National Weather Service Storm Data Analysis - Most Damaging Event Types

Synopsis

The data is analysed by firstly reading it into R as a data frame. To enable analysis of the damage to human health, the dataframe is summarised by Event type to calculate the total damage by each event type. Datapoints with comparitively little damage are then filtered out of the dataset to make the final plots more readable. A metric for the total damage to human health (fatalities + injuries) is also calculated with a weighting such that a fatality is worth twice as much as a injury. The total financial damage is calculated by first expanding the data (i.e. 2.5 and k becomes 2500) before summarising it in a similar way to the human health data. In this case the metric for total damage is simply the sum of the property and crop damage. Both datasets are also melted before plotting to make it easier to use plotting libraries (in this case ggplot2). Also, the datasets are arranged in descending order of total damage to make it easier to visualise the data when it is presented in a tabular format.

Data Processing

Loads the needed libraries

```
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)
```

```
## Use suppressPackageStartupMessages() to eliminate package startup
## messages.
```

```
library(reshape2)
```

Downloads and Reads the dataset into R.

```
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2", "data.csv.bz2
rawData <- read.csv("data.csv.bz2")
rawData <- rawData %>% group_by(EVTYPE)
```

Summarises the relevant variables (Fatalities, INJURIES, and PROPDMG) to make it easier to plot them. Also filters out the data points to remove some of the “noise”" (insignificantly small values) to make the plot more readable.

```
summarisedData <- rawData %>% group_by(EVTYPE) %>% summarise(totalFatalities = sum(FATALITIES), totalIn
humanDamageData <- summarisedData %>% filter(totalFatalities>10, totalInjuries>10)
```

Attempts to estimate the total damage to human health as a function of the totalFatilities and totalInjuries. I chose to weight them such that a fatality is worth twice as much as an injury.

```
humanDamageData <- humanDamageData %>% mutate(totalHumanDamage = 2*totalFatalities + totalInjuries) %>%
  arrange(desc(totalHumanDamage))
```

Calculates the total property and crop damage using the PROPDMGEXP column before filtering datapoints with low values to make the plot more readable

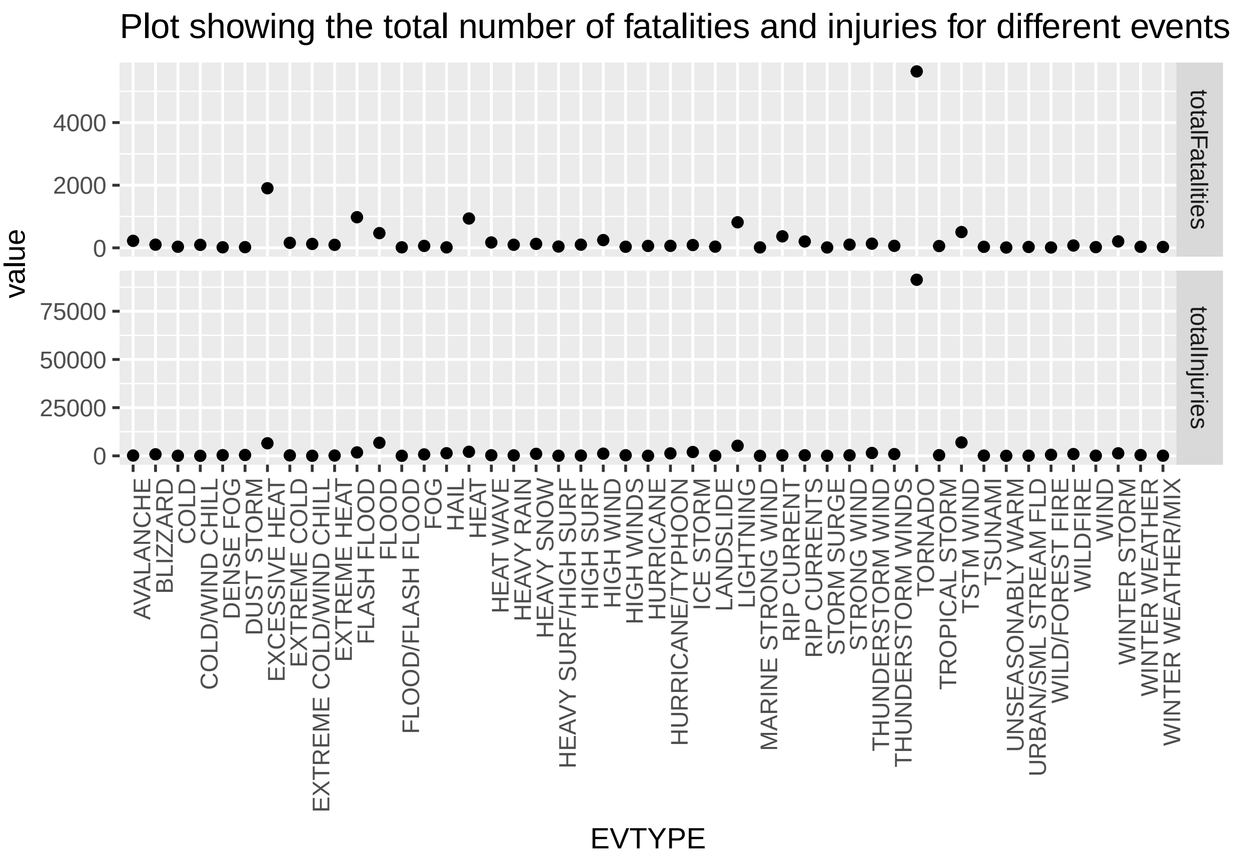
```
propDamageData <- rawData %>% mutate(completePropDamage = ifelse(tolower(PROPDMGEXP)=="k",PROPDMG*1000,
  mutate(completeCropDamage = ifelse(tolower(CROPDMGEXP)=="k",CROPDMG*1000,ifelse(tolower(CROPDMGEXP)==
  select(EVTYPE, completePropDamage, completeCropDamage) %>%
  mutate(totalDamage=completePropDamage+completeCropDamage) %>%
  filter(completePropDamage>1000000, completeCropDamage>1000000)

propDamageData <- propDamageData %>% group_by(EVTYPE) %>%
  summarise(totalPropDamage=sum(completePropDamage), totalCropDamage=sum(completeCropDamage), totalDama
```

Results

Damage to Human Life

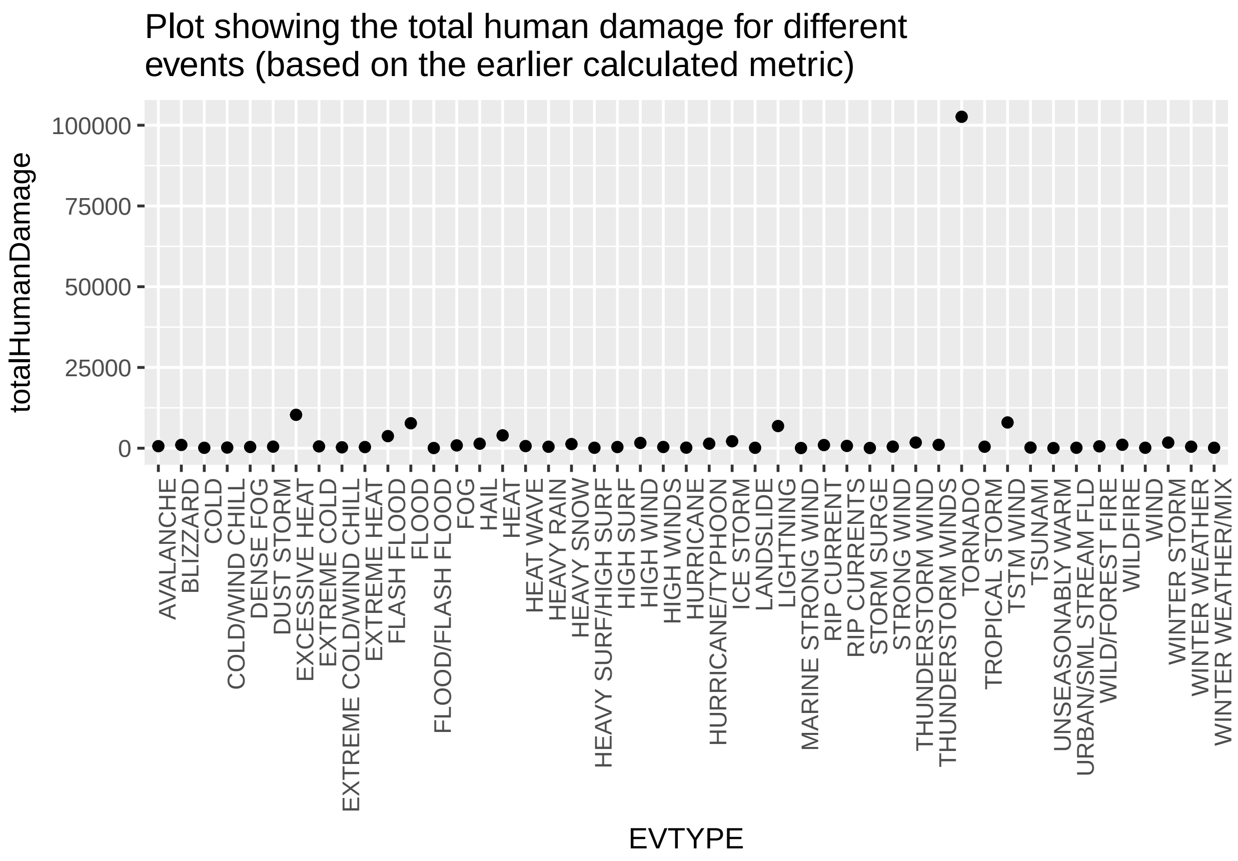
Plots only the fatality and injury data to help visualise any overall trends.

```
## Melts the data to make it a long dataset (which can be plotted in ggplot2)
fatalityAndInjuryData <- melt(humanDamageData %>% select(EVTYPE, totalFatalities, totalInjuries), id="E
g <- ggplot(fatalityAndInjuryData, aes(EVTYPE, value)) +
  facet_grid(variable~., scales="free") +
  geom_point() +
  theme(axis.text.x = element_text(angle=90, hjust=1)) +
  ggtitle("Plot showing the total number of fatalities and injuries for different events")
print(g)
```



Plots the eariler calculated metric for total human health damage. This helps to visualise the data from the 2 earlier plots and any trends between them.

```
g <- ggplot(humanDamageData, aes(EVTYPE, totalHumanDamage)) +
  geom_point() +
  theme(axis.text.x = element_text(angle=90, hjust=1)) +
  ggtitle("Plot showing the total human damage for different \nevents (based on the earlier calculated 
print(g)
```



Displays the 10 events which are most damaging to human health to help inform any decision making (by providing precise information)

```
head(arrange(humanDamageData, desc(totalHumanDamage)), 10)
```

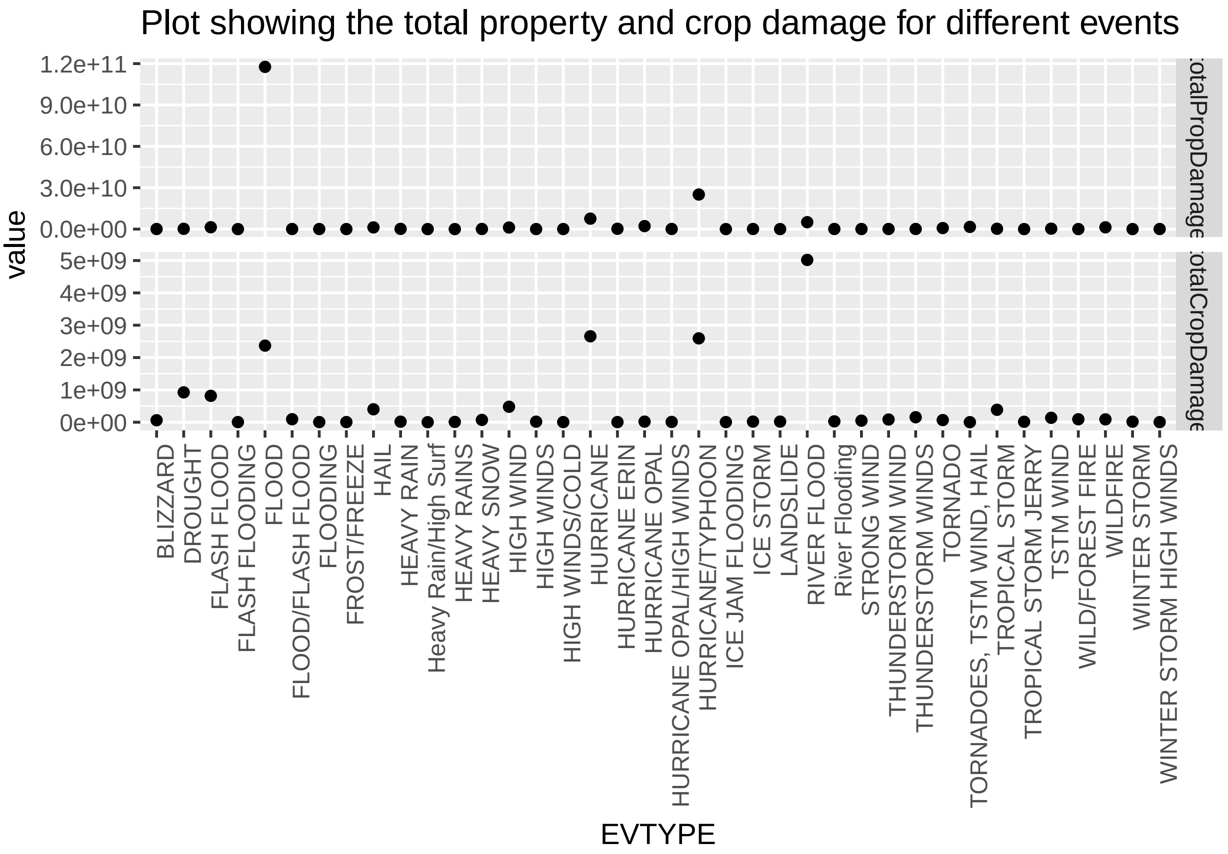
```
## # A tibble: 10 x 4
##    EVTYPE            totalFatalities totalInjuries totalHumanDamage
##    <fctr>                      <dbl>         <dbl>            <dbl>
##  1 TORNADO                    5633           91346           102612
##  2 EXCESSIVE HEAT             1903            6525            10331
##  3 TSTM WIND                   504            6957             7965
##  4 FLOOD                       470            6789             7729
##  5 LIGHTNING                   816            5230             6862
##  6 HEAT                        937            2100             3974
##  7 FLASH FLOOD                 978            1777             3733
##  8 ICE STORM                    89.0          1975             2153
##  9 THUNDERSTORM WIND           133            1488             1754
## 10 WINTER STORM                206            1321             1733
```

Property Damage

Plots the property and crop damage data side by side to help visualise any overall trends

```
##Melts the data to make it a long dataset (which can be plotted in ggplot2)
meltedDamageData <- melt(propDamageData %>% select(-totalDamage), id="EVTYPE")
g <- ggplot(meltedDamageData, aes(EVTYPE, value)) +
  facet_grid(variable~., scales="free") +
```

```r
  geom_point() +
  theme(axis.text.x = element_text(angle=90, hjust=1)) +
  ggtitle("Plot showing the total property and crop damage for different events")
print(g)
```



Displays the 10 events which cause the most overall financial impact to help inform any decision making (by providing precise information)

```r
head(arrange(propDamageData, desc(totalDamage), desc(totalPropDamage), desc(totalCropDamage)),10)
```

```
## # A tibble: 10 x 4
##    EVTYPE                     totalPropDamage totalCropDamage  totalDamage
##    <fctr>                               <dbl>           <dbl>        <dbl>
##  1 FLOOD                         117668720000      2368860000 120037580000
##  2 HURRICANE/TYPHOON              25073720000      2593840000  27667560000
##  3 HURRICANE                       7586870000      2658510000  10245380000
##  4 RIVER FLOOD                     5022800000      5019000000  10041800000
##  5 HURRICANE OPAL                  2168000000        19000000   2187000000
##  6 FLASH FLOOD                     1359230000       816510000   2175740000
##  7 HAIL                            1247370000       400800000   1648170000
##  8 HIGH WIND                       1140390000       478500000   1618890000
##  9 TORNADOES, TSTM WIND, HAIL      1600000000         2500000   1602500000
## 10 WILDFIRE                        1296220000        88900000   1385120000
```