# Analysis of Weather Events Impact on the United States Population Health and Economy

*JRB*

*July 12, 2016*

## Synopsis

As a local government or municipality it is important to understand the impact weather events could have on our local economy and population. This analysis based on the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database will identify a set of weather events that have the most impact on local economies and populations. The analysis will be performed in aggregate for the whole United States (the current analysis will not provide details by state or county) and cumulatively for the last 60 years. We will first look at events through the decade to see if climatic change would impact the event reported. We will aggregate the data through the reporting period and identify the the top 5 events in terms of impact to population and US economy.

## Data Processing

### Acquiring the data

We will first downlad the NOAA database from Storm Data [47Mb]. The file was downloaded on `2016-07-14`. We will directly read the csv file from the zipped file in to a R data frame. We use the dplyr package to facilitate further data processing.

```
if (!file.exists("./data/archive.zip")) {
    file.url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
    file.destination <- "data"
    file.name <- "storms.csv"
    date.downloaded <- Sys.Date()
    download.file(file.url, "./data/archive.zip", "curl", quiet = TRUE)
}
## read the data into R
dfstorms <- read.csv("./data/archive.zip", stringsAsFactors = TRUE)

## Making the dataframe as table as they are easier to print and manipulate
library(dplyr)
storms <- tbl_df(dfstorms)
str(storms)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    902297 obs. of  37 variables:
##  $ STATE__   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ BGN_DATE  : Factor w/ 16335 levels "1/1/1966 0:00:00",..: 6523 6523 4242 11116 2224 2224 2260 383
##  $ BGN_TIME  : Factor w/ 3608 levels "00:00:00 AM",..: 272 287 2705 1683 2584 3186 242 1683 3186 318
##  $ TIME_ZONE : Factor w/ 22 levels "ADT","AKS","AST",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ COUNTY    : num  97 3 57 89 43 77 9 123 125 57 ...
##  $ COUNTYNAME: Factor w/ 29601 levels "","5NM E OF MACKINAC BRIDGE TO PRESQUE ISLE LT MI",..: 13513
##  $ STATE     : Factor w/ 72 levels "AK","AL","AM",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ EVTYPE    : Factor w/ 985 levels "   HIGH SURF ADVISORY",..: 834 834 834 834 834 834 834 834 834 8
##  $ BGN_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ BGN_AZI   : Factor w/ 35 levels "","  N"," NW",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ BGN_LOCATI: Factor w/ 54429 levels ""," Christiansburg",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ END_DATE  : Factor w/ 6663 levels "","1/1/1993 0:00:00",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ END_TIME  : Factor w/ 3647 levels ""," 0900CST",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_END: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ COUNTYENDN: logi  NA NA NA NA NA NA ...
##  $ END_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ END_AZI   : Factor w/ 24 levels "","E","ENE","ESE",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ END_LOCATI: Factor w/ 34506 levels ""," CANTON"," TULIA",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ LENGTH    : num  14 2 0.1 0 0 1.5 1.5 0 3.3 2.3 ...
##  $ WIDTH     : num  100 150 123 100 150 177 33 33 100 100 ...
##  $ F         : int  3 2 2 2 2 2 2 1 3 3 ...
##  $ MAG       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ FATALITIES: num  0 0 0 0 0 0 0 0 1 0 ...
##  $ INJURIES  : num  15 0 2 2 2 6 1 0 14 0 ...
##  $ PROPDMG   : num  25 2.5 25 2.5 2.5 2.5 2.5 2.5 25 25 ...
##  $ PROPDMGEXP: Factor w/ 19 levels "","-","?","+",..: 17 17 17 17 17 17 17 17 17 17 ...
##  $ CROPDMG   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CROPDMGEXP: Factor w/ 9 levels "","?","0","2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ WFO       : Factor w/ 542 levels ""," CI","%SD",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ STATEOFFIC: Factor w/ 250 levels "","ALABAMA, Central",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ ZONENAMES : Factor w/ 25112 levels "","
##  $ LATITUDE  : num  3040 3042 3340 3458 3412 ...
##  $ LONGITUDE : num  8812 8755 8742 8626 8642 ...
##  $ LATITUDE_E: num  3051 0 0 0 0 ...
##  $ LONGITUDE_: num  8806 0 0 0 0 ...
##  $ REMARKS   : Factor w/ 436781 levels "","\t","\t\t",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ REFNUM    : num  1 2 3 4 5 6 7 8 9 10 ...
```

## Cleaning the data

The next steps of the data processing is to clean some of the data ahead of the analysis. We've noticed some inconsitencies in the case used for recording events and some extra white spaces. We are converting to all uper case and trimming any white spaces. Another inconsistency in the data was the use of the "TSTM WIND" event type early in the data set, that was later recorded as "THUNDERSTORM WIND". We've replaced all TSTM WIND entries by THUNDERSTOM WIND to ensure consistency in the analysis across the dataset. Finally, we converted the BGN_DATE variable into a POSIXct date object to support analysis based on date.

```r
## Cleaning up the EVTYPE, through exploration of the data noticed some
## inconsitencies so addressing those here

## Inconsistencies of reporting Thunderstorm wind as TSTM WInd and
## Thunderstorm Wind this line addresses the issue and makes it conistent as
## 'THUNDERSTORM WINDS'
library(lubridate)
storms$EVTYPE[grep("THUNDERSTORM", storms$EVTYPE)] <- "THUNDERSTORM WIND"
storms <- storms %>% mutate(EVTYPE = toupper(trimws(storms$EVTYPE, which = "both")),
    BGN_DATE = mdy_hms(as.character(storms$BGN_DATE))) %>% mutate(EVTYPE = gsub("^TSTM WIND",
    "THUNDERSTORM WIND", EVTYPE)) %>% mutate(EVTYPE = gsub("^THUNDERSTORM WIND",
    "THUNDERSTORM WIND", EVTYPE)) %>% mutate(EVTYPE = gsub("^THUNDERSTORM WINDS",
    "THUNDERSTORM WIND", EVTYPE)) %>% mutate(EVTYPE = gsub("^THUNDERSTORM WIND [G][0-9][0-9]",
    "THUNDERSTORM WIND", EVTYPE)) %>% mutate(EVTYPE = as.factor(EVTYPE))


# Multiple plot function ggplot objects can be passed in ..., or to plotlist
# (as a list of ggplot objects) - cols: Number of columns in layout -
# layout: A matrix specifying the layout. If present, 'cols' is ignored.  If
# the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then
# plot 1 will go in the upper left, 2 will go in the upper right, and 3 will
# go all the way across the bottom.
multiplot <- function(..., plotlist = NULL, file, cols = 1, layout = NULL) {
    library(grid)

    # Make a list from the ... arguments and plotlist
    plots <- c(list(...), plotlist)

    numPlots = length(plots)

    # If layout is NULL, then use 'cols' to determine layout
    if (is.null(layout)) {
        # Make the panel ncol: Number of columns of plots nrow: Number of rows
        # needed, calculated from # of cols
        layout <- matrix(seq(1, cols * ceiling(numPlots/cols)), ncol = cols,
            nrow = ceiling(numPlots/cols))
    }

    if (numPlots == 1) {
        print(plots[[1]])

    } else {
        # Set up the page
        grid.newpage()
        pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

        # Make each plot, in the correct location
        for (i in 1:numPlots) {
            # Get the i,j matrix positions of the regions that contain this subplot
            matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

            print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row, layout.pos.col = matchidx$col
        }
    }
}
```

## Data Analysis

### Selecting data of interest

We will first subset the large storms data set to a smaller data set containing: *Event start date (BGN_DATE)* Event type (EVTYPE) *Fatalities reported (FATALITIES)* Injuries reported (INJURIES) *Property damage in USD (PROPDMG)* Property damage magnitude (PROPDMGEXP) *Crop damage in USD (CROPDMG)* Crop damage magnitude (PROPDMGEXP)

```r
library(dplyr, quietly = TRUE)
library(lubridate, quietly = TRUE)
storms.impact <- storms %>% select(BGN_DATE, EVTYPE, STATE, FATALITIES, INJURIES,
    PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP)
```

### Data Processing of weather events impact on population health

We will first summarize the data (stroms.health.summary) by the total of fatalities and injuries per year, defining population heatlh impact as the sum of FATALITIES and INJURIES, creating the variable POPIMPACT. We then introduce a DECADE variable to cut the data by DECADE for further discussion. Observations with out any injuries or casualties are excluded from the analysis dataset (storms.health.summary). We create a derived dataset (storms.health.summary.by.decade) that summarizes POPIMACT (sum of casualties and injuries) by decade and weather event type. We then select the top 3 event by decade in a derived dataset named top.impact.by.decade. Finally we summarize the population impact across all decade and select the event strictly greater that the 98% percentile into a final dataset called storms.health.summary.overall.

```r
## storms.health.summary aggregate the sum of fatalities and injuries by year
## and event type
storms.health.summary <- storms.impact %>% group_by(EVTYPE, YEAR = year(BGN_DATE)) %>%
    summarise(POPIMPACT = sum(FATALITIES + INJURIES))
## The summary data is broken down by decade for results discussion
storms.health.summary <- mutate(storms.health.summary, DECADE = cut(YEAR, breaks = seq(1949,
    2020, by = 10), labels = c("50s", "60s", "70s", "80s", "90s", "00s", "10s")))
## Only events that had casualties are being considered
storms.health.summary <- filter(storms.health.summary, POPIMPACT != 0)
## Aggregate population health data by decade for future plotting
storms.health.summary.by.decade <- storms.health.summary %>% group_by(DECADE,
    EVTYPE) %>% summarise(POPIMPACT2 = sum(POPIMPACT))
## Define the top 3 impact by decades for plotting
top.impact.by.decade <- arrange(top_n(storms.health.summary.by.decade, 3), DECADE,
    desc(POPIMPACT2), EVTYPE)
## Aggregate Population Heatlh Impact since 1950 and select the 98%
## percentile and above
storms.health.summary.overall <- storms.health.summary %>% filter(DECADE %in%
    c("50s", "60s", "70s", "80s", "90s", "00s", "10s")) %>% group_by(EVTYPE) %>%
    summarise(POPIMPACT2 = sum(POPIMPACT)) %>% arrange(desc(POPIMPACT2)) %>%
    mutate(RANK = cume_dist(POPIMPACT2)) %>% filter(RANK > 0.98)
```

### Data processing of the economic impact of weather events across the United States

We will first summarize the data (stroms.costs.summary) by the total of cost per year, defining population costs impact as the sum of CROPDMG and PROPDMG, creating the variable COSTIMPACT. We then introduce a DECADE variable to cut the data by DECADE for further discussion. Observations with out any costs are excluded from the analysis dataset (storms.impact). We create a derived dataset (storms.costs.summary.by.decade) that summarizes COSTIMPACT (sum of property and crop damages) by decade and weather event type. We then select the top 3 events by decade in a derived dataset named top.costs.by.decade. Finally we summarize the population impact across all decade and select the event strictly greater that the 98% percentile into a final dataset called storms.costs.summary.overall.

```r
## adjusting the PROPDMGEXP and CROPDMGEXP to numeric values
storms.impact$PROPDMGEXP <- as.character(storms.impact$PROPDMGEXP)
storms.impact$PROPDMGEXP[storms.impact$PROPDMGEXP == "K"] <- 3
storms.impact$PROPDMGEXP[storms.impact$PROPDMGEXP == "M"] <- 6
storms.impact$PROPDMGEXP[storms.impact$PROPDMGEXP == "B"] <- 9
storms.impact$PROPDMGEXP[storms.impact$PROPDMGEXP == ""] <- 0
storms.impact$PROPDMGEXP[storms.impact$PROPDMGEXP == "+"] <- 1
storms.impact$PROPDMGEXP[storms.impact$PROPDMGEXP == "-"] <- 0
storms.impact$PROPDMGEXP[storms.impact$PROPDMGEXP == "H"] <- 2
storms.impact$PROPDMGEXP[storms.impact$PROPDMGEXP == "?"] <- 0
storms.impact$PROPDMGEXP <- as.integer(storms.impact$PROPDMGEXP)
```

```
## Warning: NAs introduced by coercion
```

```
storms.impact$CROPDMGEXP <- as.character(storms.impact$CROPDMGEXP)
storms.impact$CROPDMGEXP[storms.impact$CROPDMGEXP == "K"] <- 3
storms.impact$CROPDMGEXP[storms.impact$CROPDMGEXP == "M"] <- 6
storms.impact$CROPDMGEXP[storms.impact$CROPDMGEXP == "B"] <- 9
storms.impact$CROPDMGEXP[storms.impact$CROPDMGEXP == ""] <- 0
storms.impact$CROPDMGEXP[storms.impact$CROPDMGEXP == "+"] <- 1
storms.impact$CROPDMGEXP[storms.impact$CROPDMGEXP == "-"] <- 0
storms.impact$CROPDMGEXP[storms.impact$CROPDMGEXP == "H"] <- 2
storms.impact$CROPDMGEXP[storms.impact$CROPDMGEXP == "?"] <- 0
storms.impact$CROPDMGEXP <- as.integer(storms.impact$CROPDMGEXP)
```

```
## Warning: NAs introduced by coercion
```

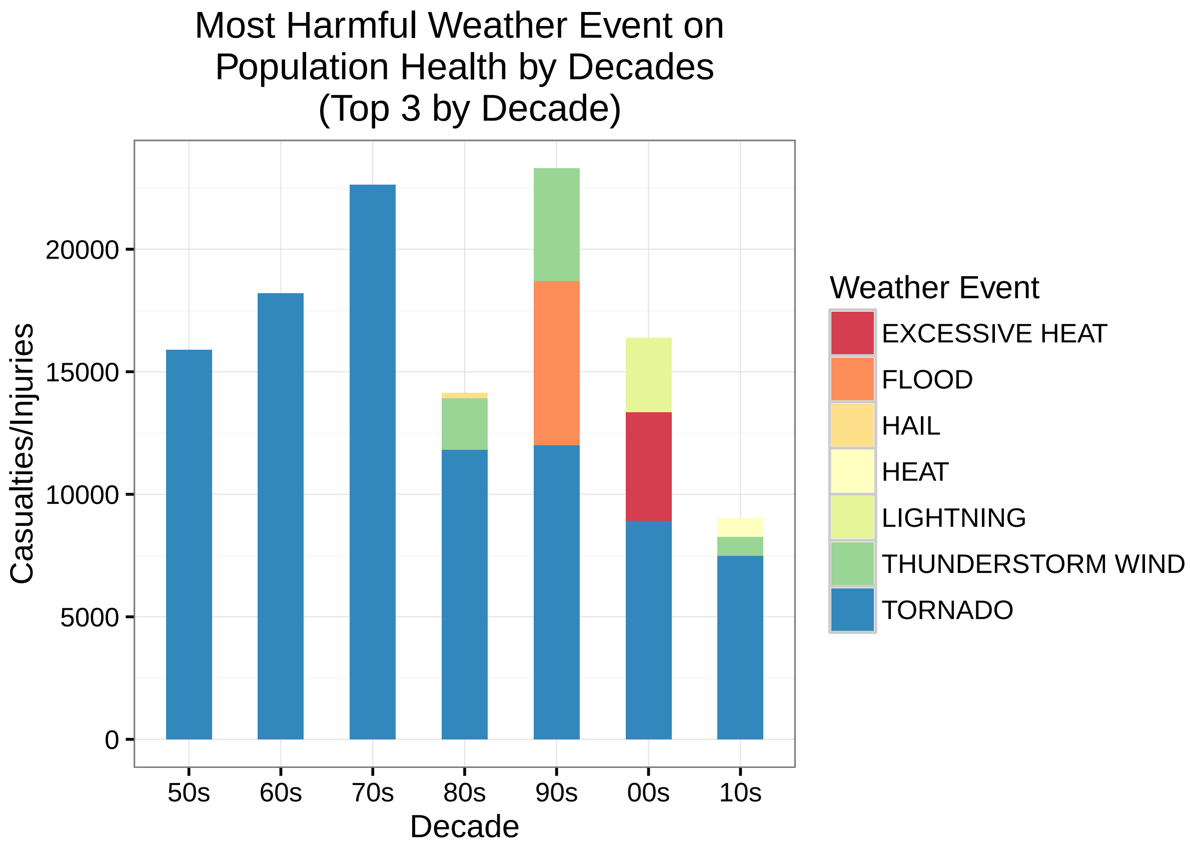
```
storms.impact.summary <- storms.impact %>% group_by(EVTYPE, YEAR = year(BGN_DATE)) %>%
    summarise(COSTIMPACT = sum(CROPDMG * 10^CROPDMGEXP + PROPDMG * 10^PROPDMGEXP))
## Breaks data by decade storms.costs.summary <- mutate( storms.impact,
## DECADE=cut(year(BGN_DATE), breaks=seq(1949,2020,by=10),
## labels=c('50s','60s','70s','80s','90s','00s','10s') ), COSTIMPACT =
## CROPDMG*10^CROPDMGEXP + PROPDMG*10^PROPDMGEXP ) Filte# Looking at
## Population costs Impact r events that had 0 casualties
storms.costs.summary <- filter(storms.impact.summary, COSTIMPACT != 0)
## looked at most impactful events by decade on heatlh
## storms.costs.summary.by.decade <- storms.costs.summary %>% group_by(
## DECADE, EVTYPE) %>% summarise( COSTIMPACT2 = sum(COSTIMPACT) ) Overall
## since 80s
storms.costs.summary.overall <- storms.costs.summary %>% group_by(EVTYPE) %>%
    summarise(COSTIMPACT2 = sum(COSTIMPACT)) %>% arrange(desc(COSTIMPACT2))
storms.costs.summary.overall <- arrange(storms.costs.summary.overall, desc(COSTIMPACT2))
## find top 3 impact by decades top.costs.by.decade <- arrange( top_n(
## storms.costs.summary.by.decade,3 ), DECADE, desc(COSTIMPACT2), EVTYPE)
```

# Results

## Part 1

The next plot represents themost harmful events for the US population by decades since the 1950s. What the graph shows is that Tornadoes were the only harmful events reported in the 50s, 60s and 70s. Differen types of events like THUNDERSTORM WINDS started to be reported in the 80s. However, as the graph shows Tornadoes are still the most prevalent weather event in term of population arms in the later decades. The analysis could have only selected the events after 1980 to reduce the over reporting of tornadoes in the previous decades, but since Tornadoes are such a prevalent harm to populations (actually the most prevalent in this dataset). The data is reported for all decades both for population arm and cost impact.

```
library(ggplot2)
myp <- ggplot(top.impact.by.decade, aes(DECADE, POPIMPACT2, fill = EVTYPE))
myp <- myp + geom_bar(stat = "identity", width = 0.5)
myp <- myp
myp <- myp + labs(title = "Most Harmful Weather Event on \n Population Health by Decades \n (Top 3 by D
    y = "Casualties/Injuries", x = "Decade")
myp <- myp + scale_fill_brewer(palette = "Spectral", name = "Weather Event")
myp <- myp + theme_bw()
print(myp)
```



**Part 2**

The next step of the analysis looks at the impact of weather events both in terms of population harm and economic costs. To that effect we have plotted (using a Cleveland dot plot) the top 10 events by population harms and costs. Tornado and Thunderstom wind have the highest impact on both population heatlh and economic costs. Flash Flood, Hail and Flood also represent a significant costs impact. Excessevive heat, flood and lightning are also very hamrful to population health.

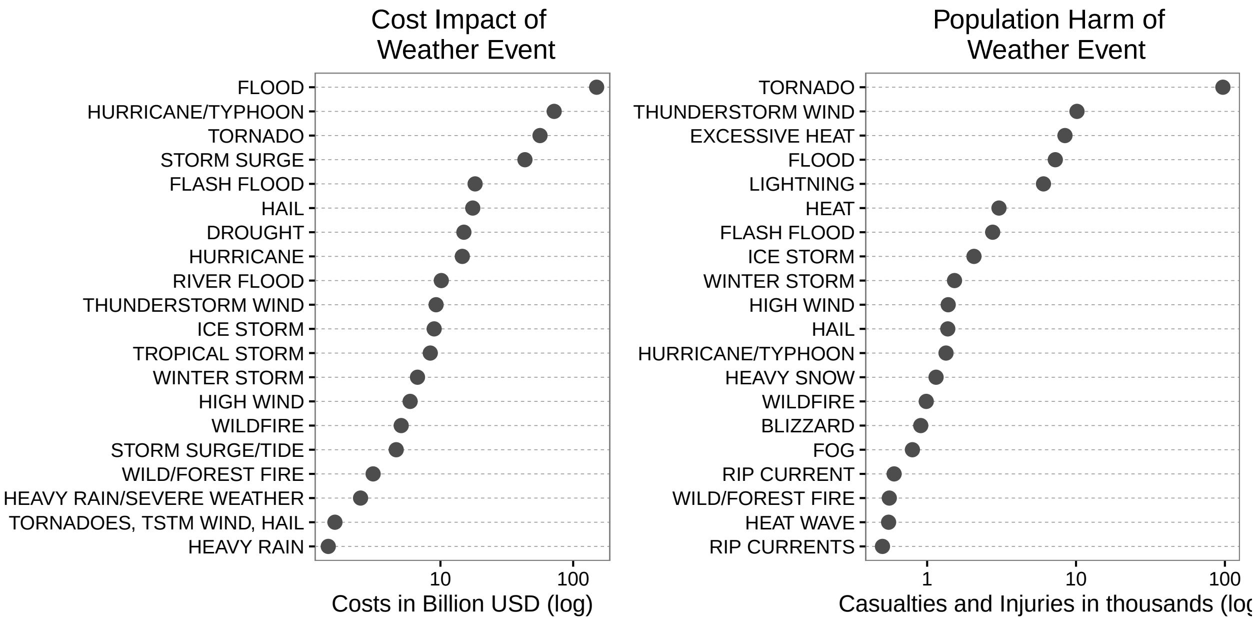
```
## Cleveland Dot Plot
cutoff <- 20
cleveland.costs <- top_n(aggregate(data = storms.costs.summary.overall, COSTIMPACT2 ~
    EVTYPE, sum), cutoff)
```

```
myp1 <- ggplot(cleveland.costs, aes(x = COSTIMPACT2/1e+09, y = reorder(EVTYPE,
    COSTIMPACT2)))

myp1 <- myp1 + geom_point(size = 3, col = "grey30")
myp1 <- myp1 + theme_bw()
myp1 <- myp1 + theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank(),
    panel.grid.major.y = element_line(colour = "grey60", linetype = "dashed"))
myp1 <- myp1 + ggtitle("Cost Impact of \n Weather Event")
myp1 <- myp1 + xlab("Costs in Billion USD (log)")
myp1 <- myp1 + theme(axis.title.y = element_blank())
myp1 <- myp1 + scale_x_log10()


##
cleveland.costs <- top_n(aggregate(data = storms.health.summary, POPIMPACT ~
    EVTYPE, sum), cutoff)
myp2 <- ggplot(cleveland.costs, aes(x = POPIMPACT/1000, y = reorder(EVTYPE,
    POPIMPACT)))
myp2 <- myp2 + geom_point(size = 3, col = "grey30")
myp2 <- myp2 + theme_bw()
myp2 <- myp2 + theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank(),
    panel.grid.major.y = element_line(colour = "grey60", linetype = "dashed"))
myp2 <- myp2 + ggtitle("Population Harm of \n Weather Event")
myp2 <- myp2 + xlab("Casualties and Injuries in thousands (log)")
myp2 <- myp2 + theme(axis.title.y = element_blank())
myp2 <- myp2 + scale_x_log10()

multiplot(myp1, myp2, cols = 2)
```



## Conclusion

Local government and municipalities should focus in priority on preparedness for the most severe events identified in this analysis namely: Tornadoes and thuderstorm wind. Also, awarenes should be developed around excessive heat events and flooding that can be harmful to the population across the US. Further analysis should focus on local data and recent events to refine local preparedness plans for local government and municipalities,