Analysis of U.S. storm event data for impact on population health and economy

NS

21/06/2020

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 analysis looks at the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database (1950-2011). 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 analysis seeks to answer the following questions:
1. Across the United States, which types of events are most harmful with respect to population health?
2. Across the United States, which types of events have the greatest economic consequences?

Data load

The U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database used in this analysis was downloaded from the course website on 22/06/2020.

The database tracks storms and weather events in the US and contains data from 1950 to Nov 2011. Data from earlier years is less complete.

The National Weather Service Storm Data Documentation provides the codebook for the data.

Load required libraries

```
library(R.utils)
library(lubridate)
library(stringr)
library(ggplot2)
library(knitr)
```

Download the load the data. Please *note* that the uncompressed data is ~500MB.

```
url <- 'https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2'
fileBz2 <- './stormData.csv.bz2'
fileCsv <- './stormData.csv'
if(!file.exists(fileBz2)) {
  download.file(url, fileBz2, mode='wb', method='curl')
```

```
}
if(!file.exists(fileCsv)) {
  bunzip2(fileBz2, fileCsv, remove=FALSE)
}
data <- read.csv(fileCsv, header = TRUE)
str(data)
```

```
## '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 ...
```

Data processing

Cleaning

Filter the data for variables that we are interested in
* EVTYPE - Event type * BGN_DATE - Event start date * FATALITIES - number of recorded fatalities *

INJURIES - number of recorded injuries * PROPDMG - property damage (rounded to 3 significant digits) * PROPDMGEXP - property damage magnitude * CROPDMG - crop damage (rounded to 3 significant digits) * CROPDMGEXP - crop damage magnitude

```
data <- data %>%
  select(EVTYPE, BGN_DATE, FATALITIES, INJURIES, PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP)
```

Set BGN_DATA to date type and extract year

```
data <- data %>%
  mutate(BGN_DATE = mdy_hms(BGN_DATE)) %>%
  mutate(year = year(BGN_DATE))
```

See how many unique EVTYPES we have

```
nlevels(as.factor(data$EVTYPE))
```

```
## [1] 985
```

```
levels(as.factor(data$EVTYPE))[1:20]
```

```
##  [1] "   HIGH SURF ADVISORY"  " COASTAL FLOOD"         " FLASH FLOOD"
##  [4] " LIGHTNING"             " TSTM WIND"             " TSTM WIND (G45)"
##  [7] " WATERSPOUT"            " WIND"                  "?"
## [10] "ABNORMAL WARMTH"        "ABNORMALLY DRY"         "ABNORMALLY WET"
## [13] "ACCUMULATED SNOWFALL"   "AGRICULTURAL FREEZE"    "APACHE COUNTY"
## [16] "ASTRONOMICAL HIGH TIDE" "ASTRONOMICAL LOW TIDE"  "AVALANCE"
## [19] "AVALANCHE"              "BEACH EROSIN"
```

Exploring EVTYPE data show duplication due to misspellings, case differences, multiple versions ..etc. Examples include `AVALANCE` vs `AVALANCHE` and `Coastal Flooding` vs `COASTAL FLOODING`. We try to normalise the data by moving similar event types into the same category. This will help when aggregating data later

```
data$evenType <- data$EVTYPE
data$evenType <- str_trim(data$evenType)
data$evenType <- str_to_upper(data$evenType)
data <- data %>%
  mutate(eventType = case_when(
    str_detect(evenType, "AVALANC") ~ "AVALANCHE",
    str_detect(evenType, "FLOOD")  ~ "FLOOD",
    str_detect(evenType, "FLOOODING")  ~ "FLOOD",
    str_detect(evenType, "EXCESSIVE WETNESS")  ~ "FLOOD",
    str_detect(evenType, "EXTREMELY WET")  ~ "FLOOD",
    str_detect(evenType, "WIND")  ~ "WIND",
    str_detect(evenType, "SNOW")  ~ "COLD",
    str_detect(evenType, "BLIZZARD")  ~ "COLD",
    str_detect(evenType, "COLD")  ~ "COLD",
    str_detect(evenType, "FREEZ")  ~ "COLD",
    str_detect(evenType, "FROST")  ~ "COLD",
    str_detect(evenType, "WINTER")  ~ "COLD",
    str_detect(evenType, "WINTRY")  ~ "COLD",
```

```
    str_detect(evenType, "UNSEASONAL LOW TEMP")  ~ "COLD",
    str_detect(evenType, "UNSEASONABLY COOL")  ~ "COLD",
    str_detect(evenType, "HYPOTHERMIA")  ~ "COLD",
    str_detect(evenType, "DRY")  ~ "DRY",
    str_detect(evenType, "DROUGHT")  ~ "DRY",
    str_detect(evenType, "DUST")  ~ "DUST",
    str_detect(evenType, "HAIL")  ~ "HAIL",
    str_detect(evenType, "RAIN")  ~ "RAIN",
    str_detect(evenType, "HURRICANE")  ~ "HURRICANE",
    str_detect(evenType, "ICE")  ~ "ICE",
    str_detect(evenType, "ICY")  ~ "ICE",
    str_detect(evenType, "LIGHTNING")  ~ "LIGHTNING",
    str_detect(evenType, "SLIDE")  ~ "MUDSLIDE",
    str_detect(evenType, "WARM")  ~ "HEAT",
    str_detect(evenType, "HOT WEATHER")  ~ "HEAT",
    str_detect(evenType, "TEMPERATURE")  ~ "HEAT",
    str_detect(evenType, "HOT SPELL")  ~ "HEAT",
    str_detect(evenType, "HOT PATTERN")  ~ "HEAT",
    str_detect(evenType, "UNSEASONABLY HOT")  ~ "HEAT",
    str_detect(evenType, "RIP")  ~ "RIP CURRENT",
    str_detect(evenType, "SLIDE")  ~ "MUDSLIDE",
    str_detect(evenType, "RAIN")  ~ "RAIN",
    str_detect(evenType, "SHOWER")  ~ "RAIN",
    str_detect(evenType, "WET WEATHER")  ~ "RAIN",
    str_detect(evenType, "WET YEAR")  ~ "RAIN",
    str_detect(evenType, "WET MONTH")  ~ "RAIN",
    str_detect(evenType, "PRECIPITATION")  ~ "RAIN",
    str_detect(evenType, "PRECIPATATION")  ~ "RAIN",
    str_detect(evenType, "UNSEASONABLY WET")  ~ "RAIN",
    str_detect(evenType, "LIGHTING")  ~ "LIGHTNING",
    str_detect(evenType, "LIGNTNING")  ~ "LIGHTNING",
    str_detect(evenType, "TORNADO")  ~ "TORNADO",
    str_detect(evenType, "TORNDAO")  ~ "TORNADO",
    str_detect(evenType, "SPOUT")  ~ "TORNADO",
    str_detect(evenType, "TYPHOON")  ~ "TORNADO",
    str_detect(evenType, "STORM")  ~ "STORM",
    str_detect(evenType, "TSTM")  ~ "STORM",
    str_detect(evenType, "TROPICAL DEPRESSION")  ~ "STORM",
    str_detect(evenType, "VOLC")  ~ "VOLCANIC",
    str_detect(evenType, "SUMMARY")  ~ "SUMMARY",
    str_detect(evenType, "FUNNEL")  ~ "FUNNEL",
    str_detect(evenType, "HEAT")  ~ "HEAT",
    str_detect(evenType, "FIRE")  ~ "FIRE",
    str_detect(evenType, "URBAN")  ~ "URBAN",
    str_detect(evenType, "STREAM")  ~ "STREAM",
    str_detect(evenType, "SEAS")  ~ "HIGH SWELLS",
    str_detect(evenType, "SURF")  ~ "HIGH SWELLS",
    str_detect(evenType, "HIGH WAVES")  ~ "HIGH SWELLS",
    str_detect(evenType, "HIGH TIDES")  ~ "HIGH SWELLS",
    str_detect(evenType, "SWELLS")  ~ "HIGH SWELLS",
    str_detect(evenType, "HAZARDOUS SURF")  ~ "HIGH SWELLS",
    str_detect(evenType, "MARINE MISHAP")  ~ "MARINE ACCIDENT",
    TRUE ~ evenType
```

```
  ))

nlevels(as.factor(data$eventType))
```

```
## [1] 82
```

We managed to reduced the number of unique event types to 82.

We explore `PROPDMG` and `CROPDMG` for NA values

```
sum(is.na(data$PROPDMG))
```

```
## [1] 0
```

```
sum(is.na(data$CROPDMG))
```

```
## [1] 0
```

Exploring `PROPDMHEXP` and `CROPDMGEXP`, we see that alphabetical characters are used to signify the magnitude e.g. “K” for thousands, “M” for millions, and “B” for billions.

```
levels(as.factor(data$PROPDMGEXP))
```

```
##  [1] ""  "-" "?" "+" "0" "1" "2" "3" "4" "5" "6" "7" "8" "B" "h" "H" "K" "m" "M"
```

```
levels(as.factor(data$CROPDMGEXP))
```

```
## [1] ""  "?" "0" "2" "B" "k" "K" "m" "M"
```

We transform the alphabetical characters to numeric exponenents to make it calculations easier

```
convertExp <- function(alphStr) {
  case_when(
    str_detect(alphStr, fixed("-")) ~ 10^0,
    str_detect(alphStr, fixed("?")) ~ 10^0,
    str_detect(alphStr, fixed("+")) ~ 10^0,
    str_detect(alphStr, fixed("0")) ~ 10^0,
    str_detect(alphStr, fixed("1")) ~ 10^1,
    str_detect(alphStr, fixed("2")) ~ 10^2,
    str_detect(alphStr, fixed("3")) ~ 10^3,
    str_detect(alphStr, fixed("4")) ~ 10^4,
    str_detect(alphStr, fixed("5")) ~ 10^5,
    str_detect(alphStr, fixed("6")) ~ 10^6,
    str_detect(alphStr, fixed("7")) ~ 10^7,
    str_detect(alphStr, fixed("8")) ~ 10^8,
    str_detect(alphStr, fixed("H")) ~ 10^2,
    str_detect(alphStr, fixed("K")) ~ 10^3,
    str_detect(alphStr, fixed("M")) ~ 10^6,
    str_detect(alphStr, fixed("B")) ~ 10^9,
    TRUE ~ 1
```

```
  )
}

data$PROPDMGEXP <- str_to_upper(data$PROPDMGEXP)
data$CROPDMGEXP <- str_to_upper(data$CROPDMGEXP)
data <- data %>%
  mutate( propDmgExponent = convertExp(PROPDMGEXP)) %>%
  mutate( cropDmgExponent = convertExp(CROPDMGEXP))
```

Exploring population health

We calculate the total fatalities and injuries by event type and display the top 5 events that result in harm to population

```
popHealth <- data %>%
  group_by(eventType) %>%
  summarize(fatalities = sum(FATALITIES), injuries = sum(INJURIES), totalHealthImpact = sum(FATALITIES +
  arrange(desc(totalHealthImpact)) %>%
  top_n(n=5, wt=totalHealthImpact)

kable(popHealth, format="markdown", col.names=c('Event Type', 'Fatalities', 'Injuries', 'Total'))
```

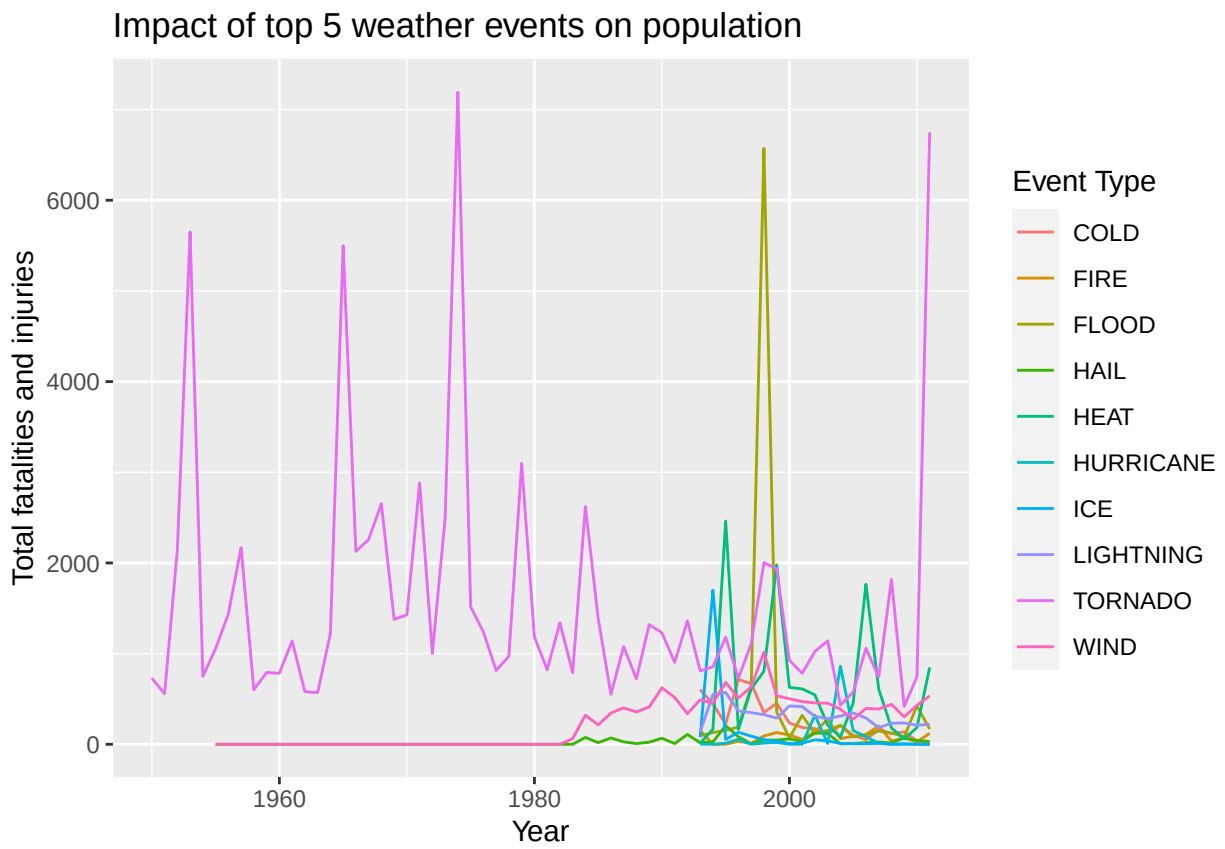
Event Type	Fatalities	Injuries	Total
TORNADO	5639	91441	97080
WIND	1451	11498	12949
HEAT	3150	9228	12378
FLOOD	1525	8604	10129
LIGHTNING	817	5232	6049

Weighting fatalities more then injuries will not result in a change to ranking of events. As such we will continue to give fatalities and injuries equal weighting.

We explore the total harm caused to the population by year for the top 5 identified event types to see if there are any trends

```
dataHarm <- data %>%
  filter(eventType %in% popHealth$eventType) %>%
  group_by(year, eventType) %>%
  summarize(harm = sum(FATALITIES + INJURIES))

g <- ggplot(dataHarm, aes(x=year, y=harm, color=eventType)) +
  geom_line() +
  labs(x="Year", y="Total fatalities and injuries",
       title="Impact of top 5 weather events on population",
       color="Event Type")
g
```



We see that there was a major flooding event in the late 90s that caused injuries and fatalities. Collection of data for heat related events only started in the mid 90s. Tornadoes are a consistent threat to population health.

Exploring economic consequences of weather events

We aggregate the data to see which weather events had the greatest economic consequence. We choose not to differentiate between property damage and crop damage

```
ecoData <- data %>%
  mutate(propDmgB = PROPDMG * propDmgExponent / 10^9) %>%
  mutate(cropDmgB = CROPDMG * cropDmgExponent / 10^9) %>%
  mutate(totalDmgB = propDmgB + cropDmgB) %>%
  group_by(eventType) %>%
  summarise(totalDmgB = sum(totalDmgB)) %>%
  arrange(desc(totalDmgB)) %>%
  top_n(n=5, wt=totalDmgB)

kable(ecoData, format="markdown", col.names=c('Event Type', 'Total Damage (Billions)'))
```

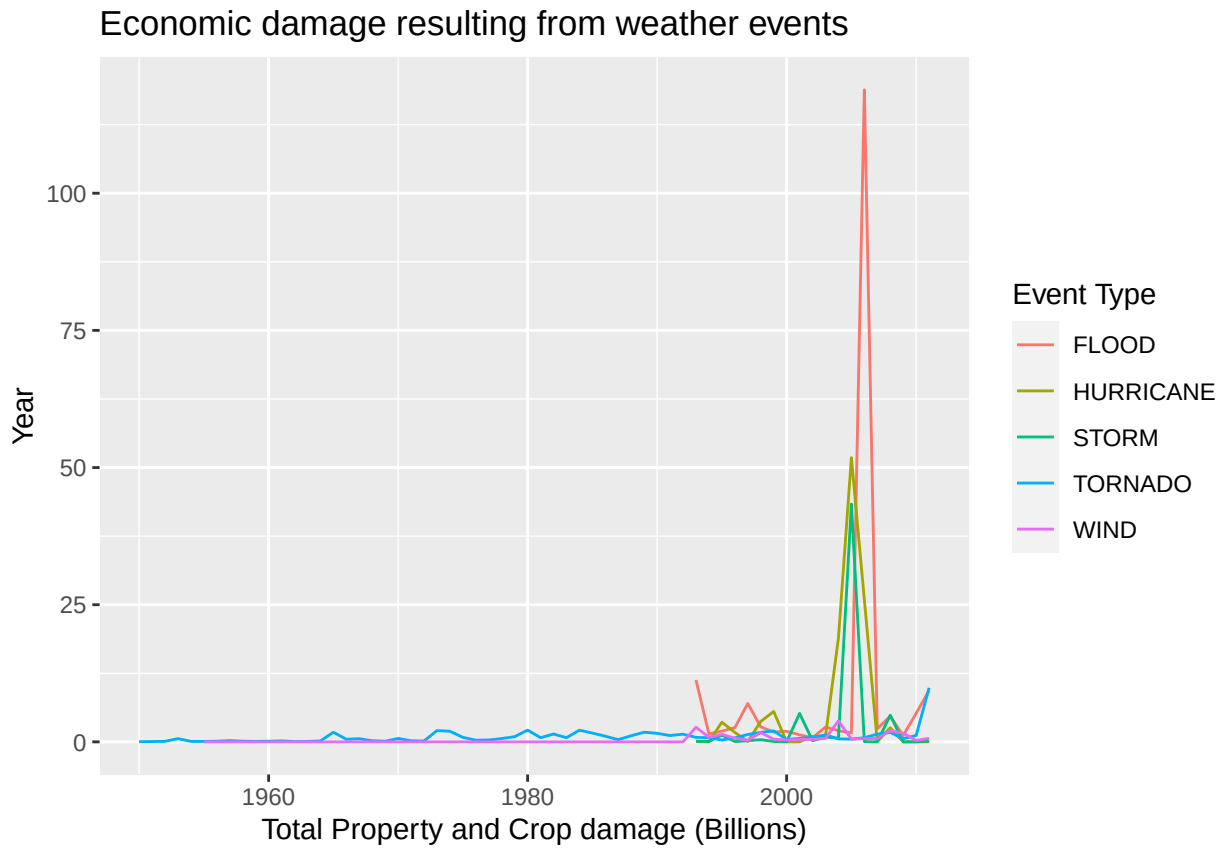
Event Type	Total Damage (Billions)
FLOOD	180.73432
HURRICANE	90.16147
TORNADO	58.02891
STORM	57.60332

Event Type	Total Damage (Billions)
WIND	19.99114

We again explore the top 5 causes of negative economic consequence by year to see if there are any trends or outliers

```
ecoData2 <- data %>%
  filter(eventType %in% ecoData$eventType) %>%
  mutate(propDmgB = PROPDMG * propDmgExponent / 10^9) %>%
  mutate(cropDmgB = CROPDMG * cropDmgExponent / 10^9) %>%
  mutate(totalDmgB = propDmgB + cropDmgB) %>%
  group_by(year, eventType) %>%
  summarise(totalDmgB = sum(totalDmgB))


g <-ggplot(ecoData2, aes(x=year, y=totalDmgB, color=eventType)) +
  geom_line() +
  labs(x="Total Property and Crop damage (Billions)", y="Year", color="Event Type", title="Economic dama
g
```



We see that there was flooding event in mid 2000 that resulted severe property and crop damage

Result

Our exploration of the NOAA storm database found that:
1. Tornados has the greatest impact on U.S. population health resulting from fatalities and injuries
2. Flooding has ths the greatest economic impact resulting from property and crop damage