Analysis of storm impact on population health and economy

Synopsis

In this analysis report we look at the National Oceanic and Atmospheric Administration's (NOAA) storm database, to address questions with respect to the impact of weather events on the economy and the population's health. More specifically, we will identify the types of events that are most harmful.

Summarizing, we have found that thunderstorm winds, excessive heat, and flood, are most harmful for the population's health. Also, we have found that flood, hurricane's, and tornado's have the greatest economic consequences.

The remainder of this report details how this information was extracted, and gives more detailed break-down information on impact of weather events on population health and economy.

Data processing

In this section we describe the steps that where taking to get from the raw data to the plots and conclusions in this report.

Downloading and extracting the data.

We create a separate data dir, download the data and extract the file for further processing. Additionally some checks are performed to prevent re-downloading and unpacking the data unnecesarily.

```
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)
library(reshape2)

url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
destFile <- "data/stormData.csv.bz2"
dataFile <- "data/stormData.csv"

if (!file.exists("data")) dir.create("data")
if (!file.exists(destFile)) download.file(url, destFile, method="curl")
if (!file.exists(dataFile)) system(paste("bunzip2 ", destFile, sep=""))
stormData <- read.csv(dataFile, stringsAsFactors=FALSE)
```

Cleaning the data.

```
# Fix some issues in the data I found while manually checking results
# First, we make sure that all event types are upper case.
fixNamingIssues <- function(x) {
  x <- toupper(x) # Make sure that all event types are upper case first
  x <- gsub("^\\s+|\\s+$", "", x) # Trim whitespace

  # The following statements unify a number of names which are very similar.
  # The strategy I chose was to rename the events with the lowest fequency to
  # those with the highest frequency.
  if (x == "WINTER WEATHER MIX") x <- "WINTER WEATHER"
  else if (x == "WHIRLWIND") x <- "WIND"
  else if (x == "WINTER STORMS") x <- "WINTER STORM"
  else if (x == "WATERSPOUT TORNADO") x <- "WATERSPOUT"
  else if (x == "URBAN AND SMALL STREAM FLOODIN") x <- "URBAN/SML STREAM FLD"
  else if (x == "THUNDERTORM WINDS") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORM WINDSS") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORM WINDS 13") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORM WINDS") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORM WINDS/HAIL") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORM  WINDS") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORM WIND G52") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORM WIND (G40)") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORMW") x <- "THUNDERTORM WIND"
  else if (x == "TSTM WIND/HAIL") x <- "THUNDERTORM WIND"
  else if (x == "TSTM WIND (G45)") x <- "THUNDERTORM WIND"
  else if (x == "TSTM WIND (G40)") x <- "THUNDERTORM WIND"
  else if (x == "TSTM WIND (G35)") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORM WIND") x <- "THUNDERTORM WIND"
  else if (x == "THUNDERSTORMS WINDS") x <- "THUNDERTORM WIND"
  else if (x == "TSTM WIND") x <- "THUNDERTORM WIND"
  else if (x == "STRONG WINDS") x <- "STRONG WIND"
  else if (x == "STORM SURGE/TIDE") x <- "STORM SURGE"
  else if (x == "SNOW SQUALLS") x <- "SNOW SQUALL"
  else if (x == "RIVER FLOODING") x <- "RIVER FLOOD"
  else if (x == "RIP CURRENTS/HEAVY SURF") x <- "RIP CURRENT"
  else if (x == "RIP CURRENTS") x <- "RIP CURRENT"
  else if (x == "RECORD HEAT") x <- "EXCESSIVE HEAT"
  else if (x == "RECORD/EXCESSIVE HEAT") x <- "EXCESSIVE HEAT"
  else if (x == "RECORD COLD") x <- "EXTREME COLD"
  else if (x == "EXTREME HEAT") x <- "EXCESSIVE HEAT"
  else if (x == "HEAT WAVES") x <- "HEAT"
  else if (x == "HEAT WAVE DROUGHT") x <- "HEAT"
  else if (x == "HEAT WAVE") x <- "HEAT"
  else if (x == "EXTREME HEAT") x <- "HEAT"
  else if (x == "HEAVY RAINS") x <- "HEAVY RAIN"
  else if (x == "WINDS") x <- "WIND"
  else if (x == "WIND STORM") x <- "WIND"
  else if (x == "WILD/FOREST FIRE") x <- "WILDFIRE"
  else if (x == "WILD FIRES") x <- "WILDFIRE"
  else if (x == "UNSEASONABLY WARM AND DRY") x <- "UNSEASONABLY WARM"
  else if (x == "AVALANCE") x <- "AVALANCHE"
```

```
  else if (x == "COASTAL FLOODING") x <- "COASTAL FLOOD"
  else if (x == "COASTAL FLOODING/EROSION") x <- "COASTAL FLOOD"
  else if (x == "COASTALSTORM") x <- "COASTAL STORM"
  else if (x == "COLD AND SNOW") x <- "COLD"
  else if (x == "COLD TEMPERATURE") x <- "COLD"
  else if (x == "COLD WAVE") x <- "COLD"
  else if (x == "COLD WEATHER") x <- "COLD"
  else if (x == "COLD/WINDS") x <- "COLD/WIND CHILL"
  else if (x == "EXCESSIVE HEAT") x <- "EXCESSIVE HEAT"
  else if (x == "DROUGHT") x <- "EXCESSIVE HEAT"
  else if (x == "DRY MIRCOBURST WINDS") x <- "DRY MICROBURST"
  else if (x == "DUST DEVIL") x <- "DUST STORM"
  else if (x == "WINTRY MIX") x <- "WINTER WEATHER"
  else if (x == "WINTER WEATHER/MIX") x <- "WINTER WEATHER"
  else if (x == "WINTER STORM HIGH WINDS") x <- "WINTER STORM"
  else if (x == "WATERSPOUT/TORNADO") x <- "WATERSPOUT"
  else if (x == "WARM WEATHER") x <- "EXCESSIVE HEAT"
  else if (x == "TROPICAL STORM GORDON") x <- "TROPICAL STORM"
  else if (x == "TORNADO F3") x <- "TORNADO"
  else if (x == "TORNADO F2") x <- "TORNADO"
  else if (x == "TORNADOES, TSTM WIND, HAIL") x <- "TORNADO"
  else if (x == "HYPOTHERMIA") x <- "HYPOTHERMIA/EXPOSURE"
  else if (x == "HYPERTHERMIA/EXPOSURE") x <- "HYPOTHERMIA/EXPOSURE"
  else if (x == "LIGHTNING.") x <- "LIGHTNING"
  else if (x == "LANDSLIDES") x <- "LANDSLIDE"
  else if (x == "ICE ROADS") x <- "ICE"
  else if (x == "ICE ON ROAD") x <- "ICE"
  else if (x == "HEAVY SNOW SHOWER") x <- "HEAVY SNOW"
  else if (x == "HEAVY SNOW/ICE") x <- "HEAVY SNOW"
  else if (x == "HEAVY SNOW/BLIZZARD/AVALANCHE") x <- "HEAVY SNOW"
  else if (x == "HEAVY SNOW AND HIGH WINDS") x <- "HEAVY SNOW"
  else if (x == "GUSTY WIND") x <- "GUSTY WINDS"
  else if (x == "GLAZE/ICE STORM") x <- "GLAZE"
  else if (x == "FOG AND COLD TEMPERATURES") x <- "FOG"
  else if (x == "FLOOD/RIVER FLOOD") x <- "FLOOD"
  else if (x == "FLOODING") x <- "FLOOD"
  else if (x == "FLOOD & HEAVY RAIN") x <- "FLOOD"
  else if (x == "FLOOD/FLASH FLOOD") x <- "FLOOD"
  else if (x == "FLASH FLOODS") x <- "FLASH FLOOD"
  else if (x == "FLASH FLOODING/FLOOD") x <- "FLASH FLOOD"
  else if (x == "FLASH FLOODING") x <- "FLASH FLOOD"
  else if (x == "FLASH FLOOD/FLOOD") x <- "FLASH FLOOD"
  x
}

stormData$EVTYPE <- sapply(stormData$EVTYPE, fixNamingIssues)
stormData$EVTYPE <- as.factor(stormData$EVTYPE)

# Merge the date and start time columns into one column

# First get rid of the time in the date field
stormData$DATE_TIME <- sub("\\s+0:00:00", "", stormData$BGN_DATE)
# Now concatenate all the date/time info and transform it into a real date object
```

```
stormData$DATE_TIME <- paste(stormData$DATE_TIME, stormData$BGN_TIME)

convertToDateTime <- function(dtString, tz) {
  strftime(strptime(dtString, tz=tz, format="%m/%d/%Y %H%M"), tz=tz)
}

stormData$DATETIME <- mapply(convertToDateTime,
                             stormData$DATE_TIME,
                             stormData$TIME_ZONE,
                             SIMPLIFY = TRUE)
```

Population health

To get insight in the impact of events on the populations health, we look at events that have injuries and fatalities. First we do some basic checks on the sanity of the data.

```
sum(is.na(stormData$FATALITIES))
```

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

```
sum(is.na(stormData$INJURIES))
```

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

Good, we don't have missing data for fatalities nor for injuries. Now we can subset the data to get only the events that resulted in injuries, fatalities or both.

```
stormDataHealth <- stormData[stormData$INJURIES > 0 |  stormData$FATALITIES > 0,]
```

We now reshape the data to make it more suitable for plotting. We want to know which events have lead to how many injuries, fatalities and the total number of victims.

```
healthSummary <- stormDataHealth %>%
  group_by(EVTYPE) %>%
  select(EVTYPE, FATALITIES, INJURIES) %>%
  summarise(
    FATALITIES = sum(FATALITIES),
    INJURIES = sum(INJURIES),
    ALL = FATALITIES + INJURIES
  ) %>%
  arrange(desc(ALL), desc(FATALITIES), desc(INJURIES))
```

There are many event types, and many of them result in only a handful of victims. In order to find the areas where policies, education and technological advancements have a maximum impact on reducing victims, we focus on the events that cause 95% of the victims.

```
ninetyfivePercentCount <- sum(healthSummary$ALL) * 0.95
rows <- 1
repeat {
   count <- sum(healthSummary$ALL[1:rows])
```

```
   if (count >= ninetyfivePercentCount) {
     break
   } else {
     rows <- rows + 1
   }
}
```

When we look at the top events of this summary, we see that tornado's are dominating, with resepect to the number of caused victims. Clearly, reducing victims in future tornado's should be top priority. To get a clear view of the remaining causes, we leave out the tornado's event for the plot.

```
healthPlotData <- healthSummary[2:rows, ]
healthPlotData <- melt(healthPlotData, id=c("EVTYPE"))
names(healthPlotData) <- c("EventType", "Category", "Count")
```

Economic impact

We take a similar approach as with the health impact. First we select the events for which an amount was recorderd for either property damage or crop damage.

```
stormDataEconomic <- stormData[stormData$PROPDMG > 0 | stormData$CROPDMG > 0, ]
```

Some additional cleaning is required. For each amount type there are two columns, a number and an exponent. To make the amounts comparable we unify the amounts by calculating it as an absolute number.

```
stormDataEconomic$PROPDMGEXP <- toupper(stormDataEconomic$PROPDMGEXP)
stormDataEconomic$CROPDMGEXP <- toupper(stormDataEconomic$CROPDMGEXP)

unifyDamage <- function(amount, exp) {
  if (!is.na(suppressWarnings(as.numeric(exp)))) {
    amount <- amount * 10^as.numeric(exp)
  } else {
    if (exp == "") { } # do nothing, keep the amount as is.
    else if (exp == "K") { amount <- amount * 1000 }
    else if (exp == "M") { amount <- amount * 1000000 }
    else if (exp == "B") { amount <- amount * 1000000000 }
    else { amount <- NA } # +, -, or H, I don't know what to with these so we ignore them
  }

  amount
}

stormDataEconomic$PROPDMG.clean <- mapply(unifyDamage,
                                          stormDataEconomic$PROPDMG,
                                          stormDataEconomic$PROPDMGEXP,
                                          SIMPLIFY = TRUE)

stormDataEconomic$CROPDMG.clean <- mapply(unifyDamage,
                                          stormDataEconomic$CROPDMG,
                                          stormDataEconomic$CROPDMGEXP,
                                          SIMPLIFY = TRUE)
```

We have cleaned up the values of property damage, let's do some verification.

```
damagerows <- nrow(stormDataEconomic)
damagerowsna <- nrow(stormDataEconomic[is.na(stormDataEconomic$PROPDMG.clean) |
                                       is.na(stormDataEconomic$CROPDMG.clean),])
```

NOTE: Further investigation is required to see if these 41 observations represent significant damage. It was not possible, due to time constraints to look into this. However, this is 19 rows out of 245031, which is a really small proportion, so I think it is safe to ignore them for now.

Next we create the summary data for plotting, using the same reasoning as for the health impact. We take the events that make up for 95% of the damage.

```
stormDataEconomic <- stormDataEconomic[!is.na(stormDataEconomic$PROPDMG.clean) &
                                       !is.na(stormDataEconomic$CROPDMG.clean),]

economicSummary <- stormDataEconomic %>%
  group_by(EVTYPE) %>%
  select(EVTYPE, PROPDMG.clean, CROPDMG.clean) %>%
  summarise(
    PROPDMG = sum(PROPDMG.clean),
    CROPDMG = sum(CROPDMG.clean),
    ALL = PROPDMG + CROPDMG
  ) %>%
  arrange(desc(ALL))

# Let's see which events cause 95% of the total damage
ninetyfivePercentCount <- sum(economicSummary$ALL) * 0.95
rows <- 1
repeat {
  count <- sum(economicSummary$ALL[1:rows])
  if (count >= ninetyfivePercentCount) {
    break
  } else {
    rows <- rows + 1
  }
}

economicPlotData <- economicSummary[1:rows, ]
economicPlotData <- melt(economicPlotData, id=c("EVTYPE"))
names(economicPlotData) <- c("EventType", "Category", "Amount")
# Divede by A bilion to get the amount in bilions for better readability in the
# plot.
economicPlotData$Amount <- economicPlotData$Amount / 1000000000
```

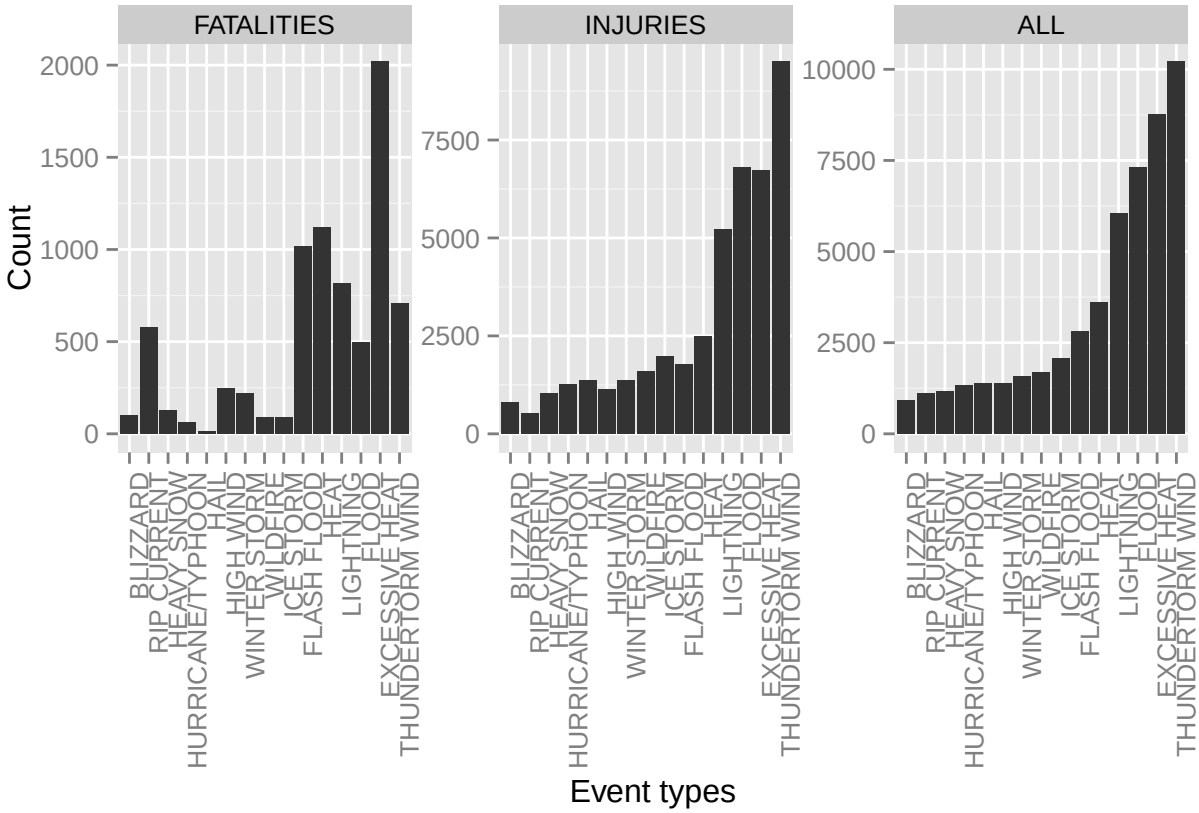
Results

In this section we present some plots, summarizing the most relevant information with respect to impact of events on health and the economy.

Health impact

As, mentioned in the Data processing section, tornado's are causing most of the injuries and fatalities. For further analysis of the remaining events, which cause 95% of the victims, we create a panel plots which shows a breakdown of these events. The plot contains a breakdown for fatalities, injuries and all. The latter, being the sum of both fatality and injury counts.

```
g <- ggplot(healthPlotData, aes(x=reorder(EventType, Count), y=Count)) +
  geom_bar(stat="identity") +
  facet_wrap(~ Category, scales="free_y") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Event types") +
  ylab("Count")
g
```



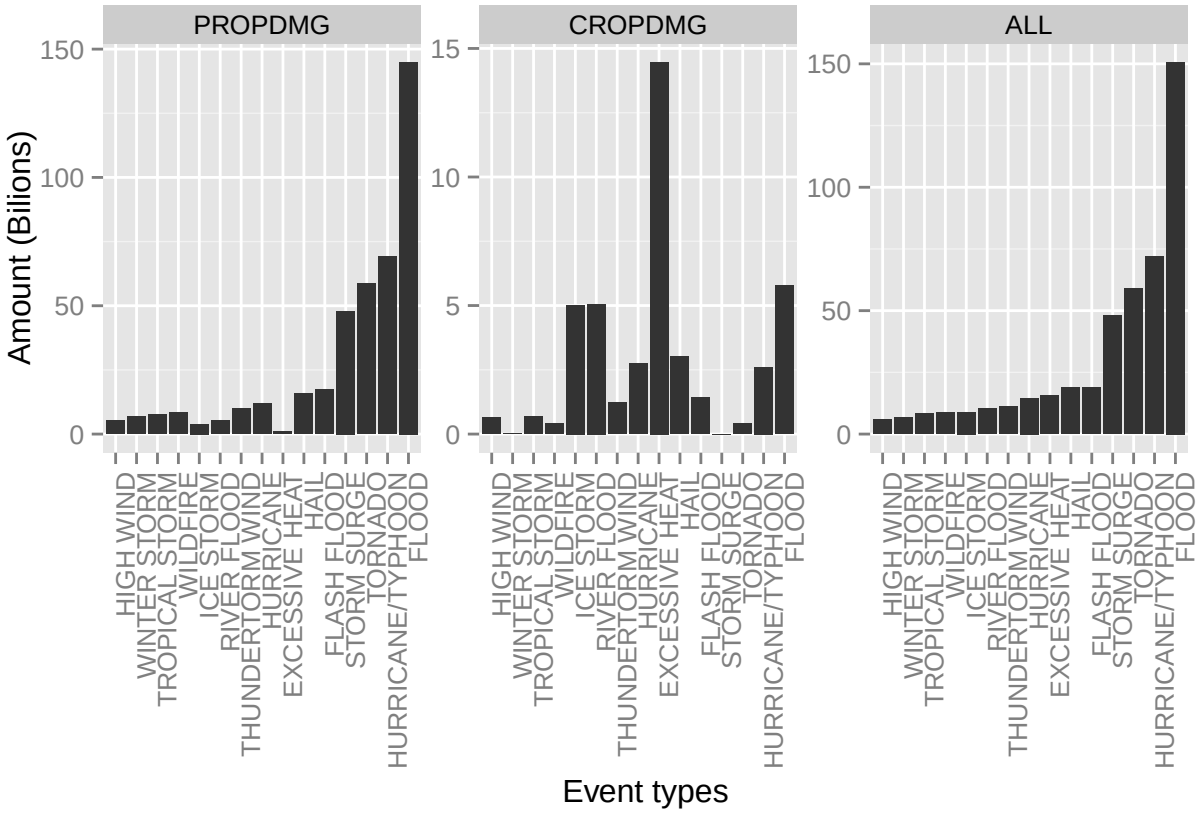
From the plots we learn that:

1. The three top events, causing injuries and fatilites are: thunderstorms, excessive heat, and flood.
2. Those three are also responsible for the most injuries.
3. While heat and flashsfloods are next to thunderstorms, the events causing most causalities.

Economic impact

Like for the health data, we construct a plot showing damage to property, crops and totals, for 95% of the caused damage.

```
g <- ggplot(economicPlotData, aes(x=reorder(EventType, Amount), y=Amount)) +
  geom_bar(stat="identity") +
  facet_wrap(~ Category, scales="free_y") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Event types") +
  ylab("Amount (Bilions)")
g
```



From the plots we learn that:

1. The three top events, causing property and crop damage are: flood, hurricanes, and tornados.
2. Those three are also responsible for the most property damage.
3. Heat, icestorms and flooding are causing most damage on crops.

If time would have permitted, I would have added a time series plot to show when these events occurred over time. Time doesn't permit though, so no such a plot here.