Analyzing Storm Data For Human and Economic Cost

JDV

June 5, 2018

Synopsis

The purpose of this report is to analyze and document data from weather events and determine which of them have the biggest economic and health-related impact. Data was obtained from the NOAA database. Floods were found to be the more disastrous in terms of economic damage, while tornados and head related events were the most damaging for human well-being. To arrive at this conclusion data was analyzed and cleaned to arrive at more normalized fields for the event type, property damage, and human victims.

Data Processing

```
# Read raw data and prepare to preprocess
data <- read.csv("repdata_data_StormData.csv.bz2")

stormData <- setDT(data)
```

Once we've loaded the data we realize we have 902297 rows and 37 columns. We can get a glimpse of what we're dealing with using `str()` if we so desire.

We are particularly interested in columns regarding health impact, economic impact and the type of event we're dealing with. So we'll take a closer a look at those that seem interesting at a glance:

```
kable(as.matrix(summary(stormData[, .(EVTYPE, FATALITIES, INJURIES, PROPDMG, CROPDMG)])))
```

EVTYPE	FATALITIES	INJURIES	PROPDMG	CROPDMG
HAIL :288661	Min. : 0.0000	Min. : 0.0000	Min. : 0.00	Min. : 0.000
TSTM WIND :219940	1st Qu.: 0.0000	1st Qu.: 0.0000	1st Qu.: 0.00	1st Qu.: 0.000
THUNDERSTORM WIND: 82563	Median : 0.0000	Median : 0.0000	Median : 0.00	Median : 0.000
TORNADO : 60652	Mean : 0.0168	Mean : 0.1557	Mean : 12.06	Mean : 1.527
FLASH FLOOD : 54277	3rd Qu.: 0.0000	3rd Qu.: 0.0000	3rd Qu.: 0.50	3rd Qu.: 0.000
FLOOD : 25326	Max. :583.0000	Max. :1700.0000	Max. :5000.00	Max. :990.000
(Other) :170878	NA	NA	NA	NA

We can see that damage values are on different scales depending on the event. We're gonna have to convert these so that a more straight-forward comparison can be made. The exponentials can be found in the `*EXP` fields. We can also see that there's no case consistency, so we'll deal with that as well.

```
# Ignore rows that only seem to have corrupt data for the exponent
stormData <- stormData[!(PROPDMGEXP %in% c("-", "?", "+"))]
stormData <- stormData[!(CROPDMGEXP %in% c("-", "?", "+"))]

# We get everything to be upper case
stormData$PROPDMGEXP <- factor(toupper(stormData$PROPDMGEXP))
stormData$CROPDMGEXP <- factor(toupper(stormData$CROPDMGEXP))
stormData$EVTYPE <- factor(toupper(stormData$EVTYPE))
```

```r
# We replace empty exponents with exponent 0 (class 2)
stormData[PROPDMGEXP == "", PROPDMGEXP := as.integer(2)]
stormData[, PROPDMGEXP := factor(PROPDMGEXP)]

# We create an exponent map and get the actual numbers
expMap <- data.frame(c(levels(stormData$PROPDMGEXP)), stringsAsFactors = F)
names(expMap) <- c("val")
row.names(expMap) <- levels(stormData$PROPDMGEXP)
expMap["H", 1] <- 2
expMap["K", 1] <- 3
expMap["M", 1] <- 6
expMap["B", 1] <- 9

stormData[, PROPDMG := PROPDMG * 10^as.integer(expMap[PROPDMGEXP, 1])]
stormData[, CROPDMG := PROPDMG * 10^as.integer(expMap[CROPDMGEXP, 1])]

head(stormData[order(-PROPDMG), PROPDMG])
```

```
## [1] 1.150e+11 3.130e+10 1.693e+10 1.126e+10 1.000e+10 7.350e+09
```

Numbers are looking better now; they're all on the same scale.

Now we have to deal with the thousands of possible `EVTYPE` values, which complicates things. I'll just do a lazy cleanup using `grep` and replacing some values, hopefully ending up with something reasonable. Ideally you'd spend more time on this.

```r
stormData[, EVTYPE := as.character(EVTYPE)]
stormData[grepl("HAIL", EVTYPE), EVTYPE := "HAIL"]
stormData[grepl("ICE", EVTYPE), EVTYPE := "ICE"]
stormData[grepl("AVALANC", EVTYPE), EVTYPE := "AVALANCHE"]
stormData[grepl("FOG|VOG", EVTYPE), EVTYPE := "FOG"]
stormData[grepl("FLOOD", EVTYPE), EVTYPE := "FLOOD"]
stormData[grepl("WIND|WND", EVTYPE), EVTYPE := "WIND"]
stormData[grepl("SNOW", EVTYPE), EVTYPE := "SNOW"]
stormData[grepl("WATER|WET|RAIN|PRECIPITATION|SHOWER", EVTYPE), EVTYPE := "WATER"]
stormData[grepl("DUST|LANDSPOUT", EVTYPE), EVTYPE := "DUST"]
stormData[grepl("BLIZZARD", EVTYPE), EVTYPE := "BLIZZARD"]
stormData[grepl("LIGHTNING", EVTYPE), EVTYPE := "LIGHTNING"]
stormData[grepl("TORNADO|TORNDAO", EVTYPE), EVTYPE := "TORNADO"]
stormData[grepl("FUNNEL", EVTYPE), EVTYPE := "FUNNEL CLOUD"]
stormData[grepl("COASTAL", EVTYPE), EVTYPE := "COASTAL"]
stormData[grepl("VOLCAN", EVTYPE), EVTYPE := "VOLCANO"]
stormData[grepl("HURRICANE", EVTYPE), EVTYPE := "HURRICANE"]
stormData[grepl("RIP", EVTYPE), EVTYPE := "RIP CURRENTS"]
stormData[grepl("LIGHTNING", EVTYPE), EVTYPE := "LIGHTNING"]
stormData[grepl("WINTER|WINTR", EVTYPE), EVTYPE := "WINTER"]
stormData[grepl("COLD|COOL|FREEZ|FROST|LOW", EVTYPE), EVTYPE := "COLD"]
stormData[grepl("HOT|HEAT|WARM|DRY|DRIEST|HIGH", EVTYPE), EVTYPE := "HOT"]
stormData[grepl("FIRE", EVTYPE), EVTYPE := "FIRE"]
stormData[grepl("URBAN", EVTYPE), EVTYPE := "URBAN"]
stormData[grepl("THUNDERSTORM|TSTM", EVTYPE), EVTYPE := "THUNDERSTORM"]
stormData[grepl("TROPICAL", EVTYPE), EVTYPE := "TROPICAL"]
stormData[grepl("MUD|LAND", EVTYPE), EVTYPE := "MUD"]
stormData <- stormData[!(grepl("SUMMARY|COUNTY|\\?|MONTH|EXCESSIVE|NONE|WEATHER|TEMPERATU", EVTYPE))]
stormData[, EVTYPE := factor(EVTYPE)]
```

Now the dirty work is done and our `EVTYPE` variable has been normalized.

Results

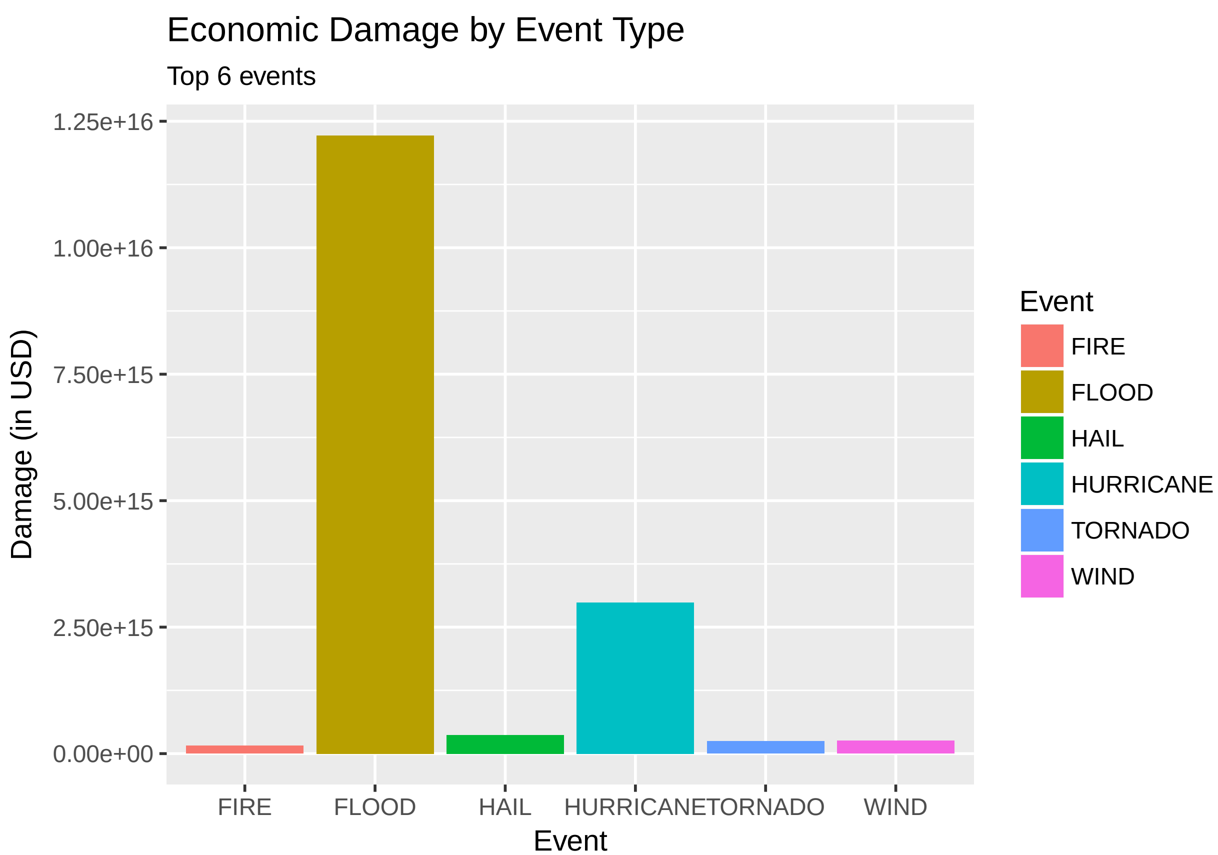
In order to get a better sense of the results, we shall create a new variable for the general economic damage generated by the storm. Let's take a look at the resulting values:

```
stormData[, DMG := PROPDMG + CROPDMG]
kable(as.matrix(summary(stormData$DMG)))
```

Min.	0.000000e+00
1st Qu.	0.000000e+00
Median	0.000000e+00
Mean	1.818369e+10
3rd Qu.	2.000000e+03
Max.	1.150012e+16

Good news is we don't have missing data so we can just go ahead and plot this for each `EVTYPE` and take a look at what we get.

```
events <- 6
ggplot(head(stormData[,sum(DMG),by=EVTYPE][order(-V1)],n=events),aes(x=EVTYPE,y=V1)) +
  geom_bar(aes(fill=EVTYPE), stat="identity") +
  labs(title="Economic Damage by Event Type",x="Event",y="Damage (in USD)", subtitle=paste("Top",events
```



From the plot above we can tell the flooding causes far more damage than other types of events. The margin is very wide.

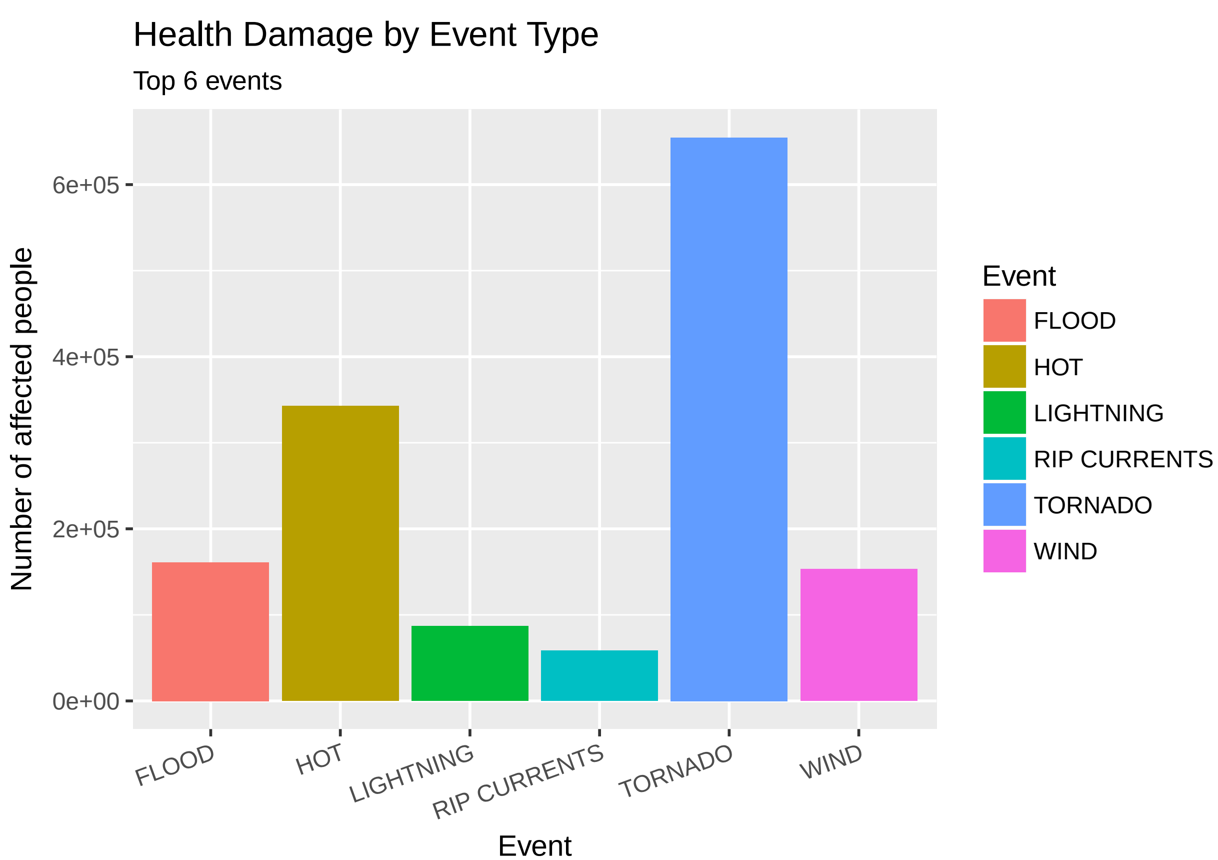
Now we tackle the well-being issue. For this we will estimate a number of affected people. Fatalities will be multiplied by a weight which we will define and which can be altered for further analyses. For the purposes of this project we will just select a weight of **'r weight**.

```
stormData[, HEALTHDMG := weight * FATALITIES + INJURIES]
kable(as.matrix(summary(stormData$HEALTHDMG)))
```

Min.	0.000000
1st Qu.	0.000000
Median	0.000000
Mean	1.834363
3rd Qu.	0.000000
Max.	58300.000000

Now that we have our one field to gauge how many people were affected by the event, we can now get a good estimate of what events are the most disastrous for human life/health.

```
ggplot(head(stormData[, sum(HEALTHDMG), by=EVTYPE][order(-V1)], n=events),
       aes(x = EVTYPE, y = V1)) +
  geom_bar(aes(fill=EVTYPE), stat="identity") +
  labs(title="Health Damage by Event Type",x="Event",
       y="Number of affected people", subtitle=paste("Top",events,"events"), fill="Event") +
  theme(axis.text.x = element_text(angle = 20, hjust = 1))
```



Tornados hurt far more people than any other event. Heat related events and floods follow torandos.