

A basic NOAA Storm Database exploration

Davide Madrisan

Wednesday, April 22, 2015

Synopsis

Storms and other severe weather events can cause both public health and economic problems for communities and municipalities. Many severe events can result in fatalities, injuries, and property damage, and preventing such outcomes to the extent possible is a key concern.

This Coursera project involves exploring the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database. This database tracks 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.

The events in the database start in the year 1950 and end in November 2011. In the earlier years of the database there are generally fewer events recorded, most likely due to a lack of good records. More recent years should be considered more complete.

The basic goal of this assignment is to explore the NOAA Storm Database and answer two basic questions about severe weather events: across the United States, which types of events (1) are most harmful with respect to population health and (2) have the greatest economic consequences.

Data Processing

First we download from internet the data (a compressed *csv* file), and load the dataset into memory.

```
website <- "https://d396qusza40orc.cloudfront.net/repdata/data"
bzarchive <- "repdata-data-StormData.csv.bz2"
weburl <- paste(website, bzarchive, sep = "/")

if(!file.exists(bzarchive)) {
  switch(Sys.info()[[ 'sysname' ]],
    Windows = {
      setInternet2(use=TRUE)
      download.file(weburl, bzarchive, "internal" ),
      { download.file(weburl, bzarchive, "curl", extra = c("-L")) }
    }
  )
}

rawdata <- read.csv(bzfile(bzarchive))
```

We can now have a look at the variable names of the NOAA Storm Database:

```
names(rawdata)

## [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"
```

For our analysis we only need a few variables, namely: EVTYPE, FATALITIES, INJURIES, PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP. We create a new smaller dataset containing this subset to speed up the further data manipulation process.

```
library(dplyr)
```

```
data <- select(rawdata, EVTYPE, FATALITIES, INJURIES,
               PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP)
```

The variable EVTYPE contains all sort of types of weather events with case combinations of strings variable across the years. Make all the letters uppercase to fix this point.

```
data$EVTYPE <- toupper(data$EVTYPE)
```

Some other cleanings are required for reducing the number and grouping the similar weather events.

```
# remove leading and trailing spaces
data$EVTYPE <- gsub("^[:space:]]+|[:space:]]+$", "", data$EVTYPE)
# remove multiple spaces
data$EVTYPE <- gsub("[:space:]]+", " ", data$EVTYPE)
# make more uniform the used vocabulary
data$EVTYPE <- gsub("EXCESSIVE|EXCESSIVELY|EXTREMELY", "EXTREME", data$EVTYPE)
data$EVTYPE <- gsub("FLOODINGINGS*|FLOODING|FLOOD", "FLOODING", data$EVTYPE)
data$EVTYPE <- gsub("FLASH FLOODING.*", "FLOODING", data$EVTYPE)
data$EVTYPE <- gsub("LIGHTNING\\.|LIGHTING|LIGNTNING", "LIGHTNING", data$EVTYPE)
data$EVTYPE <- gsub("NON-", "NON ", data$EVTYPE)
data$EVTYPE <- gsub("PROLONG", "PROLONGED", data$EVTYPE)
data$EVTYPE <- gsub("RAINS", "RAIN", data$EVTYPE)
data$EVTYPE <- gsub("RIP CURRENTS", "RIP CURRENT", data$EVTYPE)
data$EVTYPE <- gsub("STORMS", "STORM", data$EVTYPE)
data$EVTYPE <- gsub("TORNDAO|TORNADOES", "TORNADO", data$EVTYPE)
data$EVTYPE <- gsub("TSTM|TH*UND*ER*[A-Z]*RMW*|THUNDERSTROM|THUDERSTORM",
                  "THUNDERSTORM", data$EVTYPE)
data$EVTYPE <- gsub("UNUSUALLY", "UNUSUAL", data$EVTYPE)
data$EVTYPE <- gsub("WILD.*FIRE.*|WILD/FOREST.*", "WILD/FOREST FIRES", data$EVTYPE)
data$EVTYPE <- gsub("WINDS|WND", "WIND", data$EVTYPE)
data$EVTYPE <- gsub("WINTER*", "WINTER", data$EVTYPE)
data$EVTYPE <- gsub("WARMTH", "WARM", data$EVTYPE)
# grouping some events
data$EVTYPE <- gsub("^BLIZZARD.*|ICE STORM", "BLIZZARD", data$EVTYPE)
data$EVTYPE <- gsub("^COASTAL.*|.*CSTL .*", "COASTAL EROSION/FLOODING/STORM ",
                  data$EVTYPE)
data$EVTYPE <- gsub("EXTREME COLD.*|EXTENDED COLD.*", "EXTREME COLD", data$EVTYPE)
data$EVTYPE <- gsub("^DRY.*", "DRY CONDITIONS", data$EVTYPE)
data$EVTYPE <- gsub("^FLOODING.*", "FLOODING", data$EVTYPE)
data$EVTYPE <- gsub("^FREEZE|^FREEZING.*|^FROST.*",
                  "FREEZING FOG/RAIN/SLEET/SNOW", data$EVTYPE)
data$EVTYPE <- gsub("HAIL.*", "HAIL", data$EVTYPE)
```

```

data$EVTYPE <- gsub("DROUGHT|EXTREME HEAT.*|^HEAT.*", "EXTREME HEAT", data$EVTYPE)
data$EVTYPE <- gsub("HEAVY RAIN.*", "HEAVY RAIN", data$EVTYPE)
data$EVTYPE <- gsub("HURRICANE.*", "HURRICANE", data$EVTYPE)
data$EVTYPE <- gsub("HEAVY SNOW.*|^SNOW.*|EXCESSIVE SNOW", "HEAVY SNOW/ICE",
  data$EVTYPE)
data$EVTYPE <- gsub("LIGHTNING.*", "LIGHTNING", data$EVTYPE)
data$EVTYPE <- gsub("^MARINE.*", "MARINE THUNDERSTORM/ACCIDENT", data$EVTYPE)
data$EVTYPE <- gsub("RAIN.*|PROLONGEDED RAIN", "RAIN", data$EVTYPE)
data$EVTYPE <- gsub("RIP CURRENT.*|HEAVY SURF.*|HIGH SURF.*", "HEAVY SURF", data$EVTYPE)
data$EVTYPE <- gsub("SLEET.*", "SLEET", data$EVTYPE)
data$EVTYPE <- gsub("VOLCANIC.*", "VOLCANIC", data$EVTYPE)
data$EVTYPE <- gsub("THUNDERSTORM.*|SEVERE THUNDERSTORM", "THUNDERSTORM", data$EVTYPE)
data$EVTYPE <- gsub("TORNADO.*", "TORNADO", data$EVTYPE)
data$EVTYPE <- gsub("TROPICAL STORM.*", "TROPICAL STORM", data$EVTYPE)
data$EVTYPE <- gsub("UNSEASONAL.*|^UNSEASONABLE[RY].*|^UNUSUAL.*",
  "UNUSUAL WEATHER CONDITION", data$EVTYPE)
data$EVTYPE <- gsub("HIGH WIND.*|STRONG WIND.*|^WIND.*", "HIGH WIND", data$EVTYPE)
data$EVTYPE <- gsub("^WATERSPOUT.*|WATER SPOUT", "WATERSPOUT", data$EVTYPE)
data$EVTYPE <- gsub("^WINTER.*", "WINTER STORM/WIND", data$EVTYPE)
data$EVTYPE <- gsub("^NONE|^SUMMARY.*", "?", data$EVTYPE)

```

According to the [National Weather Service Instruction](#) documentation, property (PROPDMG) and crop (CROPDMG) damages estimates are rounded to three significant digits, followed by an alphabetical character signifying the magnitude of the number (provided by CROPDMGEXP and PROPDMGEXP respectively), i.e., 1.55B for \$1,550,000,000. Alphabetical characters used to signify magnitude include K for thousands, M for millions, and B for billions.

Some other symbols are present in these variables with an unknown meaning. We will ignore them, except the blank multiplication factor that we will consider as a 1. We can note that the remaining entries represent a very small percentage of the entire dataset.

```
table(toupper(data$PROPDMGEXP))
```

```
##
##      -      ?      +      0      1      2      3      4      5
## 465934    1      8      5    216     25     13      4      4     28
##      6      7      8      B      H      K      M
##      4      5      1     40      7 424665 11337
```

```
table(toupper(data$CROPDMGEXP))
```

```
##
##      ?      0      2      B      K      M
## 618413    7     19      1      9 281853 1995
```

We now replace these values by their power of ten equivalent, in millions of dollars.

```

factor_normalize <- function(base, magnitude) {
  base * switch(tolower(magnitude), ' '=1e-06, k=1e-03, m=1, b=1e+03, 0)
}

data$PROPDMG_MDOLLAR <- mapply(factor_normalize, data$PROPDMG, data$PROPDMGEXP)
data$CROPDMG_MDOLLAR <- mapply(factor_normalize, data$CROPDMG, data$CROPDMGEXP)

```

Results

We can now answer the two questions raised in the introduction.

Which types of events are most harmful with respect to population health

We extract two distinct datasets containing the list of *fatalities* and *injuries* sorted by weather event in descending order. We'll only display the top fifteen events.

```
fatalities <- select(data, EVTYPE, FATALITIES) %>%
  group_by(EVTYPE) %>%
  summarise_each(funs(sum)) %>%
  arrange(desc(FATALITIES)) %>%
  slice(1:15)

injuries <- select(data, EVTYPE, INJURIES) %>%
  group_by(EVTYPE) %>%
  summarise_each(funs(sum)) %>%
  arrange(desc(INJURIES)) %>%
  slice(1:15)
```

List of the weather events that caused the most fatalities.

```
fatalities
```

```
## Source: local data frame [15 x 2]
##
##           EVTYPE FATALITIES
## 1          TORNADO         5658
## 2    EXTREME HEAT         3117
## 3        FLOODING         1513
## 4        LIGHTNING          817
## 5        HEAVY SURF          734
## 6    THUNDERSTORM          711
## 7         HIGH WIND          429
## 8    EXTREME COLD          288
## 9 WINTER STORM/WIND          278
## 10        AVALANCHE          224
## 11         BLIZZARD          190
## 12   HEAVY SNOW/ICE          142
## 13        HURRICANE          135
## 14        HEAVY RAIN           98
## 15   COLD/WIND CHILL           95
```

List of the weather events that caused the most severe health injuries.

```
injuries
```

```
## Source: local data frame [15 x 2]
##
##           EVTYPE INJURIES
## 1          TORNADO     91364
```

## 2	THUNDERSTORM	9508
## 3	EXTREME HEAT	9178
## 4	FLOODING	8591
## 5	LIGHTNING	5232
## 6	BLIZZARD	2780
## 7	WINTER STORM/WIND	1891
## 8	HIGH WIND	1859
## 9	WILD/FOREST FIRES	1606
## 10	HAIL	1361
## 11	HURRICANE	1328
## 12	HEAVY SNOW/ICE	1137
## 13	HEAVY SURF	773
## 14	FOG	734
## 15	DUST STORM	440

The plot of these two datasets follow.

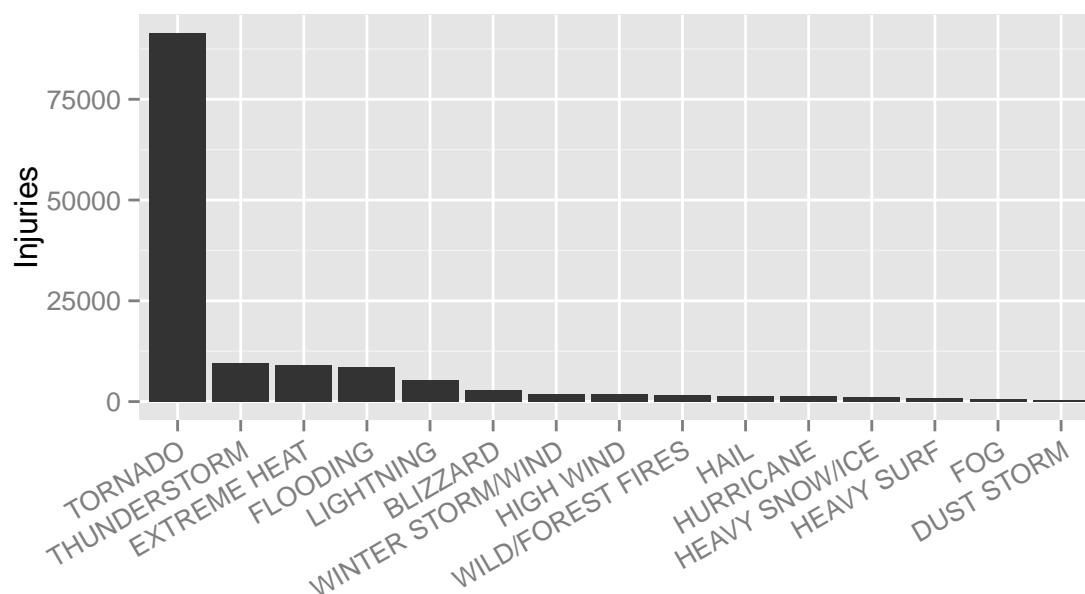
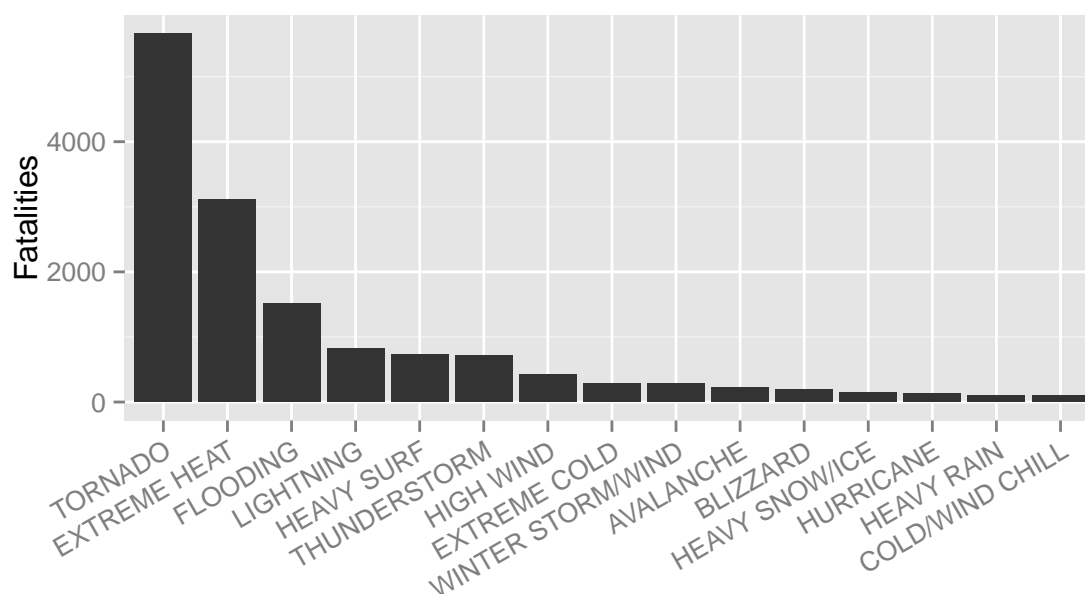
```
library(ggplot2)
library(gridExtra)
```

```
p1 <- ggplot(fatalities, aes(x=reorder(EVTYPE, -FATALITIES), y=FATALITIES)) +
  geom_bar(stat="identity") +
  ylab("Fatalities") +
  theme(axis.text.x=element_text(angle=30,hjust=1),
        axis.title.x=element_blank())

p2 <- ggplot(injuries, aes(x=reorder(EVTYPE, -INJURIES), y=INJURIES)) +
  geom_bar(stat="identity") +
  xlab("Event Type") + ylab("Injuries") +
  theme(axis.text.x=element_text(angle=30,hjust=1),
        axis.title.x=element_blank())

grid.arrange(p1, p2,
              main="Most Harmful Events with Respect to Population Health")
```

Most Harmful Events with Respect to Population Health



Which types of events have the greatest economic consequences

We summarize our dataset by the variable EVTYPE and display the top fifteen weather events.

```
damages <- select(data, EVTYPE, contains("_MDOLLAR")) %>%
  group_by(EVTYPE) %>%
  summarise(CROPDGMG_MDOLLAR=round(sum(CROPDGMG_MDOLLAR), 0),
            PROPDGMG_MDOLLAR=round(sum(PROPDGMG_MDOLLAR), 0),
            DMG_MDOLLAR=round(sum(CROPDGMG_MDOLLAR + PROPDGMG_MDOLLAR), 0)) %>%
  arrange(desc(DMG_MDOLLAR)) %>% slice(1:15)
```

damages

```
## Source: local data frame [15 x 4]
##
##           EVTYPE CROPDMG_MDOLLAR PROPDMG_MDOLLAR DMG_MDOLLAR
## 1      FLOODING           7316          161690      169006
## 2     HURRICANE           5515           84756       90271
## 3      TORNADO            417           58542       58959
## 4   STORM SURGE             0           43324       43324
## 5         HAIL           3026           15974       19000
## 6   EXTREME HEAT          14877            1066       15943
## 7   THUNDERSTORM          1243           10970       12213
## 8   RIVER FLOODING         5029            5119       10148
## 9     BLIZZARD           5134            4605        9739
## 10 WILD/FOREST FIRES         403            8492        8895
## 11  TROPICAL STORM           695            7714        8409
## 12     HIGH WIND           757            6192        6949
## 13 WINTER STORM/WIND          47            6776        6823
## 14 STORM SURGE/TIDE           1            4641        4642
## 15     HEAVY RAIN           796            3231        4027
```

And plot the result.

```
ggplot(select(damages, EVTYPE, PROPDMG_MDOLLAR),
  aes(x=reorder(EVTYPE, -PROPDGM_MDOLLAR), y=PROPDGM_MDOLLAR)) +
  geom_bar(stat="identity") + ylab("Damages (million dollars)") +
  theme(axis.text.x=element_text(angle=30,hjust=1),
  axis.title.x=element_blank()) +
  ggtitle("Most Harmful Events with Respect to Economy")
```

