
        geom_smooth(method = lm, color = "red") +
        labs(title = "Tornadoes over time",
             x = "Year", y = "Count") +
        theme_minimal() +
        theme(plot.title = element_text(hjust = .5))
```

```
## `geom_smooth()` using method = 'gam'

## Warning: Removed 8 rows containing non-finite values (stat_smooth).

## Warning: Removed 8 rows containing non-finite values (stat_smooth).

## Warning: Removed 8 rows containing missing values (geom_point).
```




```r
tornado_zone_stats <-
        tornado %>%
        group_by(time_zone) %>%
        summarise(count = n()) %>%
        arrange(desc(count))

knitr::kable(tornado_zone_stats, caption = "Torandos by Region",
             col.names = c("Region", "Tornados"))
```

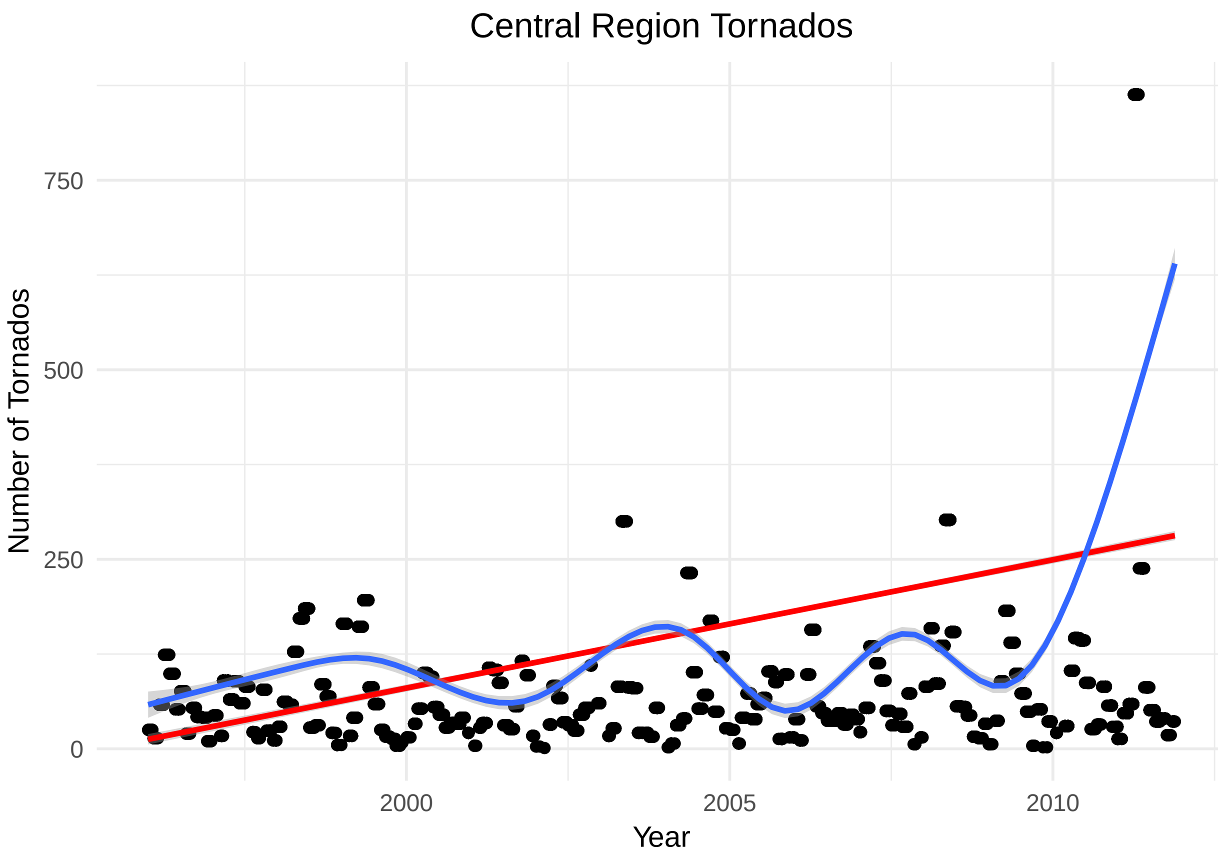
Table 4: Torandos by Region

Region	Tornados
CST	8048
EST	3833
MST	328
PST	145
AST	9
HST	2

```r
cst_tornado <- tornado[which(tornado$time_zone == "CST"),]

cst_tornado_counts <-
        cst_tornado %>%
        group_by(format(bgn_datetime, "%Y")) %>%
        mutate(count = n())
```

```r
ggplot(tornado_counts, aes(bgn_datetime, count)) +
        geom_point() +
        geom_smooth(method = lm, color = "red") +
        geom_smooth() +
        labs(title = "Central Region Tornados",
             x = "Year", y = "Number of Tornados") +
        theme_minimal() +
        theme(plot.title = element_text(hjust = 0.5))
```

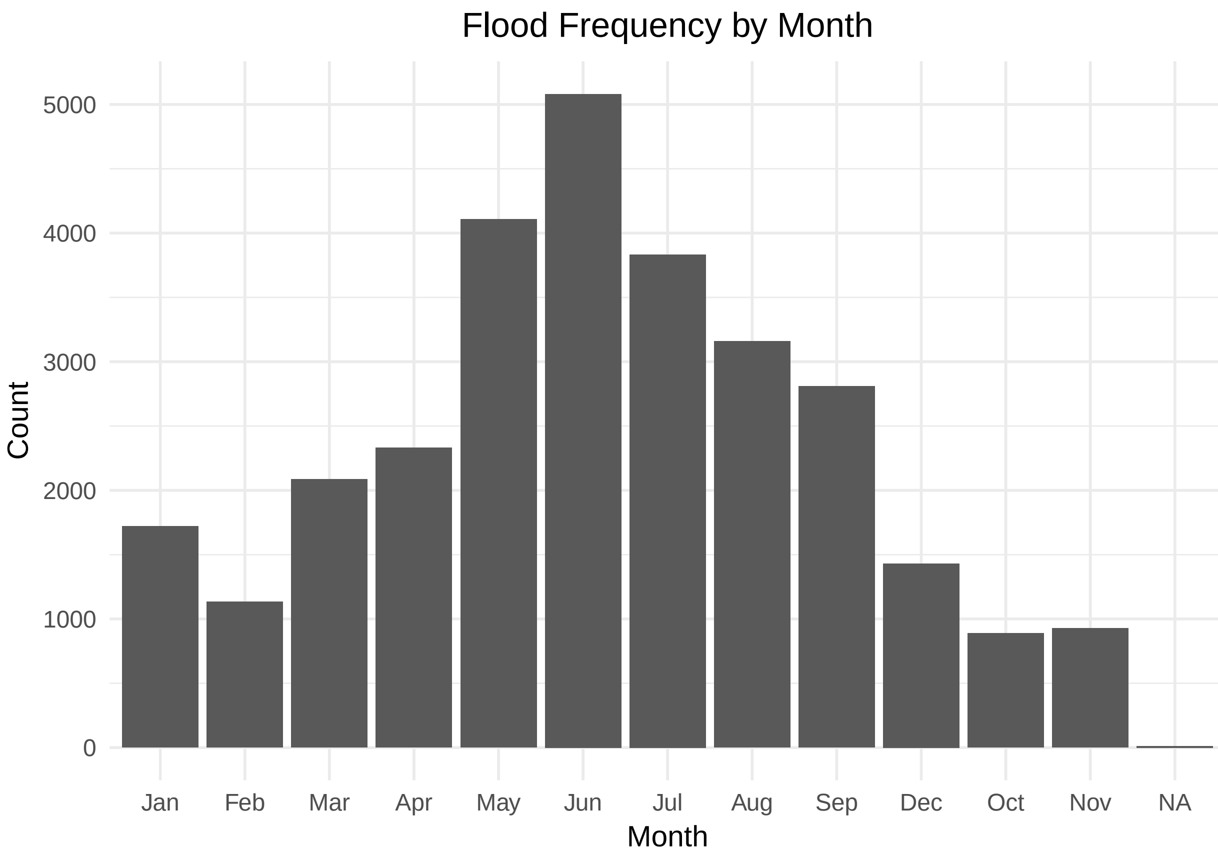
```
## Warning: Removed 8 rows containing non-finite values (stat_smooth).

## `geom_smooth()` using method = 'gam'

## Warning: Removed 8 rows containing non-finite values (stat_smooth).

## Warning: Removed 8 rows containing missing values (geom_point).
```



Flood

```r
flood <- stormdt[which(stormdt$evtype == "Flood"),]
flood_vis <- flood %>%
        mutate(month = format(bgn_datetime, "%m")) %>%
        group_by(month)


#Gotta fix the labeling issue
ggplot(flood_vis,aes(month)) +
        geom_bar() +
```

```
        labs(title = "Flood Frequency by Month",
             x = "Month", y = "Count") +
        scale_x_discrete(labels = month.abb[unique(as.numeric(flood_vis$month))]) +
        theme_minimal() +
        theme(plot.title = element_text(hjust = 0.5))
```



Examining Injuries by event type:

Summary dataset is generated through grouping data by event type, summarising total injuries, injuries sustained by occurence of event, number of event occurences, and occurences by year. ggplot2 is used to visualize the total number of injuries by event type.

```
event_injuries <- stormdt %>%
        group_by(evtype) %>%
        summarise(totalinjuries = sum(injuries),
                  injuryperyear = totalinjuries / timeframe,
                  injuryperoccurence = totalinjuries / n(),
                  occurences = n(),
                  occurencesperyear = occurences/timeframe) %>%
        arrange(desc(totalinjuries))

ggplot(event_injuries, aes(evtype, injuryperyear)) +
        geom_bar(stat = "sum") +
        labs(title = "Injuries by Event Type since 1996",
             x = "Event Type", y = "Total Injuries") +
        theme_minimal() +
        theme(axis.text.x = element_text(angle = 45, hjust = 1),
```

```
            plot.title = element_text(hjust = 0.5),
            legend.position = "none")
```

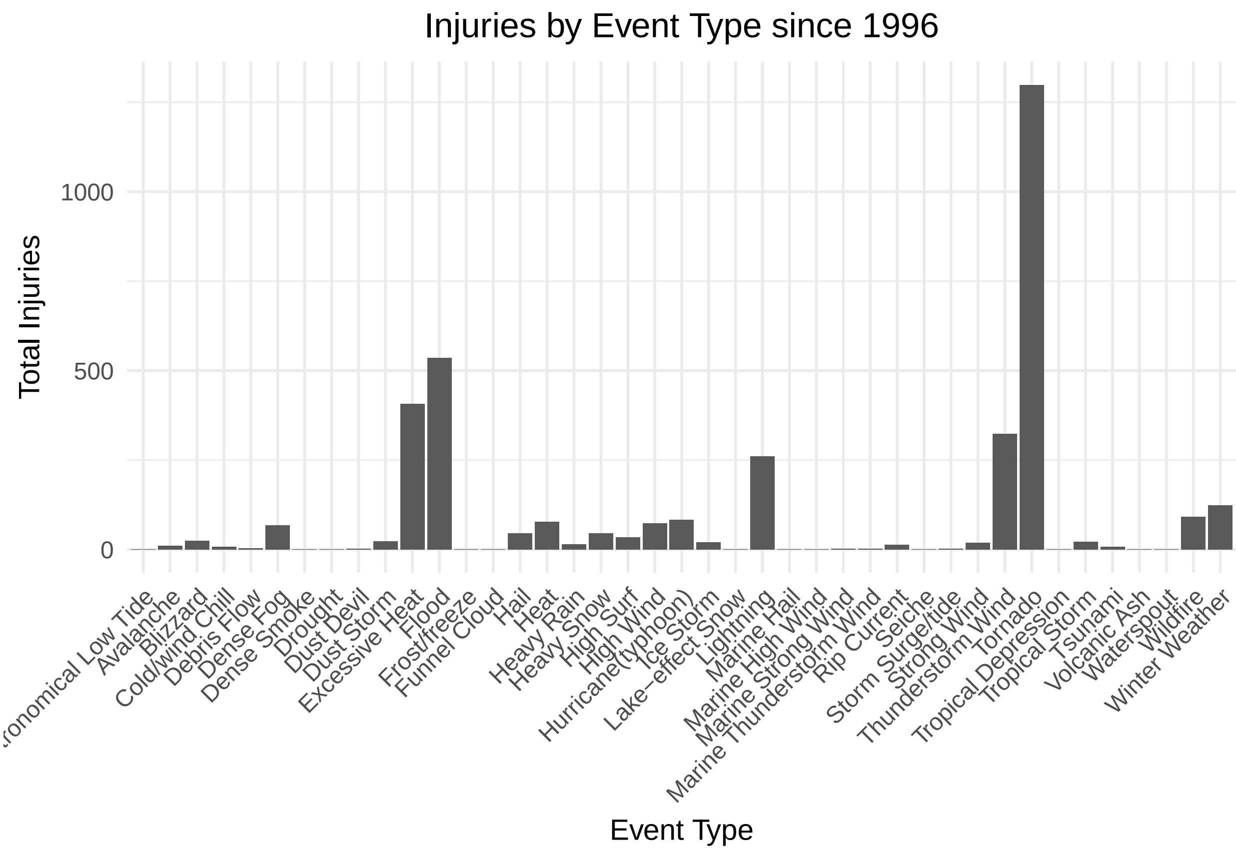
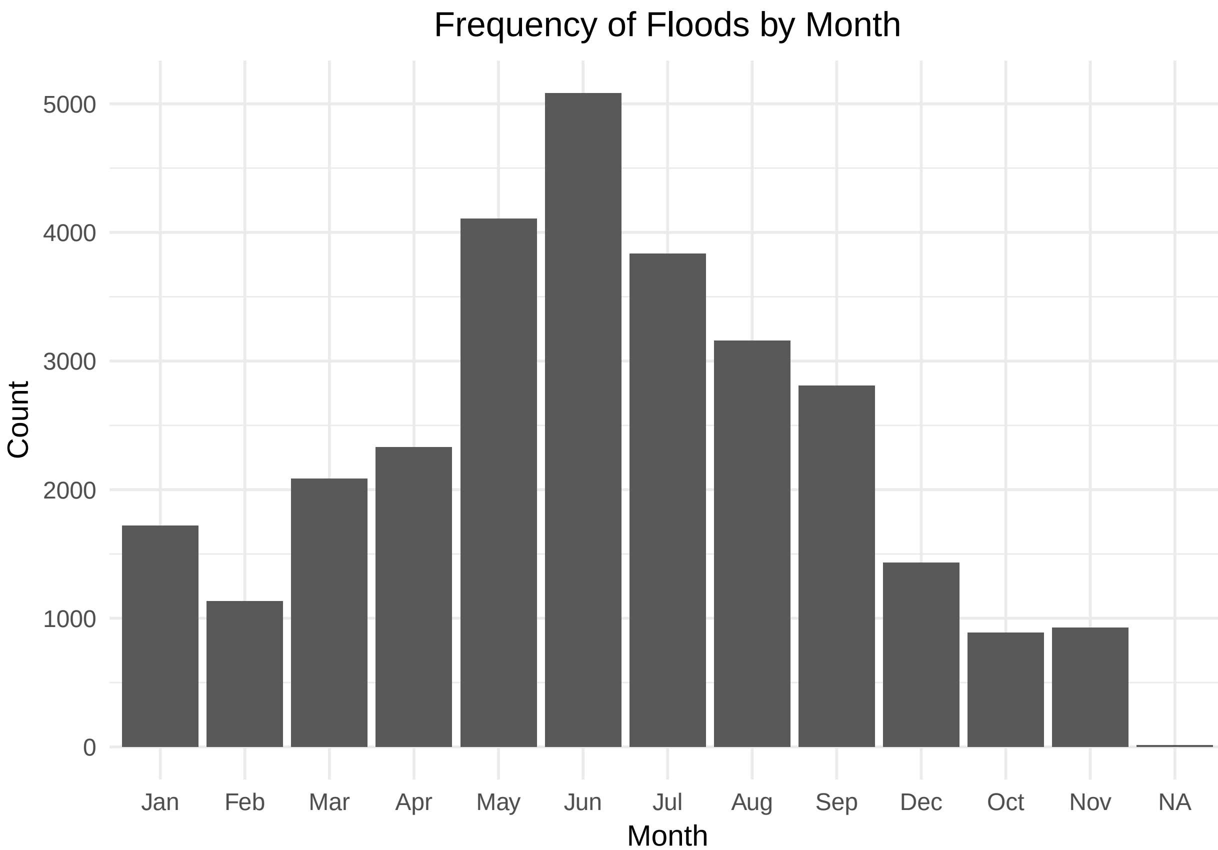


Table below expands on the data visualized above:

```
knitr::kable(head(event_injuries), caption = "Total Injuries by Event since 1996",
             col.names = c("Event", "injuryperyear, Total Injuries",
                           "Injuries per Occurence", "Occurences",
                           "Occurences per Year"))
```

```
## Error in `colnames<-`(`*tmp*`, value = c("Event", "injuryperyear, Total Injuries", : length of 'dimnames'
```

```
flood <- stormdt[which(stormdt$evtype == "Flood"),]
flood_vis <- flood %>%
        mutate(month = format(bgn_datetime, "%m")) %>%
        group_by(month) %>%
        mutate(count = n())

ggplot(flood_vis,aes(month)) +
        geom_bar() +
        labs(title = "Frequency of Floods by Month",
             x = "Month", y = "Count") +
        scale_x_discrete(labels = month.abb[unique(as.numeric(flood_vis$month))]) +
        theme_minimal() +
        theme(plot.title = element_text(hjust = .5))
```



Economic Impact

Property Damage

```
event_propdmg <- stormdt %>%
        group_by(evtype) %>%
        summarise(totalpropdmg = sum(propdmg),
                  damageperyear = totalpropdmg / timeframe,
                  damageperoccurence = totalpropdmg / n(),
                  occurences = n(),
                  occurencesperyear = occurences/timeframe) %>%
        mutate(totalpropdmg = totalpropdmg / 1E9,
               damageperoccurence = damageperoccurence / 1E6,
               damageperyear = damageperyear / 1E9) %>%
        arrange(desc(damageperyear))

ggplot(event_propdmg, aes(evtype, damageperyear)) +
        geom_bar(stat = "sum") +
        labs(title = "Property Damage per Year",
             x = "Event Type", y = "Total Damage (in Billions)") +
        theme_minimal() +
        theme(axis.text.x = element_text(angle = 45, hjust = 1),
              plot.title = element_text(hjust = 0.5),
              legend.position = "none")
```

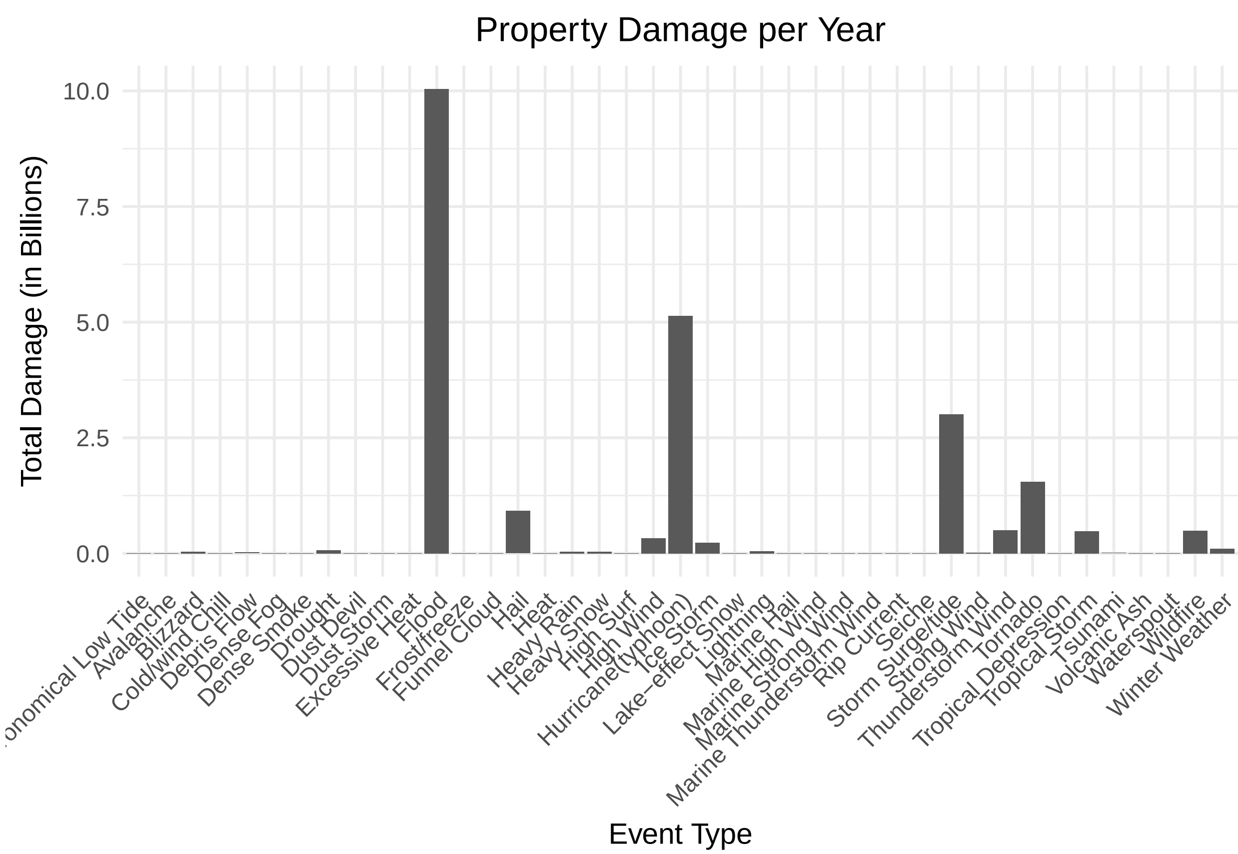


Table below expands upon property damage by event type

```
knitr::kable(head(event_propdmg, n=10),
             caption = "Property Damage by Event since 1996",
             col.names = c("Event", "Damage (in Billions)", "Damage per Year",
                           "Average Damage (in Millions)","Occurences",
                           "Occurences per Year"))
```

Table 5: Property Damage by Event since 1996

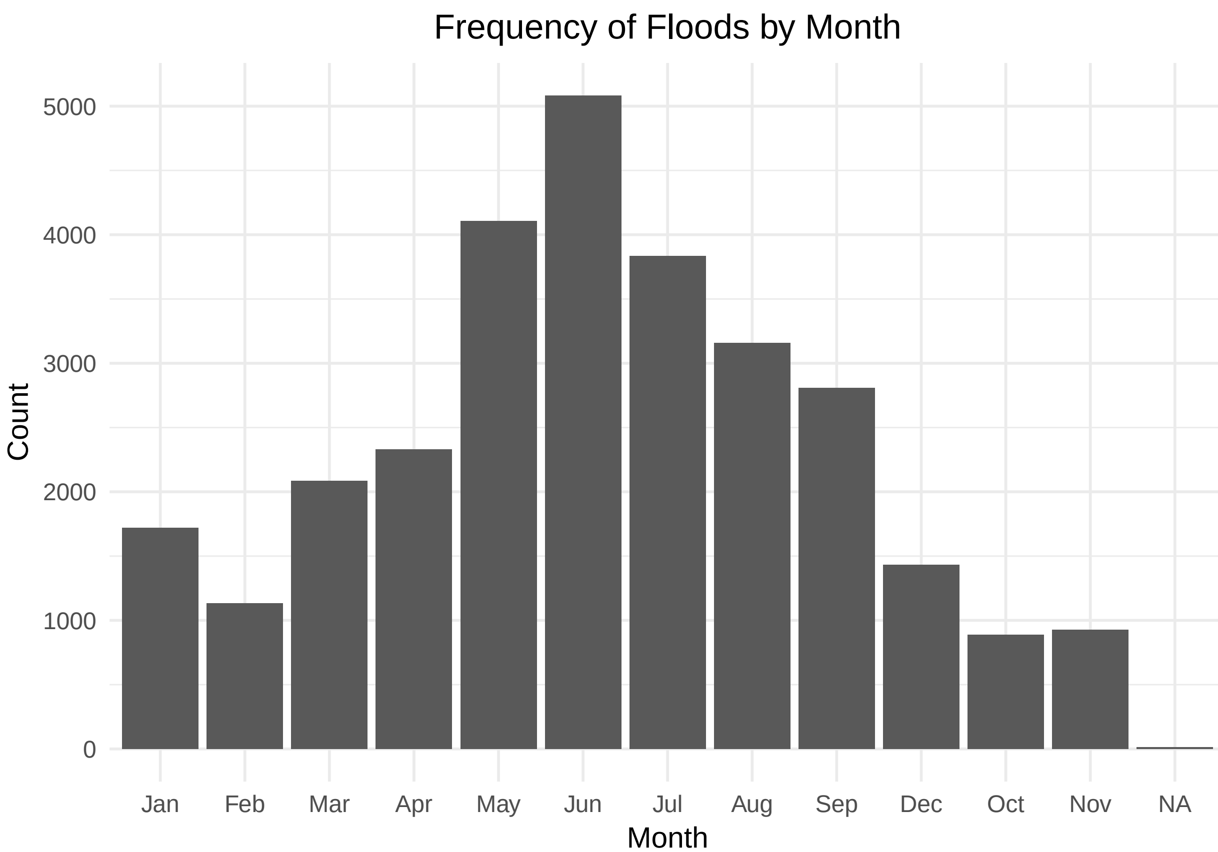
Event	Damage (in Billions)	Damage per Year	Average Damage (in Millions)	Occurences	Occurence
Flood	159.774855	10.0426374	5.4111442	29527	1
Hurricane(typhoon)	81.718889	5.1364351	392.8792741	208	
Storm Surge/tide	47.834724	3.0066482	221.4570556	216	
Tornado	24.616906	1.5472939	1.9908537	12365	
Hail	14.595213	0.9173811	0.6432443	22690	1
Thunderstorm Wind	7.915343	0.4975183	0.0750604	105453	6
Wildfire	7.760450	0.4877825	6.3144422	1229	
Tropical Storm	7.642526	0.4803704	18.4157242	415	
High Wind	5.250713	0.3300332	0.9588592	5476	
Ice Storm	3.642249	0.2289333	5.7721851	631	

Flood

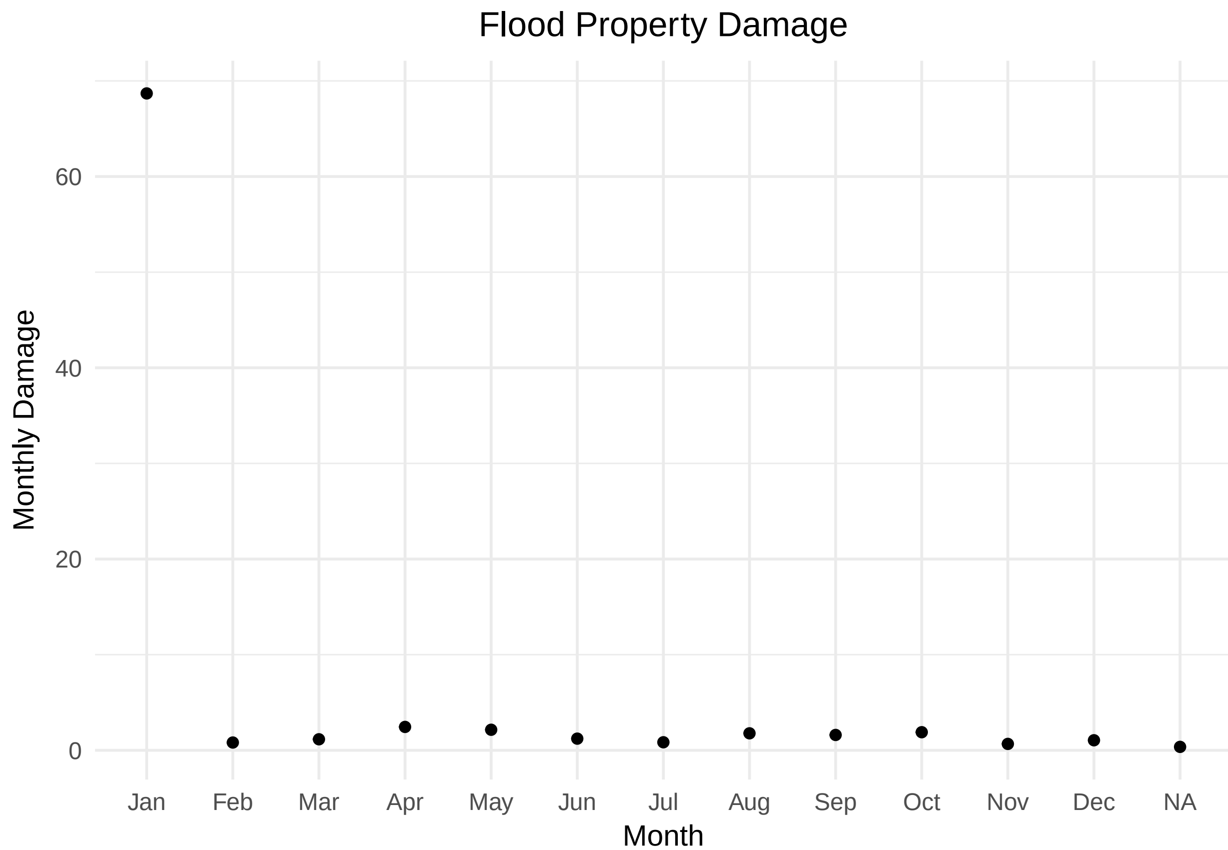
```
flood <- stormdt[which(stormdt$evtype == "Flood"),]
flood_propdmg <-
```

```r
        flood %>%
        mutate(month = format(bgn_datetime, "%m")) %>%
        group_by(month)
```

```r
ggplot(flood_vis,aes(month)) +
        geom_bar() +
        labs(title = "Frequency of Floods by Month",
             x = "Month", y = "Count") +
        scale_x_discrete(labels = month.abb[unique(as.numeric(flood_vis$month))]) +
        theme_minimal() +
        theme(plot.title = element_text(hjust = .5))
```




```r
flood_propdmg_avg <-
        flood_propdmg %>%
        summarise(monthlydmgavg = mean(propdmg)) %>%
        mutate(monthlydmgavg = monthlydmgavg / 1E6)

ggplot(flood_propdmg_avg, aes(month, monthlydmgavg)) +
        geom_point() +
        labs(title = "Flood Property Damage",
             x = "Month", y = "Monthly Damage") +
        theme_minimal() +
        theme(plot.title = element_text(hjust = .5)) +
        scale_x_discrete(labels = month.abb[as.numeric(flood_propdmg_avg$month)])
```



Hurricane

#####Storm Surge

Crop Damage

```
event_cropdmg <- stormdt %>%
        group_by(evtype) %>%
        summarise(totalcropdmg = sum(cropdmg),
                  cropdmgperyear = totalcropdmg / timeframe,
                  cropdmgperoccurence = totalcropdmg/n(),
                  occurences = n(),
                  occurencesperyear = occurences / timeframe) %>%
        mutate(totalcropdmg = totalcropdmg / 1E9,
               cropdmgperoccurence = cropdmgperoccurence/1E6,
               cropdmgperyear = cropdmgperyear / 1E6) %>%
        arrange(desc(cropdmgperyear))

ggplot(event_cropdmg, aes(evtype, totalcropdmg), fill = evtype) +
        geom_bar(stat = "sum") +
        labs(title = "Total Crop Damage by Event Type",
             x = "Event Type", y = "Total Damage (Billions)") +
        theme_minimal() +
        theme(axis.text.x = element_text(angle = 45, hjust = 1),
              plot.title = element_text(hjust = 0.5),
              legend.position = "none")
```

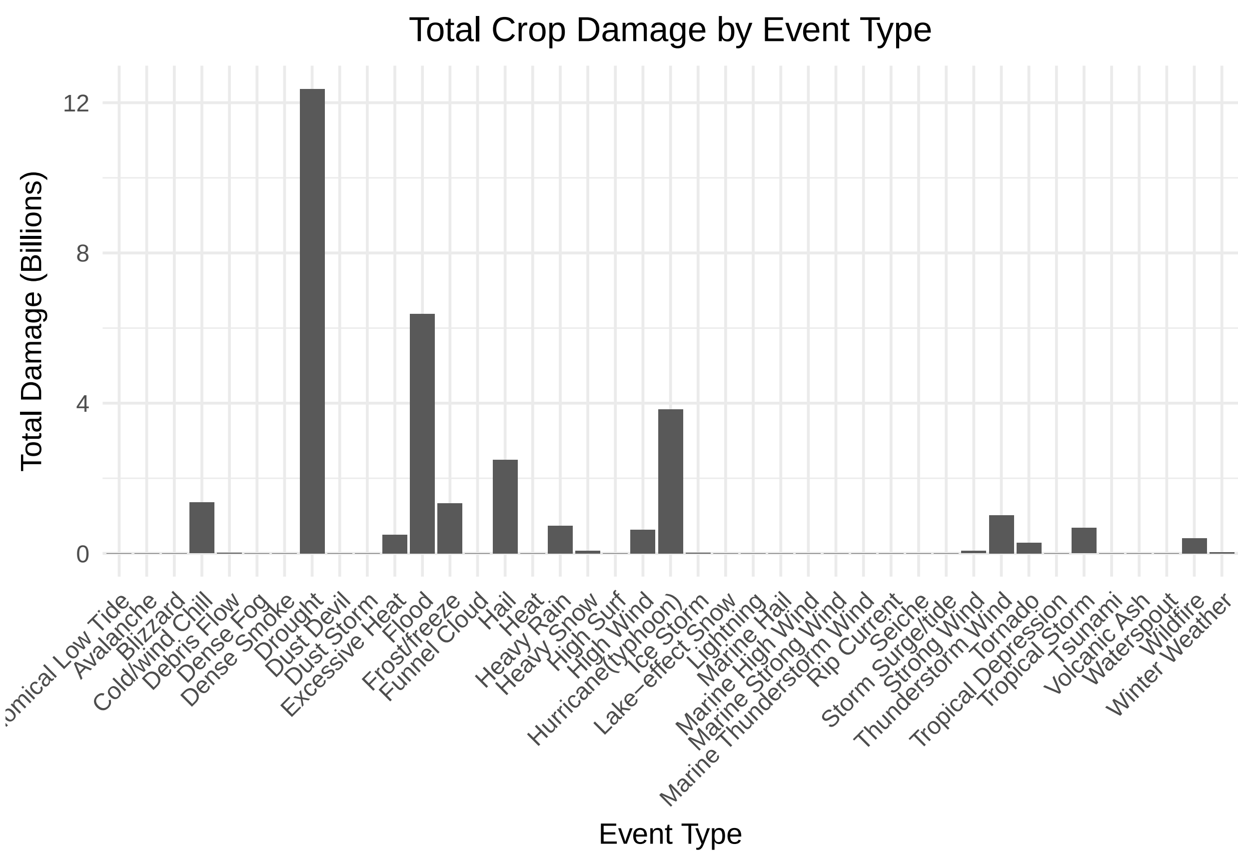


Table below expands upon crop damage.

```
knitr::kable(head(event_cropdmg, n=10), caption = "Crop Damage by Event",
             col.names = c("Event", "Total Damage (in Billions)",
                          "Damage per Year (Millions)",
                          "Damage per Occurence (in Millions)",
                          "Occurences", "Occurences per Year"))
```

Table 6: Crop Damage by Event

Event	Total Damage (in Billions)	Damage per Year (Millions)	Damage per Occurence (in Millions)
Drought	12.3675660	777.36250	47.9363023
Flood	6.3821932	401.15231	0.2161477
Hurricane(typhoon)	3.8401078	241.36971	18.4620567
Hail	2.4968224	156.93760	0.1100407
Cold/wind Chill	1.3567655	85.27940	3.2303940
Frost/freeze	1.3346310	83.88814	9.7418321
Thunderstorm Wind	1.0169576	63.92080	0.0096437
Heavy Rain	0.7384198	46.41333	0.6799446
Tropical Storm	0.6777110	42.59748	1.6330386
High Wind	0.6338613	39.84131	0.1157526

Drought

Flood

Hurricane

Results