# Analyzing Storm Data to find events causing maximum damage

*Rahulg13*

*29/12/2019*

This report explores the NOAA dataset to answer two questions - 1. Which are the events most harmful to population health? 2. Which events are associated with greatest economic consequences?

## DATA PROCESSING

```
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2", "stormdata.csv
x <- read.csv("stormdata.csv.bz2")
str(x)
```

```
## '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 3180
##  $ 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 1
##  $ 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 ...
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

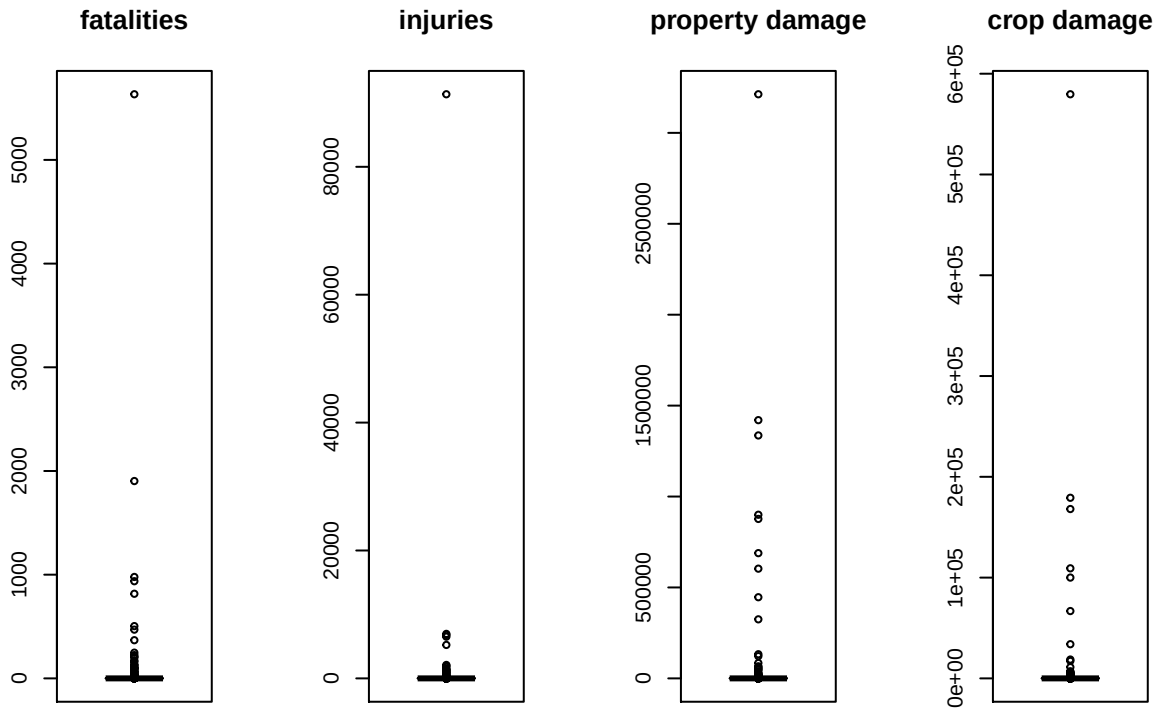
## Data Exploration

1. On exploration of dataset, it is found there 985 event types.
2. We try to find out the events with - 1. Maximum fatalities (FATALITIES) 2. Maximum injuries (INJURIES) 3. Maximum property damage (PROPDMG) 4. Maximum crop damage (CROPDMG) For this, we find totals of fatalities, injuries, property damage and crop damage using the function "tapply". Next, we order them in a decreasing order to help find the events causing maximum damage.

```
eventwise_fat <- with(x, tapply(FATALITIES, EVTYPE, sum))
eventwise_inj <- with(x, tapply(INJURIES, EVTYPE, sum))
eventwise_prop <- with(x, tapply(PROPDMG, EVTYPE, sum))
eventwise_crop <- with(x, tapply(CROPDMG, EVTYPE, sum))
eventwise_fat <- eventwise_fat[order(eventwise_fat, decreasing = TRUE)]
eventwise_inj <- eventwise_inj[order(eventwise_inj, decreasing = TRUE)]
eventwise_prop <- eventwise_prop[order(eventwise_prop, decreasing = TRUE)]
eventwise_crop <- eventwise_crop[order(eventwise_crop, decreasing = TRUE)]
```

To look at the distribution of damage among events, we draw boxplots of the four relevant variables.

```
par(mfrow = c(1, 4))
boxplot(eventwise_fat, main = "fatalities")
boxplot(eventwise_inj, main = "injuries")
boxplot(eventwise_prop, main = "property damage")
boxplot(eventwise_crop, main = "crop damage")
```




Clearly, from above, the damage caused in every type is limited to a few event types where most of the damage is concentrated. To confirm the findings above, we draw the element wise quantiles. (Note 985 is the number of type of events, so 985 quantiles represent one event for each quantile.)

```
tail(quantile(eventwise_fat, prob = seq(0, 1, length.out = 985)), 5)
```

```
##  99.5934959%  99.6951220%  99.7967480%  99.8983740% 100.0000000%
##          816          937          978         1903         5633
```

```
tail(quantile(eventwise_inj, prob = seq(0, 1, length.out = 985)), 5)
```

```
##  99.5934959%  99.6951220%  99.7967480%  99.8983740% 100.0000000%
##         5230         6525         6789         6957        91346
```

```
tail(quantile(eventwise_crop, prob = seq(0, 1, length.out = 985)), 5)
```

```
##  99.5934959%  99.6951220%  99.7967480%  99.8983740% 100.0000000%
##     100018.5     109202.6     168037.9     179200.5     579596.3
```

```
tail(quantile(eventwise_prop, prob = seq(0, 1, length.out = 985)), 5)
```

```
##  99.5934959%  99.6951220%  99.7967480%  99.8983740% 100.0000000%
##     876844.2     899938.5    1335965.6    1420124.6    3212258.2
```

The result confirms that top 5 events contribute more than 90% of damage in each category.

## Data for final analysis and plotting

To detail the events, we subset the data for top 10 categories of event types in each variable category i.e. fatalities, injuries, property damage and crop damage.

```
maxfat_events <- names(eventwise_fat[1:10])
maxinj_events <- names(eventwise_inj[1:10])
x_final <- subset(x, EVTYPE %in% c(maxfat_events, maxinj_events))

maxprop_events <- names(eventwise_prop[1:10])
maxcrop_events <- names(eventwise_crop[1:10])
x_final2 <- subset(x, EVTYPE %in% c(maxprop_events, maxcrop_events))
```

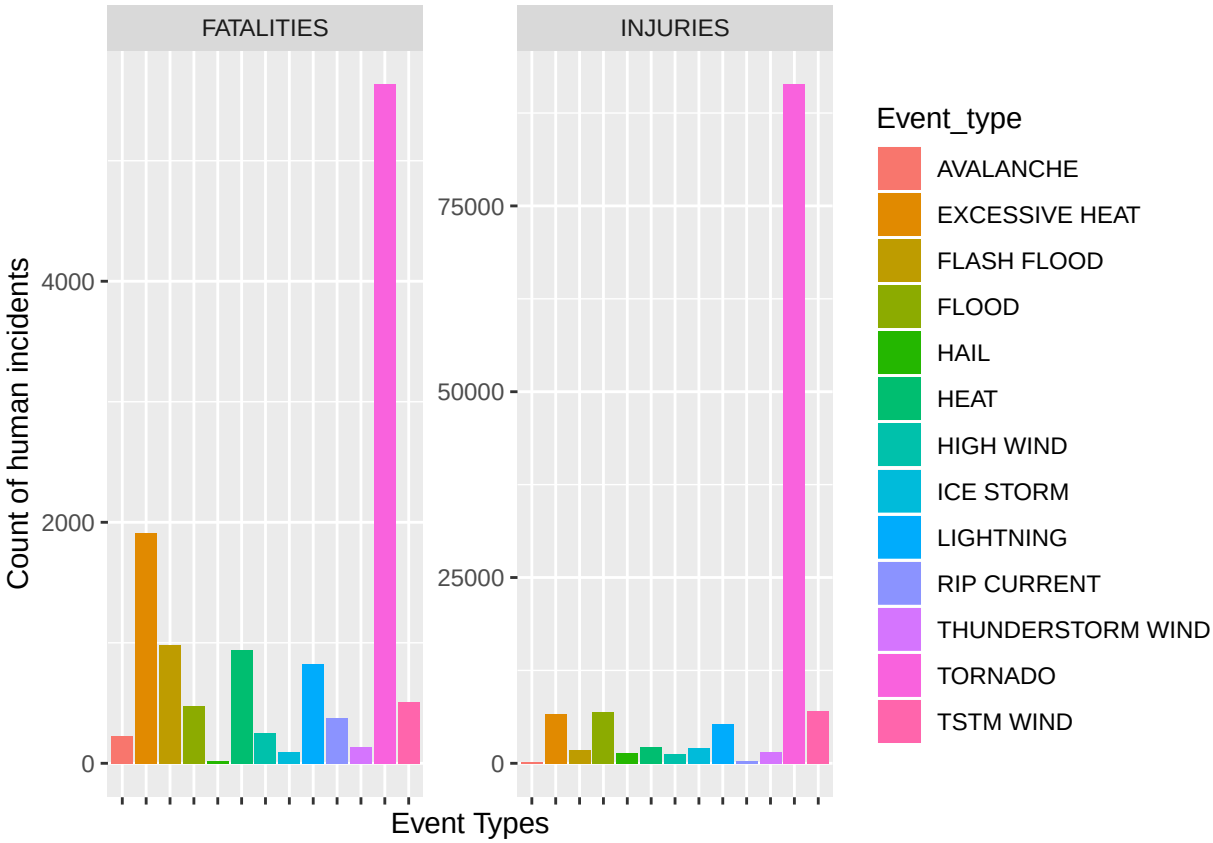
### New dataset for plotting effect on health of human population

```
l <- length(x_final$FATALITIES)
effect_count <- c(x_final$FATALITIES, x_final$INJURIES)
effect_type <- c(rep("FATALITIES", l), rep("INJURIES", l))
event_type <- c(x_final$EVTYPE, x_final$EVTYPE)
x3 <- data.frame(Effect_count = effect_count, Effect_type = effect_type, Event_type =  event_type)

x3$Event_type <- as.factor(x3$Event_type)
levelnames <- as.numeric(levels(x3$Event_type))
levelnames <- levels(x$EVTYPE)[levelnames]
levels(x3$Event_type) <- levelnames
```

### Plot of major events harming health of human population

```
g <- ggplot(x3) + geom_bar(aes(x = Event_type, y = Effect_count, fill = Event_type), stat = "identity")
g <- g + facet_wrap( ~ Effect_type, scales = "free")
g <- g + theme(axis.text.x = element_blank())
g <- g + xlab("Event Types") + ylab("Count of human incidents")
g
```


FATALITIES
INJURIES
Count of human incidents
4000
2000
0
75000
50000
25000
0
Event Types
Event_type
AVALANCHE
EXCESSIVE HEAT
FLASH FLOOD
FLOOD
HAIL
HEAT
HIGH WIND
ICE STORM
LIGHTNING
RIP CURRENT
THUNDERSTORM WIND
TORNADO
TSTM WIND