Health & Economic Impact of Severe Weather Conditions in the United States

Eshitha Kumar

August 31, 2016

## Synopsis

This project explores the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database to track 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

## Questions

Define key questions that would help a government or municipal manager preparing for severe weather events prioritize resources.

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

## Data Analysis

**Extract and Load Data**

Load necessary libraries, unzip relevant files and read data into a data frame

# If file is not present in the current directory, download file  
  
if(!"StormData.csv.bz2" %in% dir())  
 download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2", destfile = "StormData.csv.bz2")  
  
# Unzip compressed folder and read storm data into a data frame  
storm\_data <- read.csv("StormData.csv.bz2", colClasses = "character")

**Exploratory Data Analysis**

nrow(storm\_data)

## [1] 902297

ncol(storm\_data)

## [1] 37

dim(storm\_data)

## [1] 902297 37

str(storm\_data)  
summary(storm\_data)

**Filter Data**

Filter the data set to extract columns relevant for analysis. This includes

1. BGN\_DATE - Storm Begin Date
2. EVTYPE - Event Type
3. FATALITIES - Number directly killed
4. INJURIES - Number directly injured
5. PROPDMG - Proprty Damage
6. PROPDMGEXP - Property Damage multiplier (k for 1000,m for 1000000,b for 1000000000)
7. CROPDMG - Crop Damage
8. CROPDMGEXP - Crop Damage Multiplier (k for 1000,m for 1000000,b for 1000000000)

storm\_data <- storm\_data[,names(storm\_data) %in% c("BGN\_DATE","EVTYPE","FATALITIES","INJURIES","PROPDMG","PROPDMGEXP","CROPDMG","CROPDMGEXP")]  
  
names(storm\_data) <- c("BeginDate","EventType","Fatalities","Injuries","PropertyDamage","PropertyDamageMultiplier","CropDamage","CropDamageMultiplier")

Filter the data set to extract rows relevant for analysis. This includes removing entries before 1995 due to lack of good records. Observations where there are no injurties or damages are also excluded

# Use which function to eliminate NAs  
storm\_data <- storm\_data[which(storm\_data$Fatalities!=0|storm\_data$Injuries!=0|storm\_data$PropertyDamage!=0|storm\_data$CropDamage!=0),]  
  
# Convert all event types to lower case for consistency  
storm\_data <- data.frame(lapply(storm\_data, function(v){  
 if(is.character(v))   
 tolower(v)  
 }  
))  
  
#Filter observations older than 1995. Use mdy\_hms to parse dates since they have hours:minutes:seconds elements and transform dates in character format to POSIXct format. year function expects a POSIXlt or POSIXct format. Example date: 4/18/1950 0:00:00  
library(lubridate)  
storm\_data <- storm\_data[year(mdy\_hms(storm\_data$BeginDate)) >= 1996,]

**Data Cleanup and Mapping**

The list of 48 permitted storm data events can be found in section 2.1 of the National Weather Service "Storm Data Documentation" document. If the event type contains concatenated list of 2 event types only one is used for the mapping depending on what seems most appropriate. The reasoning and logic behind determining appropriate mappings was derived from the National Weather Service instruction document.

The gsub function is used to perform replacement of all observations that match the wild card pattern

storm\_data$Event<-gsub("^tornado.\*","Tornado",storm\_data$EventType)  
storm\_data$Event<-gsub("^light.\*","Lightening",storm\_data$Event)  
storm\_data$Event<-gsub("^waterspout.\*","Waterspout",storm\_data$Event)  
storm\_data$Event<-gsub("^marine hail.\*","Marine Hail",storm\_data$Event)  
storm\_data$Event<-gsub("^marine high.\*","Marine High Wind",storm\_data$Event)  
storm\_data$Event<-gsub("^(marine strong|gusty).\*","Marine Strong Wind",storm\_data$Event)  
storm\_data$Event<-gsub("^marine thunder.\*","Marine Thunderstorm Wind",storm\_data$Event)  
storm\_data$Event<-gsub("^marine tstm.\*","Marine Thunderstorm Wind",storm\_data$Event)  
storm\_data$Event<-gsub("^drought.\*","Drought",storm\_data$Event)  
storm\_data$Event<-gsub("^(heat wave|\*excessive heat|record heat|extreme heat|record/excessive heat).\*","Excessive Heat",storm\_data$Event)  
storm\_data$Event<-gsub("^(heat|unseasonably warm).\*","Heat",storm\_data$Event)  
storm\_data$Event<-gsub("^winter storm.\*","Winter Storm",storm\_data$Event)  
storm\_data$Event<-gsub("^(hurricane|typhoon).\*","Hurricane",storm\_data$Event)  
storm\_data$Event<-gsub("^(dense fog|fog).\*","Dense Fog",storm\_data$Event)  
storm\_data$Event<-gsub("^dense smoke.\*","Dense Smoke",storm\_data$Event)  
storm\_data$Event<-gsub("^(freezing fog|glaze).\*","Freezing Fog",storm\_data$Event)  
storm\_data$Event<-gsub("^(rip|gradient).\*","Rip Current",storm\_data$Event)  
storm\_data$Event<-gsub("^(coastal flood|tidal flood).\*","Coastal Flood",storm\_data$Event)  
storm\_data$Event<-gsub("^avalanche.\*","Avalanche",storm\_data$Event)  
storm\_data$Event<-gsub("^(dust storm|blowing dust).\*","Dust Storm",storm\_data$Event)  
storm\_data$Event<-gsub("^(dust devil).\*","Dust Devil",storm\_data$Event)  
storm\_data$Event<-gsub(".\*surge.\*","Storm Surge",storm\_data$Event)  
storm\_data$Event<-gsub(".\*low tide.\*","Astonomical Low Tide",storm\_data$Event)  
storm\_data$Event<-gsub("^seiche.\*","Seiche",storm\_data$Event)  
storm\_data$Event<-gsub("^(lakeshore flood|lake flood).\*","Lakeshore Flood",storm\_data$Event)  
storm\_data$Event<-gsub(".\*lake.\*snow.\*","Lake-Effect Snow",storm\_data$Event)  
storm\_data$Event<-gsub(".\*fire.\*","Wildfire",storm\_data$Event)  
storm\_data$Event<-gsub("^(tropical storm|coastal storm).\*","Tropical Storm",storm\_data$Event)  
storm\_data$Event<-gsub("^tropical depression.\*","Tropical Depression",storm\_data$Event)  
storm\_data$Event<-gsub("^tsunami.\*","Tsunami",storm\_data$Event)  
storm\_data$Event<-gsub("^volcanic.\*","Volcanic Ash",storm\_data$Event)  
storm\_data$Event<-gsub(" ?flash flood.\*","Flash Flood",storm\_data$Event)  
storm\_data$Event<-gsub("^(ice jam|dam break).\*","Flash Flood",storm\_data$Event)  
storm\_data$Event<-gsub(".\*slide.\*","Debris Flow",storm\_data$Event)  
storm\_data$Event<-gsub("^(hail|small hail).\*","Hail",storm\_data$Event)  
storm\_data$Event<-gsub("^(blizzard|ground blizzard|blowing snow).\*","Blizzard",storm\_data$Event)  
storm\_data$Event<-gsub("^funnel cloud.\*","Funnel Cloud",storm\_data$Event)  
storm\_data$Event<-gsub("^cold.\*","Cold",storm\_data$Event)  
storm\_data$Event<-gsub("^(extreme cold|extreme wind|record cold|extended cold).\*","Extreme Cold",storm\_data$Event)  
storm\_data$Event<-gsub("^(wind|high wind).\*","High Wind",storm\_data$Event)  
storm\_data$Event<-gsub("^(thunder|severe thunder|gustnado|thunerstorm|thuderstorm|thundeer|tunderstorm).\*","Thunderstorm Wind",storm\_data$Event)  
storm\_data$Event<-gsub(" ?tstm.\*","Thunderstorm Wind",storm\_data$Event)  
storm\_data$Event<-gsub(".\*(burst|whirlwind).\*","Thunderstorm Wind",storm\_data$Event)  
storm\_data$Event<-gsub("^(sleet|black ice).\*","Sleet",storm\_data$Event)  
storm\_data$Event<-gsub(".\*ice storm.\*","Ice Storm",storm\_data$Event)  
storm\_data$Event<-gsub("^(ice.\*strong winds).\*","Ice Storm",storm\_data$Event)  
storm\_data$Event<-gsub("^(hyper|hypo).\*","Ice Storm",storm\_data$Event)  
storm\_data$Event<-gsub("^(freeze|hard freeze|damaging freeze|frost).\*","Freeze",storm\_data$Event)  
storm\_data$Event<-gsub("^(hvy|rain|heavy rain|heavy preci|heavy shower|excessive wetness).\*","Heavy Rain",storm\_data$Event)  
storm\_data$Event<-gsub(".\*(rainfall|rainstorm).\*","Heavy Rain",storm\_data$Event)  
storm\_data$Event<-gsub(".\*(high tide|surf|high sea|high wave|rogue wave|rough sea|heavy sea|heavy swell|erosion).\*","High Surf",storm\_data$Event)  
storm\_data$Event<-gsub("^(heavy snow|excessive snow|record snow|snow and heavy snow|snow/heavy snow|snow accu|snow quall|snow/high winds).\*","Heavy Snow",storm\_data$Event)  
storm\_data$Event<-gsub("^(strong wind|storm force wind|non tstm wind|non-tstm wind|wind storm).\*","Strong Wind",storm\_data$Event)  
storm\_data$Event<-gsub("^(falling snow|freezing drizzle|freezing rain|light freezing rain|light snow|low temp|mixed preci|winter|wintry).\*","Winter Weather",storm\_data$Event)  
storm\_data$Event<-gsub("^(snow).\*","Winter Weather",storm\_data$Event)  
storm\_data$Event<-gsub("^(flood|breakup flood|major flood|minor flood).\*","Flood",storm\_data$Event)  
storm\_data$Event<-gsub(".\*(river|rural flood|small stream flood|urban flood).\*","Flood",storm\_data$Event)

**Data Aggregation**

Compute total fatalities, injuries and damage by event type. The data frame is converted into a data table and new columns are created to house aggregate data

library(data.table)  
storm\_data <- data.table(storm\_data)  
  
#Multiply multiplier for property and crop damage.   
storm\_data <- storm\_data[ ,Multiplier:=   
(  
ifelse(PropertyDamageMultiplier=="k",1000,  
ifelse(PropertyDamageMultiplier=="m",1000000,  
ifelse(PropertyDamageMultiplier=="b",1000000000,NA)))  
)]  
  
storm\_data <- storm\_data[,TotalFatalities:=sum(as.numeric(Fatalities)),by=Event]  
storm\_data <- storm\_data[,TotalInjuries:=sum(as.numeric(Injuries)), by=Event]  
storm\_data <- storm\_data[,TotalPropertyDamage:=sum(as.numeric(PropertyDamage)), by=Event]  
storm\_data <- storm\_data[,TotalCropDamage:=sum(as.numeric(CropDamage)), by=Event]

**Results**

Compute events that led to most damage

#list function to concatenate list of columns needed.  
fatalities <- data.table(storm\_data[,list(Event,TotalFatalities)])  
fatalities <- unique(fatalities[order(-TotalFatalities)])  
  
injuries <- data.table(storm\_data[,list(Event,TotalInjuries)])  
injuries <- unique(injuries[order(-TotalInjuries)])  
  
property\_damage <- data.table(storm\_data[!is.na(storm\_data$Multiplier)])  
property\_damage <- property\_damage[,list(Event, TotalPropertyDamage,Multiplier)]   
property\_damage <- unique(property\_damage[order(-TotalPropertyDamage\*Multiplier)])  
  
crop\_damage <- data.table(storm\_data[!is.na(storm\_data$Multiplier)])  
crop\_damage <- crop\_damage[,list(Event, TotalCropDamage,Multiplier)]   
crop\_damage <- unique(crop\_damage[order(-TotalCropDamage\*Multiplier)])

1. Events with highest fatality rate

head(fatalities,5)

## Event TotalFatalities  
## 1: Thunderstorm Wind 212362  
## 2: Hail 207775  
## 3: Flash Flood 54694  
## 4: Tornado 28515  
## 5: Flood 26128

1. Events with the highest injury rate

head(injuries,5)

## Event TotalInjuries  
## 1: Thunderstorm Wind 293350  
## 2: Hail 214838  
## 3: Tornado 148025  
## 4: Lightening 76990  
## 5: Flash Flood 65440

1. Events with the highest property damage

property\_damage <- as.data.frame(property\_damage)  
property\_damage[1:5,1:2]

## Event TotalPropertyDamage  
## 1 Flash Flood 10323434  
## 2 Hail 10123152  
## 3 Tornado 7019249  
## 4 Flood 5281062  
## 5 High Wind 2974897

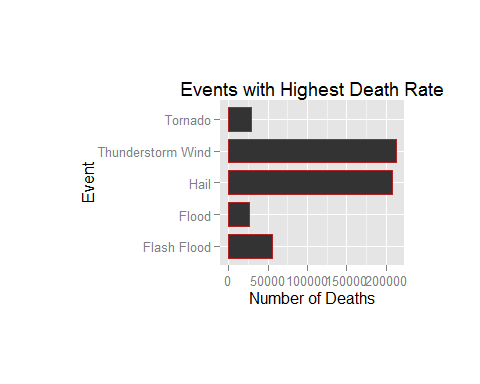
1. Events with the highest crop damage

crop\_damage <- as.data.frame(crop\_damage)  
crop\_damage[1:5,1:2]

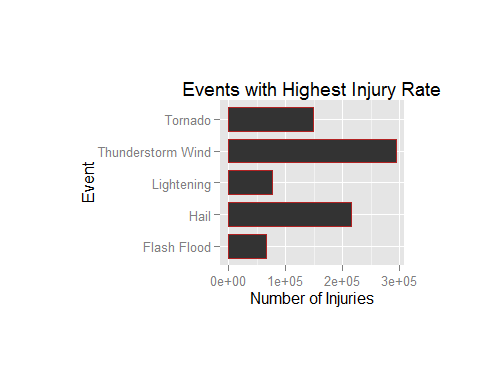
## Event TotalCropDamage  
## 1 Hail 1773812  
## 2 Flash Flood 390715  
## 3 Flood 320142  
## 4 Tornado 233226  
## 5 High Wind 58004

**Data Visualization**

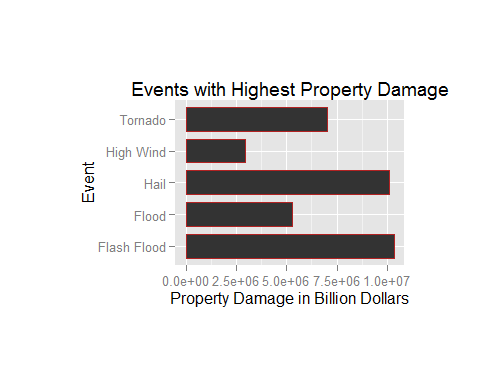
library(ggplot2)  
library(grid)  
  
#Use stat=identity so that geom\_bar treats data as is, instead of computing counts for bins  
f<-head(fatalities,5)  
  
ggplot(data=f,aes(x=Event, y=TotalFatalities))+geom\_bar(stat="identity",color="firebrick",width=0.75)+coord\_flip()+xlab("Event")+ylab("Number of Deaths")+ggtitle("Events with Highest Death Rate")+theme(plot.margin = unit(c(2, 2, 2, 2), "cm"))



i<-head(injuries,5)  
ggplot(data=i,aes(Event, TotalInjuries))+geom\_bar(stat="identity",color="firebrick",width=0.75)+coord\_flip()+xlab("Event")+ylab("Number of Injuries")+ggtitle("Events with Highest Injury Rate")+theme(plot.margin = unit(c(2, 2, 2, 2), "cm"))



pd<-head(property\_damage,5)  
ggplot(data=pd,aes(Event, TotalPropertyDamage))+geom\_bar(stat="identity",color="firebrick",width=0.75)+coord\_flip()+xlab("Event")+ylab("Property Damage in Billion Dollars")+ggtitle("Events with Highest Property Damage")+theme(plot.margin = unit(c(2, 2, 2, 2), "cm"))



cd<-head(crop\_damage,5)  
ggplot(data=cd,aes(Event, TotalCropDamage))+geom\_bar(stat="identity",color="firebrick",width=0.75)+coord\_flip()+xlab("Event")+ylab("Crop Damage in Billion Dollars")+ggtitle("Events with Highest Crop Damage")+theme(plot.margin = unit(c(2, 2, 2, 2), "cm"))

