Analysis on the Consequences of weather events in U.S.

Steven Zhang

Dec 1, 2016

1 - 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 results to the extent possible is a key concern.

This analysis 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.

During this analysis, I will focus on these questions:

- Across the United States, which types of events are most harmful with respect to population health?
- Across the United States, which types of events have the greatest economic consequences?

2 - Data Preprocessing

2.1 - Before Loading

The data is downloaded and extracted locally. I have tried to load the data directly but failed. Seems that there are a lot of mistakes in the data, including inline eof, multiple quotation marks etc. To solve this problem, I have used the MS Excel to open the file, and performed following actions:

- Delete the description column, which is long string type and leads to lots of errors.
- Delete all the mismatch rows, which are not properly decoded due to some unknown reason.

After these steps, I got a 902141 * 36 CSV file.

2.2 - Data Loading

```
df <- read.csv("repdata%2Fdata%2FStormData1.csv")
names(df)
```

```
##  [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] "REFNUM"
```

2.3 - Features & Measures Choosing

To answer the questions, the columns in need are listed below:

- EVTYPE: Event Type
- BNG_DATE: Date Time
- FATALITIES: Number of Fatalities
- INJURIES: Number of Injuries
- PROPDMG: Property Damage
- PROPDMGEXP: Units for Property Damage (magnitudes - K,B,M)
- CROPDMG: Crop Damage
- CROPDMGEXP: Units for Crop Damage (magnitudes - K,BM,B)

```
df <- select(df, c(EVTYPE, BGN_DATE, FATALITIES, INJURIES, PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP))
```

We need to notice that property damage and crop damage are with multiple magnitudes, we need to handle it firstly. Attention, the symbols like "+", "-" and "?" will be omitted.

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
df$PROPDMGEXP <- mapvalues(df$PROPDMGEXP, from = c("K", "M", "", "B", "m", "+", "0", "5", "6", "?", "4"
df$PROPDMGEXP <- as.numeric(as.character(df$PROPDMGEXP))
df$PROPDMG <- (df$PROPDMG * df$PROPDMGEXP)
df$CROPDMGEXP <- mapvalues(df$CROPDMGEXP, from = c("", "M", "K", "m", "B", "?", "0", "k", "2"), to = c(
df$CROPDMGEXP <- as.numeric(as.character(df$CROPDMGEXP))
df$CROPDMG <- (df$CROPDMG * df$CROPDMGEXP)
detach("package:plyr", unload=TRUE)
```

```
## Warning: 'plyr' namespace cannot be unloaded:
##   namespace 'plyr' is imported by 'ggplot2', 'scales' so cannot be unloaded
```

```
df <- select(df, c(-CROPDMGEXP, -PROPDMGEXP))
```

So far, The data is ready.

3 - Exploratory Research

There are too many types to plot. I have to aggregate data firstly.

```
df.by.event <- df %>% group_by(EVTYPE) %>% summarise(FATALITIES = sum(FATALITIES), INJURIES = sum(INJUR
df.by.event.fatal <- arrange(df.by.event, desc(FATALITIES))
df.by.event.injury <- arrange(df.by.event, desc(INJURIES))
```

```
df.by.event.property <- arrange(df.by.event, desc(PROPDMG))
df.by.event.crop <- arrange(df.by.event, desc(CROPDMG))
```

3.1 - Question A

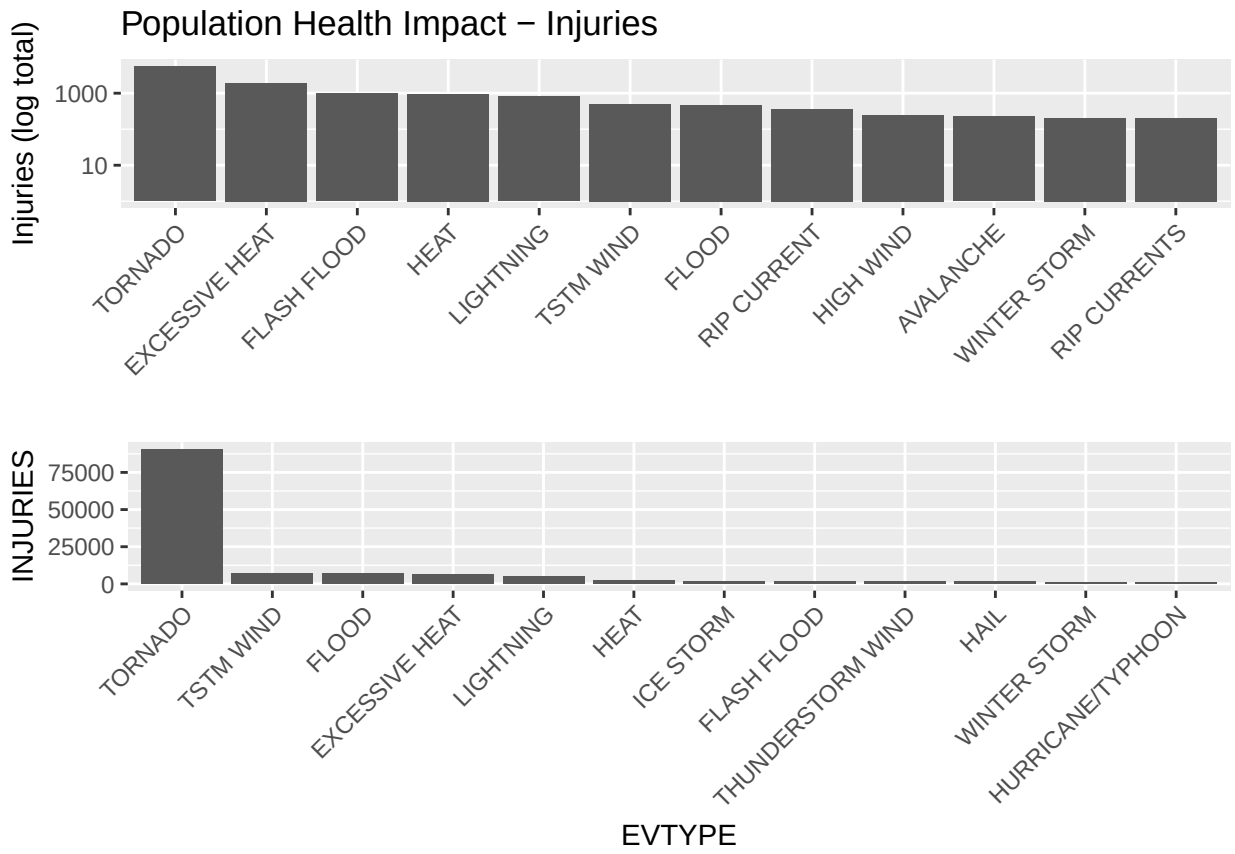
Across the United States, which types of events are most harmful with respect to population health?

```
temp <- df.by.event.fatal[1:12,]
temp$EVTYPE <- reorder(temp$EVTYPE, -temp$FATALITIES)
f <- ggplot(temp, aes(EVTYPE, FATALITIES))
p1 <- f + geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
p1 <- p1 + xlab("")+ ylab("Fatalities (log total)") + ggtitle("Population Health Impact - Fatalities")+
```

```
temp <- df.by.event.injury[1:12,]
temp$EVTYPE <- reorder(temp$EVTYPE, -temp$INJURIES)
f <- ggplot(temp, aes(EVTYPE, INJURIES))
p2 <- f + geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
p1 <- p1 + xlab("") + ylab("Injuries (log total)") + ggtitle("Population Health Impact - Injuries")+ sca
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```

```
grid.arrange(p1, p2, ncol = 1, nrow = 2)
```



3.2 - Question B

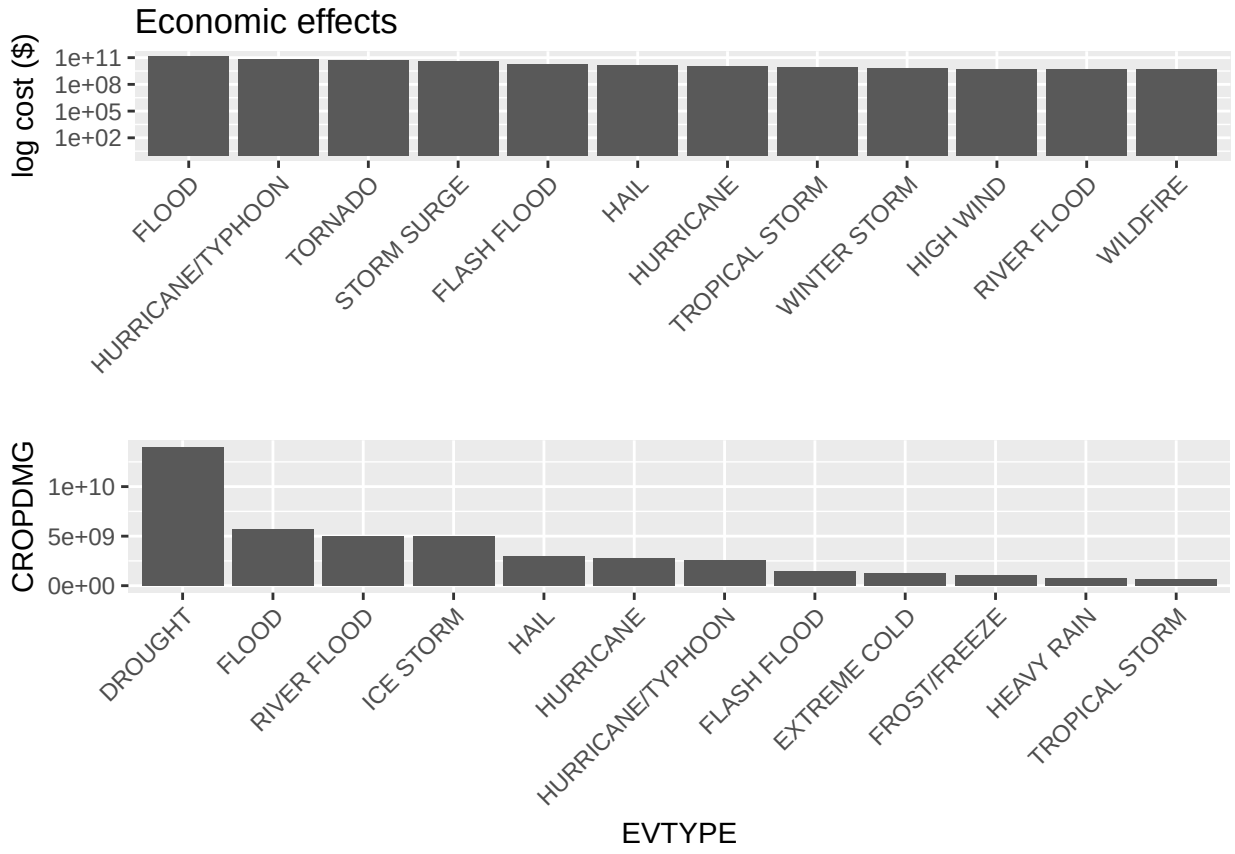
Across the United States, which types of events have the greatest economic consequences?

```
temp <- df.by.event.property[1:12,]
temp$EVTYPE <- reorder(temp$EVTYPE, -temp$PROPDMG)
f <- ggplot(temp, aes(EVTYPE, PROPDMG))
p1 <- f + geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
p1 <- p1 + xlab("")  + ylab("log cost ($)") + ggtitle("Economic effects")+ scale_y_log10()
```

```
temp <- df.by.event.crop[1:12,]
temp$EVTYPE <- reorder(temp$EVTYPE, -temp$CROPDMG)
f <- ggplot(temp, aes(EVTYPE, CROPDMG))
p2 <- f + geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
p1 <- p1 + xlab("") + ylab("log cost ($)") + ggtitle("Economic effects")+ scale_y_log10()
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```

```
grid.arrange(p1, p2, ncol = 1, nrow = 2)
```



4 - Results

As demonstrated by the charts, **TORNADO** causes the most of deaths and injures. Specifically in fatalities, **EXCESSIVE HEAT**, **FLASH FLOOD** and **HEAT** come behind. Specifically in injuries, we see **TSTM WIND**, **FLOOD** and **EXCESSIVE HEAT**.

About property, **FLOOD** causes the most property damage, followed by **HURRICANE**, **TORNADO** and **STORM SURGE**. **DROUGHT** causes most crop damage, follwed by **FLOOD**, **RIVER FLOOD** and **ICE STORM**.

Based on evidences demonstrated above, **TORNADO** and **FLOOD** are the most serious weather events. The government and other NGOs need to pay attention on them mostly. We need to prepare to face them, and teach people about what to do during these events.

5 - Summary

5.1 - Reference

- *National Weather Service Storm Data Documentation*
- *National Climatic Data Center Storm Events FAQ*

5.2 - Software & Hardware Env.

```
sessionInfo()
```

```
## R version 3.2.5 (2016-04-14)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 10586)
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.936
## [2] LC_CTYPE=Chinese (Simplified)_China.936
## [3] LC_MONETARY=Chinese (Simplified)_China.936
## [4] LC_NUMERIC=C
## [5] LC_TIME=Chinese (Simplified)_China.936
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dplyr_0.5.0   ggplot2_2.2.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.7      knitr_1.14       magrittr_1.5     munsell_0.4.3
##  [5] colorspace_1.2-6 R6_2.2.0         stringr_1.1.0    plyr_1.8.4
##  [9] tools_3.2.5      grid_3.2.5       gtable_0.2.0     DBI_0.5-1
## [13] htmltools_0.3.5  yaml_2.1.13      lazyeval_0.2.0   digest_0.6.10
## [17] assertthat_0.1   tibble_1.2       formatR_1.4      codetools_0.2-15
## [21] evaluate_0.9     rmarkdown_1.0    labeling_0.3     stringi_1.1.2
## [25] scales_0.4.1
```