

NOAA Exploratory Report

Akhil Ahuja

10/10/2020

HEALTH AND ECONOMIC CONSEQUENCES OF SEVERE WEATHER

SYNOPSIS

Storms and other severe weather conditions prove very harmful for public health and being. Hence it is the need of hour to analyze all the data that humans have collected and use it to predict future events and save as much destruction of life and property as possible. This report focuses on using the data from the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database to answer the following questions:-

1. Across the United States, which types of events (as indicated in the EVTYPE 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

Loading necessary packages

If you don't have the following packages installed, you might need to install the packages to reproduce the same results. Use the command "install.packages(dplyr)" and "install.packages(RColorBrewer)" to install.

```
library(dplyr)
library(RColorBrewer)
library(maps)
```

Downloading and Reading Data

```
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2",
              "repdata_data_StromData.csv.bz2")
SD <- read.csv("repdata_data_StromData.csv.bz2", header = TRUE)
```

Getting to know the Data

```
names(SD)
```

```
## [1] "STATE_"      "BGN_DATE"    "BGN_TIME"    "TIME_ZONE"   "COUNTY"
## [6] "COUNTYNAME" "STATE"       "EVTYPE"      "BGN_RANGE"   "BGN_AZI"
## [11] "BGN_LOCATI"  "END_DATE"    "END_TIME"    "COUNTY_END" "COUNTYENDN"
## [16] "END_RANGE"   "END_AZI"     "END_LOCATI"  "LENGTH"      "WIDTH"
## [21] "F"           "MAG"         "FATALITIES"  "INJURIES"    "PROPDMG"
## [26] "PROPDMGEXP"  "CROPDMG"     "CROPDMGEXP"  "WFO"          "STATEOFFIC"
## [31] "ZONENAMES"   "LATITUDE"    "LONGITUDE"   "LATITUDE_E"  "LONGITUDE_"
## [36] "REMARKS"     "REFNUM"
```

```
str(SD)
```

```
## 'data.frame':    902297 obs. of  37 variables:
## $ STATE_ : num  1 1 1 1 1 1 1 1 1 1 ...
## $ BGN_DATE : chr   "4/18/1950 0:00:00" "4/18/1950 0:00:00" "2/20/1951 0:00:00" "6/8/1951 0:00:00" .
## $ BGN_TIME : chr   "0130" "0145" "1600" "0900" ...
## $ TIME_ZONE : chr   "CST" "CST" "CST" "CST" ...
## $ COUNTY : num  97 3 57 89 43 77 9 123 125 57 ...
## $ COUNTYNAME: chr   "MOBILE" "BALDWIN" "FAYETTE" "MADISON" ...
## $ STATE : chr   "AL" "AL" "AL" "AL" ...
## $ EVTYPE : chr   "TORNADO" "TORNADO" "TORNADO" "TORNADO" ...
## $ BGN_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
## $ BGN_AZI : chr   "" "" "" "" ...
## $ BGN_LOCATI: chr   "" "" "" "" ...
## $ END_DATE : chr   "" "" "" "" ...
## $ END_TIME : chr   "" "" "" "" ...
## $ 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 : chr   "" "" "" "" ...
## $ END_LOCATI: chr   "" "" "" "" ...
## $ 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: chr   "K" "K" "K" "K" ...
## $ CROPDMG : num  0 0 0 0 0 0 0 0 0 0 ...
## $ CROPDMGEXP: chr   "" "" "" "" ...
## $ WFO : chr   "" "" "" "" ...
## $ STATEOFFIC: chr   "" "" "" "" ...
## $ ZONENAMES : chr   "" "" "" "" ...
## $ 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 : chr   "" "" "" "" ...
## $ REFNUM : num  1 2 3 4 5 6 7 8 9 10 ...
```

Cleaning the Data

We need only the following columns “EVTYPE”, “FATALITIES”, “INJURIES”, “PROPDMG”, “PROPDMGEXP”, “CROPDMG”, “CROPDMGEXP”. So, dropping the rest of the columns.

```
SD <- SD[,c("EVTYPE",
            "FATALITIES",
            "INJURIES",
            "PROPDMG",
            "PROPDMGEXP",
            "CROPDMG",
            "CROPDMGEXP",
            "LATITUDE",
            "LONGITUDE")]
```

Checking for Zereos

```
str(SD[SD$FATALITIES == 0 &
      SD$INJURIES == 0 &
      SD$PROPDMG == 0 &
      SD$CROPDMG == 0, ])
```

```
## 'data.frame': 647664 obs. of 9 variables:
## $ EVTYPE : chr "TSTM WIND" "HAIL" "HAIL" "TSTM WIND" ...
## $ FATALITIES: num 0 0 0 0 0 0 0 0 0 0 ...
## $ INJURIES : num 0 0 0 0 0 0 0 0 0 0 ...
## $ PROPDMG : num 0 0 0 0 0 0 0 0 0 0 ...
## $ PROPDMGEXP: chr "" "" "" "" ...
## $ CROPDMG : num 0 0 0 0 0 0 0 0 0 0 ...
## $ CROPDMGEXP: chr "" "" "" "" ...
## $ LATITUDE : num 3335 3435 3442 3323 3342 ...
## $ LONGITUDE : num 8647 8700 8654 8700 8735 ...
```

There are 647664 rows in the data-frame containing values as 0. We are not interested in the 0 values because we want to find the events that have the maximum impact on health and economy, not the 0 impact. Hence we can filter our required data and remove all the rows in which both health and economy have 0 damage.

```
SD <- SD[SD$FATALITIES > 0 |
        SD$INJURIES > 0 |
        SD$PROPDMG > 0 |
        SD$CROPDMG > 0 , ]
dim(SD)
```

```
## [1] 254633 9
```

We will find the different types of values present in the exponent column of property damage (PROPDMGEXP).

```
unique(SD$PROPDMGEXP)
```

```
## [1] "K" "M" "" "B" "m" "+" "0" "5" "6" "4" "h" "2" "7" "3" "H" "-"
```

Now replacing the exponents with their numerical values.

```
options(scipen = 999) # For disabling scientific notation
SD$PROPDMGEXP[SD$PROPDMGEXP == "" |
               SD$PROPDMGEXP == "0" |
               SD$PROPDMGEXP == "+" |
               SD$PROPDMGEXP == "?" |
               SD$PROPDMGEXP == "-"] <- 1
SD$PROPDMGEXP <- gsub("1", 10, SD$PROPDMGEXP, fixed = TRUE)
SD$PROPDMGEXP <- gsub("[Hh]|2", 100, SD$PROPDMGEXP)
SD$PROPDMGEXP <- gsub("[Kk]|3", 1000, SD$PROPDMGEXP)
SD$PROPDMGEXP <- gsub("4", 10000, SD$PROPDMGEXP, fixed = TRUE)
SD$PROPDMGEXP <- gsub("5", 100000, SD$PROPDMGEXP, fixed = TRUE)
SD$PROPDMGEXP <- gsub("[Mm]|6", 1000000, SD$PROPDMGEXP)
SD$PROPDMGEXP <- gsub("7", 10000000, SD$PROPDMGEXP, fixed = TRUE)
SD$PROPDMGEXP <- gsub("8", 100000000, SD$PROPDMGEXP, fixed = TRUE)
SD$PROPDMGEXP <- gsub("B", 1000000000, SD$PROPDMGEXP, fixed = TRUE)
SD$PROPDMGEXP <- as.numeric(SD$PROPDMGEXP)
```

Now we can multiply the property damage values with their exponents to get the exact values.

```
SD$PROPDGM <- SD$PROPDGM * SD$PROPDMGEXP
```

Similarly, finding the unique values present in Crop Damage Exponents column(CROPDMGEXP).

```
unique(SD$CROPDMGEXP)
```

```
## [1] "" "M" "K" "m" "B" "?" "0" "k"
```

Replacing the Crop Damage Exponents with their numerical values.

```
SD$CROPDMGEXP[SD$CROPDMGEXP == "" |
               SD$CROPDMGEXP == "0" |
               SD$CROPDMGEXP == "?"] <- 1
SD$CROPDMGEXP <- gsub("2", 100, SD$CROPDMGEXP)
SD$CROPDMGEXP <- gsub("[Kk]|3", 1000, SD$CROPDMGEXP)
SD$CROPDMGEXP <- gsub("[Mm]|6", 1000000, SD$CROPDMGEXP)
SD$CROPDMGEXP <- gsub("B", 1000000000, SD$CROPDMGEXP, fixed = TRUE)
SD$CROPDMGEXP <- as.numeric(SD$CROPDMGEXP)
```

Now we can multiply the crop damage values with their exponents to get the exact values.

```
SD$CROPDGM <- SD$CROPDGM * SD$CROPDMGEXP
str(SD)
```

```
## 'data.frame': 254633 obs. of 9 variables:
## $ EVTYPE : chr "TORNADO" "TORNADO" "TORNADO" "TORNADO" ...
## $ FATALITIES: num 0 0 0 0 0 0 0 0 1 0 ...
## $ INJURIES : num 15 0 2 2 2 6 1 0 14 0 ...
```

```
## $ PROPDGM : num 25000 2500 25000 2500 2500 2500 2500 2500 2500 25000 25000 ...
## $ PROPDMGEXP: num 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ...
## $ CROPDMG : num 0 0 0 0 0 0 0 0 0 0 ...
## $ CROPDMGEXP: num 1 1 1 1 1 1 1 1 1 1 ...
## $ LATITUDE : num 3040 3042 3340 3458 3412 ...
## $ LONGITUDE : num 8812 8755 8742 8626 8642 ...
```

Aggregating the data

Aggregating the output of the variables for different Event Types(EVTYPE) using the aggregate function. Type of aggregation is sum.

```
required_data <- SD %>%
  group_by(EVTYPE) %>%
  summarise(FATALITIES = sum(FATALITIES, na.rm = TRUE),
            INJURIES = sum(INJURIES, na.rm = TRUE),
            PROPDGM = sum(PROPDGM, na.rm = TRUE),
            CROPDMG = sum(CROPDMG, na.rm = TRUE),
            .groups = "keep")

summary(required_data)
```

```
##      EVTYPE      FATALITIES      INJURIES      PROPDGM
## Length:488      Min.   :    0.00      Min.   :    0      Min.   :         0
## Class :character 1st Qu.:    0.00      1st Qu.:    0      1st Qu.:        4000
## Mode  :character Median :    0.00      Median :    0      Median :        55000
##              Mean  :   31.04      Mean  :   288      Mean   :   877510120
##              3rd Qu.:    1.00      3rd Qu.:    2      3rd Qu.:   1316250
##              Max.   : 5633.00      Max.   : 91346      Max.   :144657709870
##      CROPDMG
## Min.   :         0
## 1st Qu.:         0
## Median :         0
## Mean   : 100623345
## 3rd Qu.:         5000
## Max.   :13972566000
```

PLOTTING THE DATA

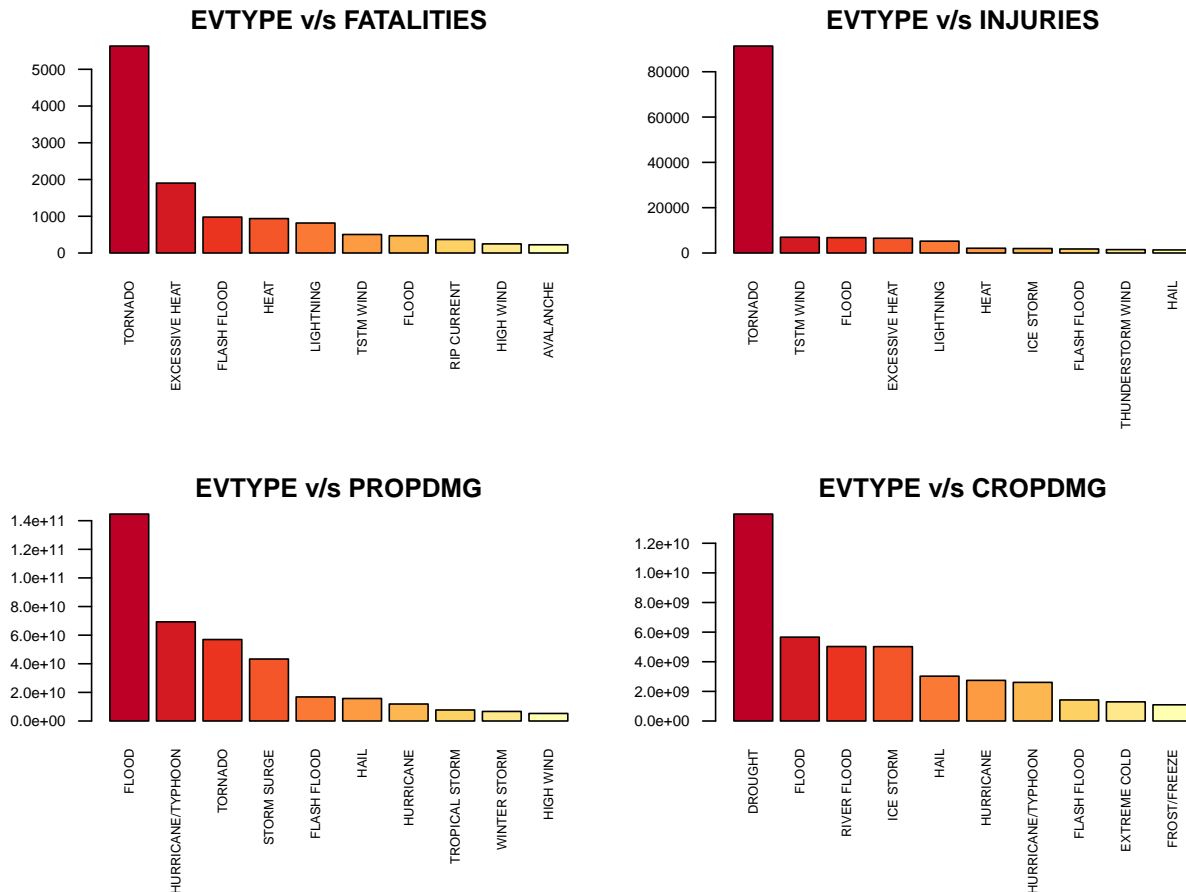
Now, after all the processing, our data ready for plotting in the data_frame called “required_data”. Four bar-graphs will be plotted for FATALITIES, INJURIES, PROPDGM and CROPDMG respectively.

```
options(scipen = 0) # For enabling scientific notation
par(mfrow = c(2,2), mar = c(8,4,2,1))
cols <- brewer.pal(5, "YlOrRd")
pal <- colorRampPalette(cols)
color <- rev(pal(10))
for (x in c(2:5))
{
  names(required_data)
  plotting_data <- as.data.frame(required_data[order(-required_data[, x]), ])
  plotting_data <- plotting_data[c(1:10), ]
```

```

barplot(height = plotting_data[, x],
        main = paste("EVTYPE v/s", names(plotting_data)[x]),
        col = color,
        las = 2,
        names.arg = plotting_data[, 1],
        cex.axis = 0.7,
        cex.names = 0.6)
}

```



As we can see from above, most FATALITIES, INJURIES, PROPDMG and CROPPDMG are caused by TORNADO, TORNADO, FLOOD, DROUGHT respectively.

RESULTS

As we can see from the above graphs, if we talk about health damage, TORNADO event highly exceeds all other events, followed by EXCESSIVE HEAT and TSTM WIND with a difference of 4k deaths and 80k injuries respectively as compared to Tornado.

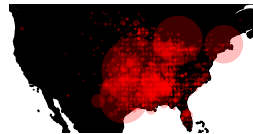
If we talk about economic damage, FLOOD causes maximum property damage followed by HURRICANE/TYPHOON and DROUGHT causes maximum crop damage followed by FLOOD.

Hence, according to this data, maximum measures should be taken to protect life, crop and property from Tornadoes and Floods.

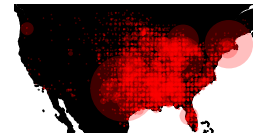
Additional Findings

We have discovered which events do the most damage to health and economy. Let's find out the origin of these events. Size of the circle in the following plots represents the damage done by the event and intensity the color represents number of event occurs.

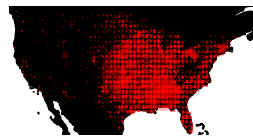
```
par(mar = c(0, 0, 3.5, 0), mfrow = c(2, 2), oma = c(5, 0, 0, 0))
Event <- c("TORNADO", "FLOOD", "TSTM WIND", "FLASH FLOOD")
for (x in Event){
  for (y in c(2, 3, 4, 6)){
    plotting_data <- SD[SD$EVTYPE == x & SD[,y] > 0, ]
    plotting_data$LONGITUDE <- -plotting_data$LONGITUDE/100
    plotting_data$LATITUDE <- plotting_data$LATITUDE/100
    map("world", xlim = c(-125, -65), ylim = c(25, 50), fill = T)
    points(plotting_data$LONGITUDE, plotting_data$LATITUDE,
           cex = 10 * plotting_data[, y] / max(plotting_data[, y]),
           pch = 20,
           col = rgb(1, 0, 0, 0.25))
    title(xlab = names(plotting_data)[y])
  }
  mtext(paste("Origin of", x),
        outer = T,
        cex = 1.5,
        line = 3,
        side = 1)
}
```



FATALITIES



INJURIES



PROPDMG



CROPDMG

Origin of TORNADO



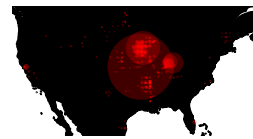
FATALITIES



INJURIES



PROPDMG

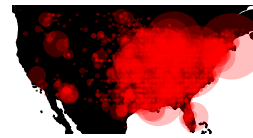


CROPDMG

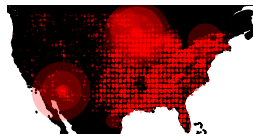
Origin of FLOOD



FATALITIES



INJURIES

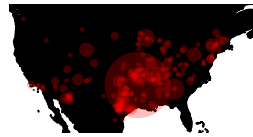


PROPDMG



CROPDMG

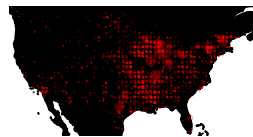
Origin of TSTM WIND



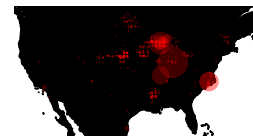
FATALITIES



INJURIES



PROPDMG



CROPDMG

Origin of FLASH FLOOD

Abbreviations Used

- EVTYPE ——— Event Type
- FATALITIES ——— Fatalities/deaths caused
- INJURIES ——— Injuries caused
- PROPDMG ——— Damage caused to property
- PROPDMGEXP — Exponents for damage caused to property
- CROPDMG ——— Damage caused to crop
- CROPDMGEXP — Exponents for damage caused to crop
- LONGITUDE ——— Longitude of the origin of the event
- LATITUDE ——— Latitude of the origin of the event