Taken by Storms

An Analysis of Damage Reported in the National Weather Service's Storm Data

Coursera Student

Monday, May 20, 2015

Synopsis

After a major storm, an unusual bout of weather, or a significant intense non-storm weather event, the National Weather Service (NWS) and other people and agencies gather information about the storm, including where it happened, when it happened, meteorological details, and the damage it caused, and publish that in the National Oceanic and Atmospheric Administration's (NOAA) *Storm Data* a copy of which can be found here. In this analysis, we take a quick look at Storm Data to see which types of storms are the most harmful to public health and which cause the most economic damage. To do this, we group the data by type of event, count the injuries and fatalities reported to determine the most injurious to health and combine property and crop damage estimated to determine which type of event caused the greatest economic harm. Based on this methodology, we find that according to this data: * tornados have caused the most injuries and fatalities * floods have caused the most economic damage

Data Processing

We start by reading in the libraries we'll need to do the analysis:

```
library(plyr)
library(dplyr)
library(reshape)
library(ggplot2)
library(scales)
library(lubridate)
```

We then read in the file. We have already downloaded and unzipped it into our working directory.

```
allStormData <- read.csv("repdata-data-StormData.csv", stringsAsFactors=FALSE)
```

For this analysis, we care about the column that shows the event type, the columns that show deaths and injuries, and the columns that show property and crop damage. So, we're going to only focus on those columns and the rows that have a value in at least one of those columns.

```
damageData <- allStormData %>%
  select(EVTYPE,FATALITIES,INJURIES,PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP) %>%
  filter(FATALITIES+INJURIES+PROPDMG+CROPDMG>0)

dim(damageData)
```

```
## [1] 254633      7
```

At this point, we want to note that this is not the full extent of damage caused by storms, it is just the damage that is both reported and easily accessed through this table. From NOAA and NWS's documentation about this data set, we learn that might not include indirect fatalities or injuries (that information will be embedded in text in the remarks) nor will it include damage to people or property that occurs post event. Those numbers are also in text in the remarks. The data is further skewed because it doesn't include all events, just severe and unusual ones. In other words, a light snowfall in Georgia that causes injuries and crop damage will be included in the data, a similarly light snowfall in Vermont might not be included. Further information on data accuracy can be found here.

With that caveat to its accuracy, we take the data that we have, and get a closer look at the event type in the data frame.

```
length(unique(damageData$EVTYPE))
```

```
## [1] 488
```

In our dataset, we have 488 EVTYPES. The issue with that, is that there should be at most 48 event types (Reference:, Table 1). Among the data in our EVTYPE column are a number of mispellings (e.g, "AVALANCE" instead of "AVALANCHE"), mislabelings ("Coastal Flooding" rather than "Coastal Flood"), inconsistent capitalizations (e.g., "Dust Devil" and "DUST DEVIL"), specific rather than generic (e.g., "Hurricane Emily" rather than "Hurricane"), and some badly labeled items (e.g.,"?"). To remedy this, we create a CSV that will translate between the values in our set data set and the allowed values (plus "Other" for EVTYPES that defy classification) and will allow us to continue with our analysis. A permanent copy of that csv can be found here. To use it, we store it in the working directory. We'll then read it in and merge it with our data frame.

```
evtypeConversion <- read.csv("EvtypeConversion.csv",stringsAsFactors=FALSE)
damageData <- merge(x=damageData, y=evtypeConversion, by.x="EVTYPE", by.y="ActualEVTYPE")
length(unique(damageData$ValidValue))
```

```
## [1] 47
```

We can see that there are an expected number of types now.

Next, we change property & crop damage to numbers rather than a coefficient (in the PROPDMG and CROPDMG columns) and an exponent (in the PROPDMGEXP and CROPDMGEXP columns).

If the significands are numbers, we'll assume that they mean 10 to the power of that number. From the documentation, we know that "B" is billions, "M" is millions, "K" is thousands, and "H" is hundreds. We will treat any other value is going to be treated as 1. We'll then merge those into the data frame and multiplying them across, once for Property Damage and once for Crop Damage

```
scientificNotation <-
  data.frame(original=c(0,1,2,3,4,5,6,7,8,"h","H","k","K","m","M","B"),
             multiplier=c(1,10,100,1000,10000,100000,1000000,10000000,100000000,100,100,1000,1000,100000

## Property Damage

damageData <- merge(x=damageData,
                    y=scientificNotation,
                    by.x="PROPDMGEXP",
                    by.y="original",
                    all.x=TRUE)
damageData <- damageData %>%
```

```
  mutate(multiplier=as.numeric(ifelse(is.na(multiplier),1,multiplier)),
         PropertyDamage=multiplier*PROPDMG) %>%
  select(-multiplier)

## Crop Damage

damageData <- merge(x=damageData,
                    y=scientificNotation,
                    by.x="CROPDMGEXP",
                    by.y="original",
                    all.x=TRUE)
damageData <- damageData %>%
  mutate(multiplier=as.numeric(ifelse(is.na(multiplier),1,multiplier)),
         CropDamage=multiplier*CROPDMG) %>%
  select(-multiplier)
```

Next, we create two columns, one that combines the health damage (FATALITIES and INJURIES) and one that combines the economic damage (PropertyDamage and CropDamage)

```
damageData <- mutate(damageData,
                     allHealthDamage=FATALITIES+INJURIES,
                     allEconomicDamage=PropertyDamage+CropDamage)
```

And now we create a summary dataframe that adds all of the property and public health damage by event type.

```
damageSummary <- data.frame(summarise(group_by(damageData,ValidValue),
                                      Fatalities = sum(FATALITIES),
                                      Injuries = sum(INJURIES),
                                      HealthDamage = sum(allHealthDamage),
                                      PropertyDamage = sum(PropertyDamage),
                                      CropDamage = sum(CropDamage),
                                      EconomicLoss= sum(allEconomicDamage)))

colnames(damageSummary)[1] <- c("EventType")
sample_n(damageSummary,10)
```

```
##             EventType Fatalities Injuries HealthDamage PropertyDamage
## 1           Avalanche        225      170          395      3.722e+06
## 47     Winter Weather         85      766          851      6.594e+07
## 2            Blizzard        102      819          921      6.705e+08
## 24 Hurricane-Typhoon        135     1333         1468      8.536e+10
## 35              Sleet          4       26           30      1.950e+06
## 23          High Wind        293     1472         1765      6.005e+09
## 6           Dense Fog         81     1077         1158      2.283e+07
## 14       Freezing Fog          1        0            1      0.000e+00
## 37        Strong Wind        153      439          592      1.940e+08
## 28        Marine Hail          0        0            0      4.000e+03
##    CropDamage EconomicLoss
## 1   0.000e+00    3.722e+06
## 47  1.502e+07    8.096e+07
## 2   1.121e+08    7.826e+08
```

```
## 24  5.516e+09    9.087e+10
## 35  0.000e+00    1.950e+06
## 23  6.863e+08    6.691e+09
## 6   0.000e+00    2.283e+07
## 14  0.000e+00    0.000e+00
## 37  7.622e+07    2.702e+08
## 28  0.000e+00    4.000e+03
```

Looking at the sample, it's clear we now have a very easily read table. The table will be stored as a CSV in the working directory. There's also a copy here for easy reference.

```r
write.csv(damageSummary, file="StormDataDamage.csv")
```

We then split it into two. One for economic damage and one for health damage. We'll also largest to smallest amount of total damage for each category.

```r
healthSummary <- damageSummary %>%
  select(EventType,Fatalities,Injuries,HealthDamage) %>%
  filter(HealthDamage>0) %>%
  arrange(desc(HealthDamage))

economicSummary <- damageSummary %>%
  select(EventType,PropertyDamage,CropDamage,EconomicLoss) %>%
  filter(EconomicLoss>0) %>%
  arrange(desc(EconomicLoss))

sample_n(economicSummary,5)
```

```
##           EventType PropertyDamage CropDamage EconomicLoss
## 12     Winter Storm      6.749e+09   32444000    6.781e+09
## 14       Heavy Rain      3.238e+09  938505800    4.176e+09
## 13        High Wind      6.005e+09  686321900    6.691e+09
## 35        Avalanche      3.722e+06          0    3.722e+06
## 4  Storm Surge/Tide      4.796e+10     855000    4.797e+10
```

```r
sample_n(healthSummary,5)
```

```
##             EventType Fatalities Injuries HealthDamage
## 36      Coastal Flood          9        7           16
## 34 Marine Strong Wind         14       22           36
## 31       Frost/Freeze          9       59           68
## 19     Winter Weather         85      766          851
## 28        Debris Flow         49       58          107
```

Results

We can now take a look at our data.

First harm to public health. As a reminder, the table is sorted from most to least damaging, so the top of the table has the most injuries and fatalities combined.

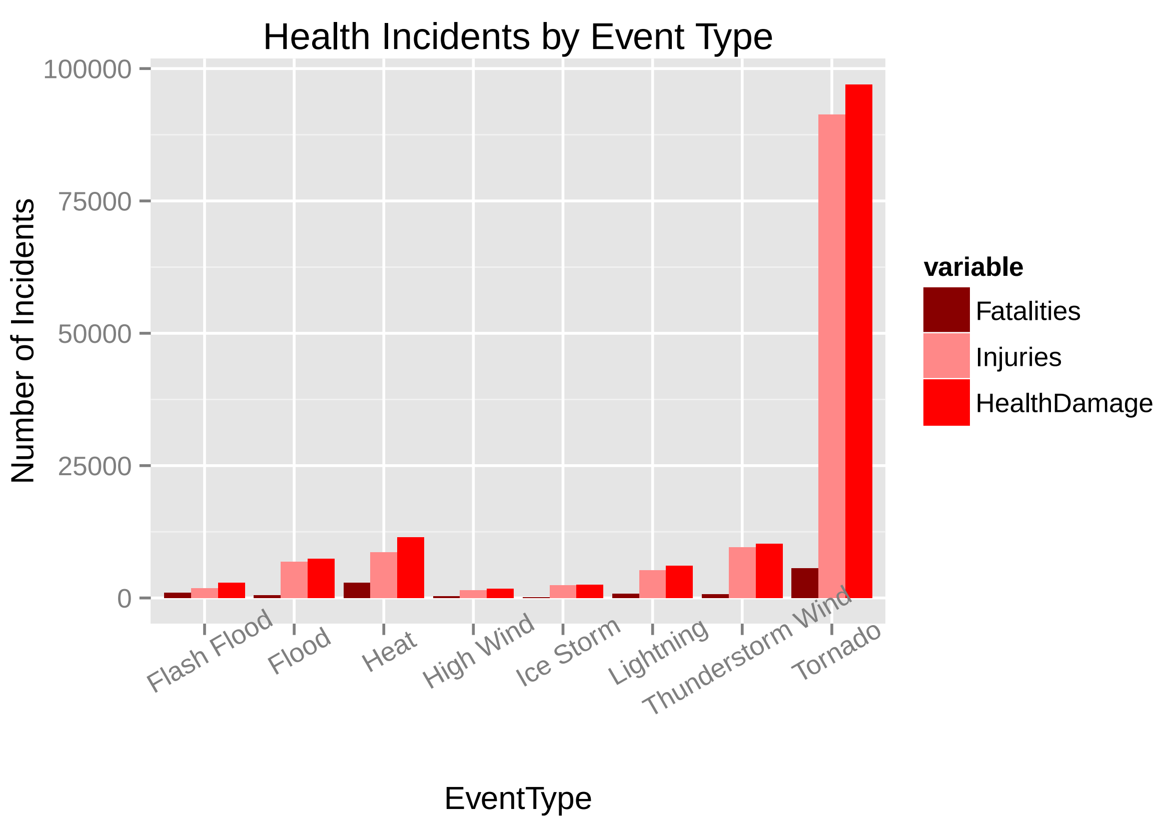
```r
head(healthSummary,8)
```

```
##            EventType Fatalities Injuries HealthDamage
## 1            Tornado       5658    91367        97025
## 2               Heat       2840     8627        11467
## 3 Thunderstorm Wind         715     9538        10253
## 4              Flood        514     6874         7388
## 5          Lightning        817     5232         6049
## 6        Flash Flood       1036     1800         2836
## 7          Ice Storm        112     2383         2495
## 8          High Wind        293     1472         1765
```

Tornados have nearly 9 times the total damage of the next most devastating event type, Heat. They also caused twice as many fatalities as Heat did.

We are going to transform the top 8 events into a tidy table and look at it graphically.

```r
topHealthSummary <- melt(head(healthSummary,8),c("EventType"))
ggplot(topHealthSummary, aes(EventType))+
  geom_bar(aes(y=value, fill=variable),stat="identity", position="dodge")+
  theme(axis.text.x=element_text(angle=30))+
  ylab("Number of Incidents")+
  ggtitle("Health Incidents by Event Type")+
  scale_fill_manual(values=c("#880000","#FF8888","#FF0000"))
```



We do the same for economic damage

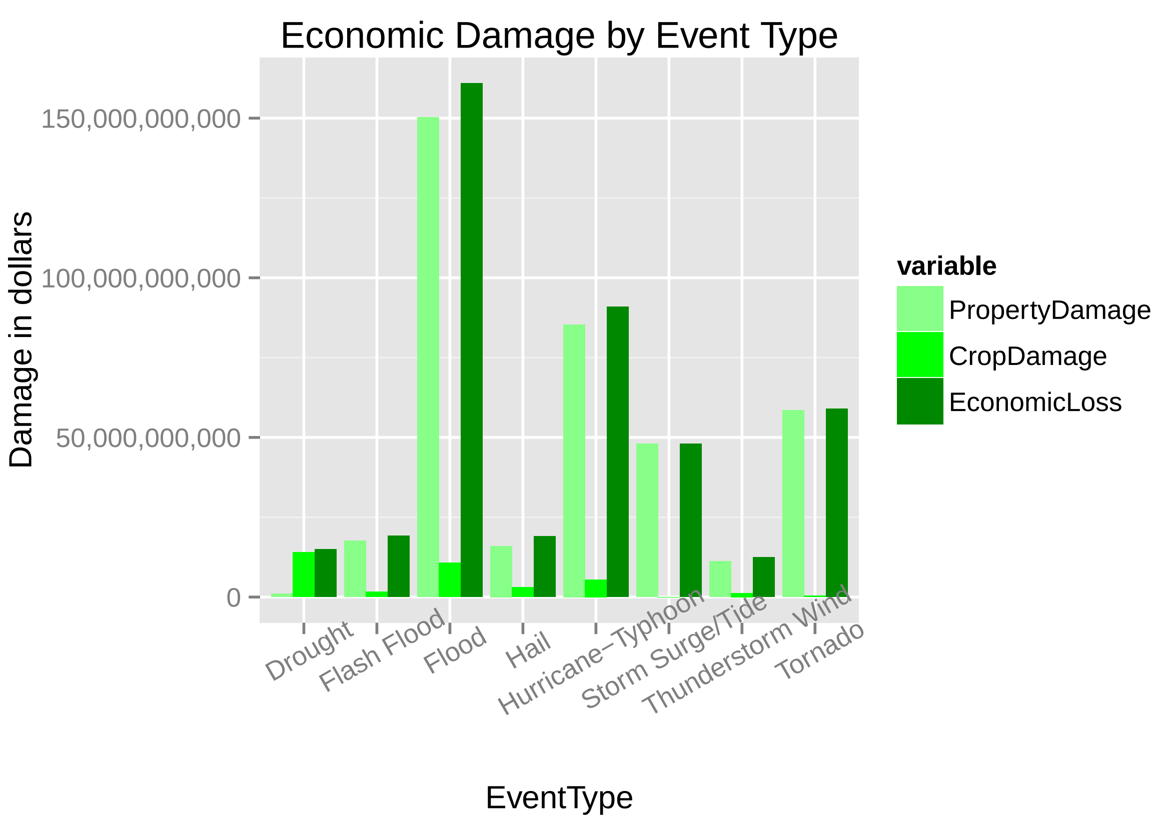
```r
head(economicSummary,8)
```

```
##           EventType PropertyDamage CropDamage EconomicLoss
## 1             Flood      1.502e+11  1.074e+10    1.610e+11
## 2 Hurricane-Typhoon      8.536e+10  5.516e+09    9.087e+10
## 3           Tornado      5.855e+10  4.175e+08    5.897e+10
## 4  Storm Surge/Tide      4.796e+10  8.550e+05    4.797e+10
## 5       Flash Flood      1.759e+10  1.645e+09    1.923e+10
## 6              Hail      1.598e+10  3.047e+09    1.902e+10
## 7           Drought      1.046e+09  1.397e+10    1.502e+10
## 8 Thunderstorm Wind      1.119e+10  1.272e+09    1.246e+10
```

Floods are twice as devastating economically as the next most damaging event type; they cause a full third of the total economic loss reported in Storm Data.

Again, we'll plot the top 8 most damaging event types

```r
topEconomicSummary <- melt(head(economicSummary,8),c("EventType"))
ggplot(topEconomicSummary, aes(EventType))+
  geom_bar(aes(y=value, fill=variable),stat="identity", position="dodge")+
  theme(axis.text.x=element_text(angle=30))+
  ylab("Damage in dollars")+
  scale_y_continuous(labels = comma)+
  ggtitle("Economic Damage by Event Type")+
  scale_fill_manual(values=c("#88FF88","#00FF00","#008800"))
```



It is clear how much more damaging floods are than any other type of event.