Storms: Health and Economic Impacts

Francesc Rocher

September 2014

Synopsis

This report explores the social and economic impact of storm events since 1996 to 2011 across USA.

Starting with data from NOAA storm database, there is a deep data clean process to adjust data and events to the description in the codebook. It has been a hard work to complete this part because raw data contains lot of coding errors that are outside the codebook. Moreover, data used has been filtered between 1996 and 2011: previous data is incomplete and even more error prone.

The report concludes with the top 10 events that are most harmful with respect to population, both fatalities and injuries, and with respect to economic impact. Surprisingly, the event that causes more fatalities is *excessive heat*.

Data Processing

This section contains all preliminar data processing. It consist mainly in the filtering, cleaning and management of raw data to get a *tidy data set*.

Libraries used during this process, and later, are:

```
library(ggplot2)
library(plyr)
library(stringr)
```

Downloading Data

First at all, data can be downloaded from the following URL:

```
if(! file.exists("StormData.csv.bz2")) {
    url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
    download.file(url, destfile="StormData.csv.bz2", method="curl")
}
```

Reading data

We first begin by reading the complete *raw* data set:

```
StormData.raw <- read.csv("StormData.csv.bz2", header=TRUE)
```

Filtering by year

According to the NOAA description of Storm Events Database Contents, it is since 1996 that all storms events are stored in the database. We will use storm data recorded since 1996 to address our report. Before 1996 only *few part* of stroms were reported. We are interested in all kind of storms, so taking a long period of time without complete data is not a good point.

The raw data set filtered by year:

```r
StormData.raw.1996_1999 <- grep("19[9][6-9]", StormData.raw$BGN_DATE)
StormData.raw.2000_ <- grep("20[0-9][0-9]", StormData.raw$BGN_DATE)
StormData.raw <- StormData.raw[c(StormData.raw.1996_1999, StormData.raw.2000_),]
```

Removing rows without interest

To address the questions about the most harmful events with respect population and economic consequences, we will use only those events where `FATALITIES`, `INJURIES`, `PROPDMG` or `CROPDMG` contain information (are positive).

The rest of the events are useless for the purpose of this report:

```r
StormData.raw <- subset(StormData.raw, FATALITIES>0 | INJURIES>0 | PROPDMG>0 | CROPDMG>0)
```

Cleaning EVTYPE column

This column needs severe cleaning. In the current data set we have at this stage, column `EVTYPE` contains 0 different values.

We start by doing a *first normalization of strings*, which consists in removing of starting and trailing spaces and converting them to lowercase:

```r
StormData.raw$EVTYPE <- str_trim(tolower(StormData.raw$EVTYPE))
```

Factors in this column are larger than the 48 values specified in the codebook.

```r
EVTYPE.raw <- sort(unique(StormData.raw$EVTYPE))
length(EVTYPE.raw)
```

```
## [1] 183
```

For example, in the first 20 values shown here there are values not considered in the codebook:

```r
head(EVTYPE.raw, n=20)
```

```
##  [1] "agricultural freeze"       "astronomical high tide"
##  [3] "astronomical low tide"     "avalanche"
##  [5] "beach erosion"             "black ice"
##  [7] "blizzard"                  "blowing dust"
##  [9] "blowing snow"              "brush fire"
## [11] "coastal erosion"           "coastal flood"
## [13] "coastal flooding"          "coastal  flooding/erosion"
## [15] "coastal flooding/erosion"  "coastalstorm"
## [17] "coastal storm"             "cold"
## [19] "cold and snow"             "cold temperature"
```

Even more, there is a curious value `other` to specify accidents. For example:

```r
as.character(StormData.raw[StormData.raw$EVTYPE=="other", "REMARKS"][1])
```

```
## [1] "After 3 months of drought, snows finally came...followed by avalanches and then flooding.  A rec
```

From this event *REMARKS* the final *EVENT* could be *drought*, *heavy snow*, *avalanche*, *flood*, *excessive heat* or *flash flood*. Futhermore, the event could be split into several events occurring in different places: heavy snow in the city of Valdez, one avalanche in the Richardson Highway and another one in a local home, and finally a flood in the city streets (Valdez?).

There are 34 rows with this *EVTYPE* value, with vague descriptions or with multipe events. We will ignore these events. This leads a number of *EVTYPE* values of:

```r
StormData.raw <- subset(StormData.raw, EVTYPE!="other")
EVTYPE.raw <- sort(unique(StormData.raw$EVTYPE))
length(EVTYPE.raw)
```

```
## [1] 182
```

To clean `EVTYPE` column we create a new column `EVENT` that maps current *EVTYPE* values into valid *EVENT* values. The map has been elaborated manually, exploring each one of the 179 *EVTYPE* values and assigning the most likely *EVENT* value.

This is the table used to map current *EVTYPE* values to **EVENT** values:

From this table, the map is:

```r
EVTYPEmap <- data.frame(

    EVTYPE=c("agricultural freeze", "astronomical high tide", "avalanche", "beach erosion", "black ice"
             "blowing dust", "blowing snow", "brush fire", "coastal  flooding/erosion", "coastal erosion
             "coastal flood", "coastal flooding", "coastal flooding/erosion", "coastal storm", "coastal
             "cold", "cold and snow", "cold temperature", "cold weather", "cold/wind chill", "dam break
             "damaging freeze", "dense fog", "downburst", "drought", "drowning", "dry microburst", "dus
             "dust storm", "early frost", "erosion/cstl flood", "excessive heat", "excessive snow", "ex
             "extreme cold", "extreme cold/wind chill", "extreme windchill", "falling snow/ice", "flash
             "flash flood/flood", "flood", "flood/flash/flood", "fog", "freeze", "freezing drizzle", "f
             "freezing rain", "freezing spray", "frost", "frost/freeze", "funnel cloud", "glaze", "grad
             "gusty wind", "gusty wind/hail", "gusty wind/hvy rain", "gusty wind/rain", "gusty winds",
             "hard freeze", "hazardous surf", "heat", "heat wave", "heavy rain", "heavy rain/high surf"
             "heavy seas", "heavy snow", "heavy snow shower", "heavy surf", "heavy surf and wind",
             "heavy surf/high surf", "high seas", "high surf", "high surf advisory", "high swells", "hi
             "high wind", "high wind (g40)", "high winds", "hurricane", "hurricane edouard", "hurricane
             "hyperthermia/exposure", "hypothermia/exposure", "ice jam flood (minor", "ice on road", "i
             "ice storm", "icy roads", "lake effect snow", "lake-effect snow", "lakeshore flood", "land
             "landslides", "landslump", "landspout", "late season snow", "light freezing rain", "light
             "light snowfall", "lightning", "marine accident", "marine high wind", "marine strong wind"
             "marine thunderstorm wind", "marine tstm wind", "microburst", "mixed precip", "mixed preci
             "mud slide", "mudslide", "mudslides", "non tstm wind", "non-severe wind damage", "non-tstm
             "rain", "rain/snow", "record heat", "rip current", "rip currents", "river flood", "river f
             "rock slide", "rogue wave", "rough seas", "rough surf", "seiche", "small hail", "snow", "s
             "snow squall", "snow squalls", "storm surge", "storm surge/tide", "strong wind", "strong w
             "thunderstorm", "thunderstorm wind", "thunderstorm wind (g40)", "tidal flooding", "tornado
             "torrential rainfall", "tropical depression", "tropical storm", "tstm wind", "tstm wind  (
             "tstm wind (41)", "tstm wind (g35)", "tstm wind (g40)", "tstm wind (g45)", "tstm wind 40",
             "tstm wind 45", "tstm wind and lightning", "tstm wind g45", "tstm wind/hail", "tsunami", "
             "unseasonable cold", "unseasonably cold", "unseasonably warm", "unseasonal rain",
```

```
              "urban/sml stream fld", "volcanic ash", "warm weather", "waterspout", "wet microburst", "wh
              "wild/forest fire", "wildfire", "wind", "wind and wave", "wind damage", "winds", "winter st
              "winter weather", "winter weather mix", "winter weather/mix", "wintry mix"),

    EVENT=c("frost/freeze", "storm surge/tide", "avalanche", "coastal flood", "ice storm", "blizzard",
              "ice storm", "wildfire", "coastal flood", "coastal flood", "coastal flood", "coastal flood
              "coastal flood", "marine thunderstorm wind", "marine thunderstorm wind", "cold/wind chill"
              "cold/wind chill", "cold/wind chill", "cold/wind chill", "cold/wind chill", "flash flood",
              "frost/freeze", "dense fog", "thunderstorm wind", "drought", "heavy rain", "drought", "dus
              "dust storm", "frost/freeze", "coastal flood", "excessive heat", "heavy snow",
              "extreme cold/wind chill", "extreme cold/wind chill", "extreme cold/wind chill",
              "extreme cold/wind chill", "heavy snow", "flash flood", "flash flood", "flood", "flash flo
              "dense fog", "frost/freeze", "frost/freeze", "freezing fog", "winter strom", "freezing fog
              "frost/freeze", "frost/freeze", "funnel cloud", "frost/freeze", "strong wind", "high wind"
              "thunderstorm wind", "high wind", "thunderstorm wind", "high wind", "hail", "ice storm", "
              "heat", "heat", "heavy rain", "heavy rain", "high surf", "heavy snow", "heavy snow", "high
              "high surf", "high surf", "high surf", "high surf", "high surf", "high surf", "high surf",
              "high wind", "high wind", "hurricane (typhoon)", "hurricane (typhoon)", "hurricane (typhoo
              "excessive heat", "heavy snow", "flood", "ice storm", "ice storm", "ice storm", "ice storm
              "lake-effect snow", "lake-effect snow", "lakeshore flood", "debris flow", "debris flow", "
              "debris flow", "winter weather", "freezing fog", "heavy snow", "heavy snow", "lightning",
              "marine high wind", "marine high wind", "marine strong wind", "marine thunderstorm wind",
              "marine thunderstorm wind", "blizzard", "sleet", "heavy rain", "debris flow", "debris flow
              "debris flow", "high wind", "high wind", "high wind", "heavy rain", "sleet", "excessive he
              "flood", "flood", "flood", "debris flow", "high surf", "high surf", "high surf", "seiche",
              "heavy snow", "ice storm", "heavy snow", "heavy snow", "storm surge/tide", "storm surge/ti
              "strong wind", "strong wind", "thunderstorm wind", "thunderstorm wind", "thunderstorm wind
              "coastal flood", "tornado", "heavy rain", "tropical depression", "tropical storm", "thunde
              "thunderstorm wind", "thunderstorm wind", "thunderstorm wind", "thunderstorm wind",
              "thunderstorm wind", "thunderstorm wind", "thunderstorm wind", "thunderstorm wind",
              "thunderstorm wind", "thunderstorm wind", "tsunami", "typhoon", "extrem cold/wind chill",
              "extrem cold/wind chill", "excessive heat", "heavy rain", "flood", "volcanic ash", "excess
              "waterspout", "waterspout", "funnel cloud", "wildfire", "wildfire", "high wind", "high win
              "strong wind", "high wind", "winter storm", "winter weather", "winter weather", "winter wea
              "winter weather")

 )
```

We can create the new column `EVENT` by mergin *StormData.raw* and *EVTYPEmap* including all rows in *StormData.raw*:

```
StormData.raw <- merge(StormData.raw, EVTYPEmap, all.x=TRUE)
```

Now `EVENT` column contains only valid values:

```
length(unique(StormData.raw$EVENT))
```

```
## [1] 48
```

The only value not assigned is *astronomical low tide*: no *EVTYPE* value has been found that could match this value.

Preparing data for health analysis

Only a subset of the current `StormData.raw` is needed to address health impact:

```
Storm.health <- subset(StormData.raw, FATALITIES>0 | INJURIES>0,
                       c("EVENT", "FATALITIES", "INJURIES"))
```

Preparing data for economic analysis

The same occurs with economic impact:

```
Storm.economy <- subset(StormData.raw, PROPDMG>0 | CROPDMG>0,
                        c("EVENT", "PROPDMG", "PROPDMGEXP", "CROPDMG", "CROPDMGEXP"))
```

In the case of economic analysis we need to convert columns `PROPDMG` and `CROPDMG` to valid values depending on the 10th power in the corresponding columns `PROPDMGEXP` and `CROPDMGEXP`. We first *normalize strings* in these columns as a cleanup procedure:

```
Storm.economy$PROPDMGEXP <- str_trim(tolower(Storm.economy$PROPDMGEXP))
Storm.economy$CROPDMGEXP <- str_trim(tolower(Storm.economy$CROPDMGEXP))
```

The cleanup is finished. Magnitude indicators are correctly set:

```
unique(Storm.economy$PROPDMGEXP)
```

```
## [1] ""  "k" "m" "b"
```

```
unique(Storm.economy$CROPDMGEXP)
```

```
## [1] "m" ""  "k" "b"
```

The conversion of columns `PROPDMG` and `CROPDMG` to integer values is as follows:

```
power10 <- data.frame(factor=c("", "k", "m", "b"),
                      magnitude=c(1, 1000, 1000000, 1000000000))

Storm.economy <- merge(x=Storm.economy, y=power10,
                       by.x=as.factor("PROPDMGEXP"), by.y="factor",
                       all.x=TRUE, sort=FALSE)
colnames(Storm.economy)[6] <- "magnitude.prop"
Storm.economy$PROPDMG <- Storm.economy$PROPDMG * Storm.economy$magnitude.prop

Storm.economy <- merge(x=Storm.economy, y=power10,
                       by.x=as.factor("CROPDMGEXP"), by.y="factor",
                       all.x=TRUE, sort=FALSE)
colnames(Storm.economy)[7] <- "magnitude.crop"
Storm.economy$CROPDMG <- Storm.economy$CROPDMG * Storm.economy$magnitude.crop
```

At this point we have a **tidy data set** to start the analysis of health and economic impacts of storm events.

Data Analysis

Analysis made on the tidy data set is quite obvious: data aggregation on a given aspect will show the total impact of each event.

Health Impact

We consider of crucial importance to separate *fatalities* from *injuries*:

```
fatalities <- aggregate(Storm.health$FATALITIES, by=list(Storm.health$EVENT), FUN=sum)
names(fatalities) <- c("EVENT", "total")
head(fatalities)
```

```
##             EVENT total
## 1       avalanche   223
## 2        blizzard    70
## 3   coastal flood     6
## 4 cold/wind chill   131
## 5     debris flow    43
## 6       dense fog    69
```

```
injuries <- aggregate(Storm.health$INJURIES, by=list(Storm.health$EVENT), FUN=sum)
names(injuries) <- c("EVENT", "total")
head(injuries)
```

```
##             EVENT total
## 1       avalanche   156
## 2        blizzard   385
## 3   coastal flood     8
## 4 cold/wind chill    24
## 5     debris flow    55
## 6       dense fog   855
```

Economic Impact

Conversely, economic impact can be aggregated without separation between property and crop damage, that should be made on a more focused report. The aggregation made below makes also a *normalization of totals* to express them as milions of dollars:

```
Storm.economy$total <- (Storm.economy$PROPDMG + Storm.economy$CROPDMG)/1000000
economy <- aggregate(Storm.economy$total, by=list(Storm.economy$EVENT), FUN=sum)
names(economy) <- c("EVENT", "total")
head(economy)
```

```
##             EVENT   total
## 1       avalanche   3.712
## 2        blizzard 532.739
## 3   coastal flood 407.319
## 4 cold/wind chill   3.144
## 5     debris flow 346.652
## 6       dense fog  20.465
```

Results

Once the data analysis has been concluded it's time to present final results. The result is presented as the top ten events of each impact.

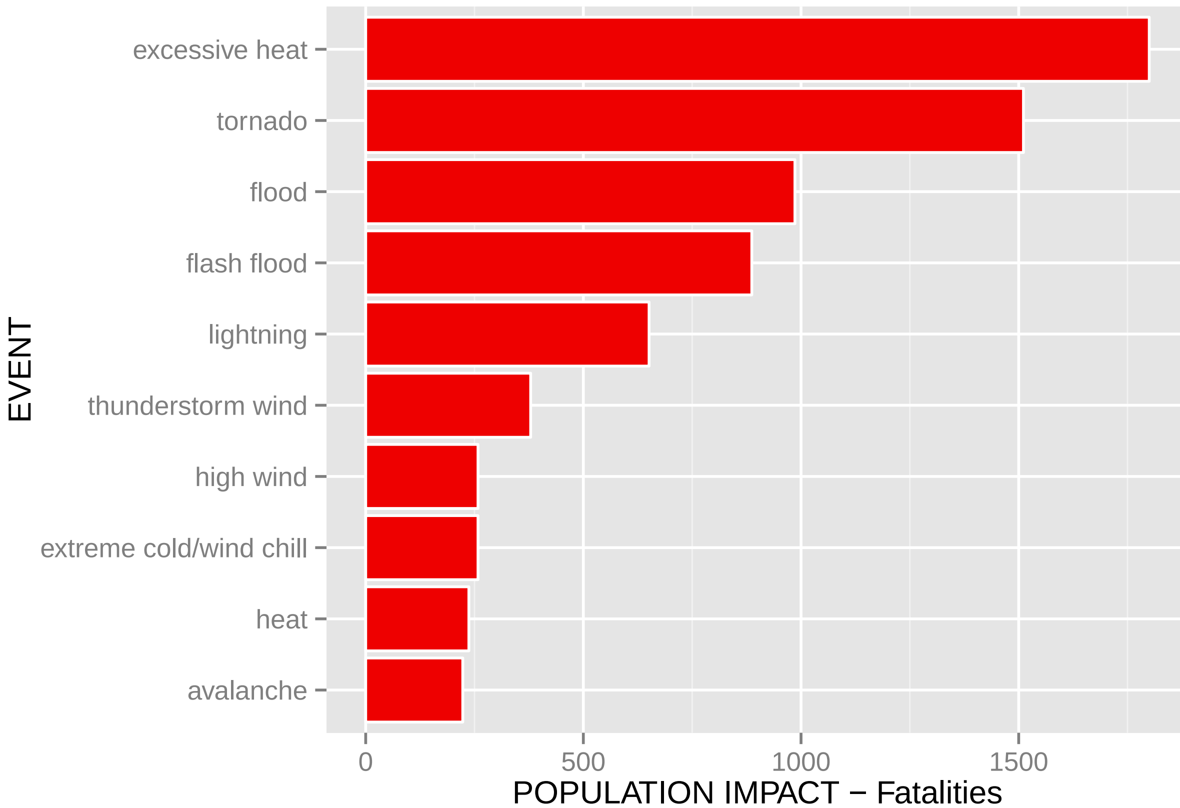
Health impact

Fatalities Top ten events that are most harmful with respect to population, resulting in fatalities:

```r
fatalities.top10 <- arrange(fatalities, desc(total))[1:10, ]
fatalities.top10
```

```
##                      EVENT total
## 1           excessive heat  1800
## 2                  tornado  1511
## 3                    flood   986
## 4              flash flood   887
## 5                lightning   651
## 6        thunderstorm wind   379
## 7  extreme cold/wind chill   258
## 8                high wind   258
## 9                     heat   237
## 10               avalanche   223
```

```r
ggplot(fatalities.top10, aes(x=reorder(EVENT, total), y=total)) +
    geom_bar(colour="white", fill="red2", stat="identity") +
    xlab("EVENT") +
    ylab("POPULATION IMPACT - Fatalities") +
    coord_flip()
```

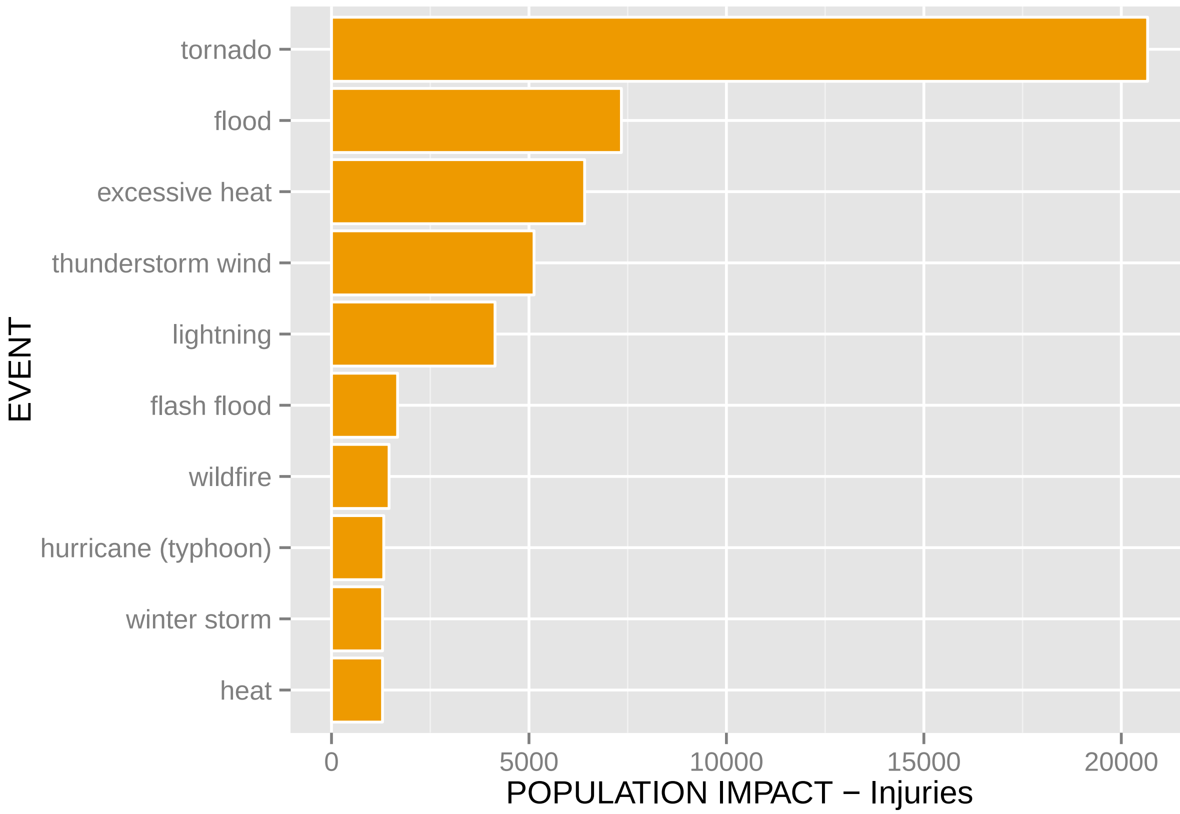


Injuries Top ten events that are most harmful with respect to population, resulting in injuries:

```
injuries.top10 <- arrange(injuries, desc(total))[1:10, ]
injuries.top10
```

```
##                  EVENT total
## 1              tornado 20667
## 2                flood  7341
## 3       excessive heat  6410
## 4    thunderstorm wind  5129
## 5            lightning  4141
## 6          flash flood  1674
## 7             wildfire  1458
## 8  hurricane (typhoon)  1323
## 9                 heat  1292
## 10        winter storm  1292
```

```
ggplot(injuries.top10, aes(x=reorder(EVENT, total), y=total)) +
    geom_bar(colour="white", fill="orange2", stat="identity") +
    xlab("EVENT") +
    ylab("POPULATION IMPACT - Injuries") +
    coord_flip()
```



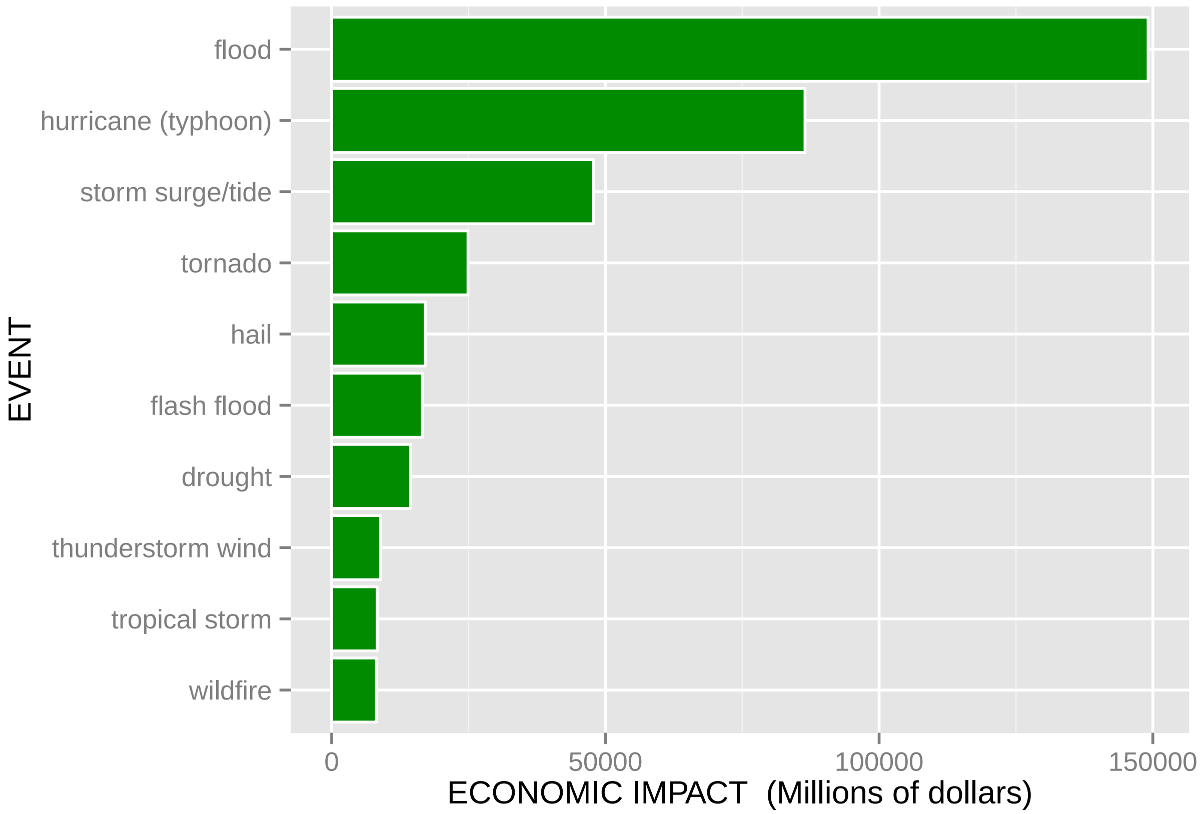
Economic impact

Top ten events that are most harmful with respect to economic impact, both crops and properties:

```
economy.top10 <- arrange(economy, desc(total))[1:10, ]
economy.top10
```

```
##                  EVENT  total
## 1                flood 149143
## 2  hurricane (typhoon)  86468
## 3     storm surge/tide  47845
## 4              tornado  24900
## 5                 hail  17092
## 6          flash flood  16558
## 7              drought  14415
## 8    thunderstorm wind   8931
## 9       tropical storm   8320
## 10            wildfire   8163
```

```
ggplot(economy.top10, aes(x=reorder(EVENT, total), y=total)) +
    geom_bar(colour="white", fill="green4", stat="identity") +
    xlab("EVENT") +
    ylab("ECONOMIC IMPACT  (Millions of dollars)") +
    coord_flip()
```


flood
hurricane (typhoon)
storm surge/tide
tornado
hail
flash flood
drought
thunderstorm wind
tropical storm
wildfire
EVENT
0
50000
100000
150000
ECONOMIC IMPACT (Millions of dollars)