NOAA Storm Data Report

Synopsis

This report looks at the damage caused to both human populations and economic systems over a 50 year period by different types of weather events. The goal of this analysis is to provide administrators and emergency managers with information that will help them predict the severity of different types of weather and forecast resources appropriately.

Data Loading and Processing

The dataset is downloaded from the cloudfront address below, and loaded into the variable storm_data. Documentation for the dataset can be found at this link (https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf).

```
  if (!file.exists("storm_data.csv.bz2")) {
  download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz
2", destfile = "storm_data.csv.bz2")
  }
  storm_data <- read.csv("storm_data.csv.bz2")
```

Package requirements:

```
  require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

Results

Harm to Population Health

First, we set out to remove all the observations which had no impact on population health (meaning they caused neither injuries nor fatalities). These events were then used to generate a factor, so we could go back to the original dataset and select only EVTYPEs that had ever caused a fatality or an injury.

```
  health_data <- storm_data[(storm_data$INJURIES != 0) | (storm_data$FATALITIES != 0),]
  health_data$EVTYPE <- factor(health_data$EVTYPE)
  percent_health_danger <- nlevels(health_data$EVTYPE) / nlevels(storm_data$EVTYPE)
```

As percent_health_danger shows above, only **0.2233503**% of event types have ever caused a fatality or injury, so we can safely eliminate the majority of event types. To accomplish this, the factor health_data$EVTYPE is applied to storm_data to produce another subset that includes only the events that have ever caused population health damage.

```
  pop_danger_data <- storm_data[storm_data$EVTYPE %in% health_data$EVTYPE,]
  pop_danger_data$EVTYPE <- factor(pop_danger_data$EVTYPE)
```

This step is necessary because analyzing only the specific events that have caused fatalities could lead to misleading results. We must also know the total number of observations for each event type, so that we can determine the likelihood that a given event will cause population health damage.

To determine which events have the greatest likelihood to cause any amount population damage, we need to calculate the likelihood for each possible event type:

```
  damage_percents <- data.frame(eventType=character(), damageLikelihood=numeric(), num_ob
s=numeric(), damage_score=numeric(), avg_dmg_score=numeric())
  for (level in levels(pop_danger_data$EVTYPE)) {
    subset <- pop_danger_data[(pop_danger_data$EVTYPE == level),]
    total_obs <- nrow(subset)
    damage_subset <- nrow(pop_danger_data[(pop_danger_data$EVTYPE == level) & (pop_dange
r_data$INJURIES != 0 | pop_danger_data$FATALITIES != 0),])
    total_injury_death <- (sum(subset$INJURIES) * .5) + (sum(subset$FATALITIES))
    avg_dmg_score <- total_injury_death / total_obs
    likelihood <- damage_subset/total_obs
    damage_percents <- rbind(damage_percents, data.frame(eventType=level, damageLikelihoo
d=likelihood, num_obs=total_obs, damage_score=total_injury_death, avg_dmg_score=avg_dmg_s
core))
  }
```

We can get a good idea of how destructive a given event is by looking at the avg_dmg_score column that was calculated in the above loop. It is equal to the total damage (a arbitrary number defined as 1 point for a fatality, 1/2 point for an injury) caused by an event type, divided by the total number of observations of that event on record. A summary of the avg_dmg_score gives us some interesting information:

```
  summary(damage_percents$avg_dmg_score)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00022  0.09890  0.50000  1.64500  1.00000 35.00000
```

Apparently, there is a big jump in the top quartile of the average population damage - from 1.0 to 35. For this reason, the mean value is actually higher than the 3rd quartile. Looking into this further:

```
  quantile(damage_percents$avg_dmg_score, c(.90, .925, .95, .975, .99))
```

```
##       90%     92.5%       95%     97.5%       99% 
##  3.500000  4.821875  7.890625 13.262500 23.955000
```

The difference between the average damage of events in the 97.5th percentile and those in the 99th percentile is remarkable. We can be confident that the data in the 90th percentile and above are, on average, by far the most damaging to population health. These data will now be subsetted out and plotted.

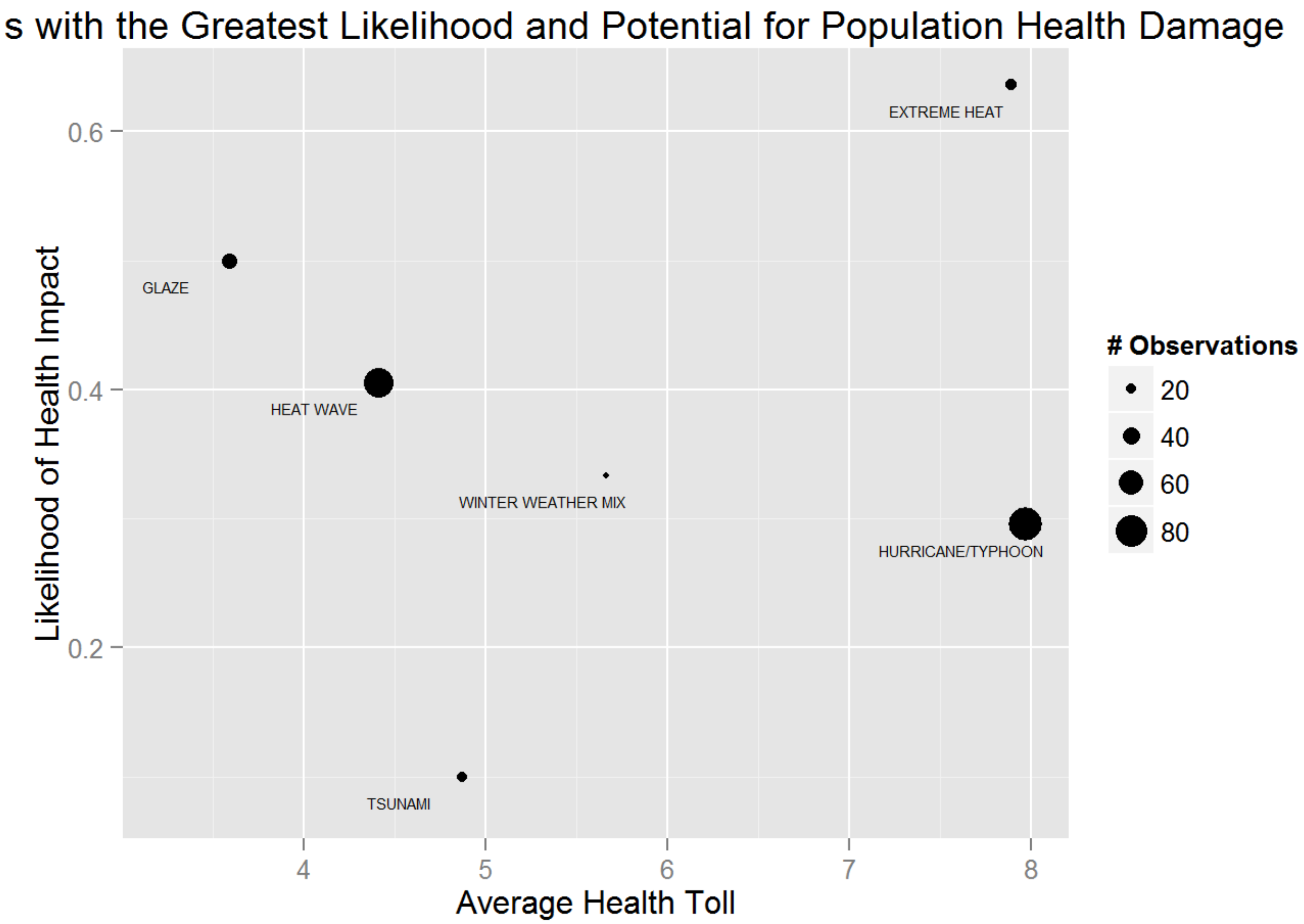
```
  final_pop_data <- damage_percents[damage_percents$avg_dmg_score >= quantile(damage_perc
ents$avg_dmg_score, c(.90))[1],]
  final_pop_data[,c(1,2,3,5)]
```

```
##                      eventType damageLikelihood num_obs avg_dmg_score
## 17               COLD AND SNOW        1.0000000       1     14.000000
## 39                EXTREME HEAT        0.6363636      22      7.886364
## 61                       GLAZE        0.5000000      32      3.593750
## 62             GLAZE/ICE STORM        1.0000000       1      7.500000
## 70                   Heat Wave        1.0000000       1     35.000000
## 71                   HEAT WAVE        0.4054054      74      4.412162
## 72           HEAT WAVE DROUGHT        1.0000000       1     11.500000
## 95          HIGH WIND AND SEAS        1.0000000       1     13.000000
## 97              HIGH WIND/SEAS        1.0000000       1      4.000000
## 109          HURRICANE/TYPHOON        0.2954545      88      7.971591
## 130              MARINE MISHAP        1.0000000       2      4.750000
## 138     NON-SEVERE WIND DAMAGE        1.0000000       1      3.500000
## 146      RECORD/EXCESSIVE HEAT        0.3333333       3      5.666667
## 154                ROUGH SEAS        1.0000000       3      3.500000
## 164             SNOW/HIGH WINDS        1.0000000       2      9.000000
## 181               THUNDERSTORMW        1.0000000       1     13.500000
## 187 TORNADOES, TSTM WIND, HAIL        1.0000000       1     25.000000
## 190      TROPICAL STORM GORDON        1.0000000       1     29.500000
## 196                    TSUNAMI        0.1000000      20      4.875000
## 208                 WILD FIRES        0.2500000       4     19.500000
## 215    WINTER STORM HIGH WINDS        1.0000000       1      8.500000
## 216              WINTER STORMS        0.3333333       3      6.166667
## 218          WINTER WEATHER MIX        0.3333333       6      5.666667
```

Clearly, some of these are outliers - a good way to clean out some of them is to remove all the items for which there are less than 5 observations in this 50 year period (e.g., “Heat Wave” vs. “HEAT WAVE”, or “TROPICAL STORM GORDON”, which refers to one specific event, not a type of event). That will be accomplished in the plotting phase, below.

```
  final_pop_data <- final_pop_data[final_pop_data$num_obs >= 5,]
  pop_plot <- qplot(avg_dmg_score, damageLikelihood, data=final_pop_data, xlab="Average H
ealth Toll", ylab="Likelihood of Health Impact", size=num_obs)
  pop_plot <- pop_plot + labs(title = "Events with the Greatest Likelihood and Potential
for Population Health Damage")
  pop_plot <- pop_plot + labs(size = "# Observations")
  pop_plot <- pop_plot + annotate("text", x=(final_pop_data$avg_dmg_score - 0.35), y=(fin
al_pop_data$damageLikelihood - 0.02), label=final_pop_data$eventType, size=2)
```

The resulting graph gives a good idea of the average toll, likelihood of damage, and confidence (based on number of observations) for the most serious event types:



Harm to Economic Health

As before, we set out to remove all the observations which had no impact on economic health (meaning they caused neither crop nor property damage). These events were then used to generate a factor, so we could go back to the original dataset and select only EVTYPEs that had ever caused economic damage.

```
  economic_data <- storm_data[(storm_data$PROPDMG != 0) | (storm_data$CROPDMG != 0),]
  economic_data$EVTYPE <- factor(economic_data$EVTYPE)
  percent_economic_danger <- nlevels(economic_data$EVTYPE) / nlevels(storm_data$EVTYPE)
```

As percent_economic_danger shows above, only **0.4375635**% of event types have ever caused a economic damage, so we can safely eliminate the majority of event types. To accomplish this, the factor economic_data$EVTYPE is applied to storm_data to produce another subset that includes only the events that have ever caused economic damage.

```
  eco_danger_data <- storm_data[storm_data$EVTYPE %in% economic_data$EVTYPE,]
  eco_danger_data$EVTYPE <- factor(eco_danger_data$EVTYPE)
```

Now we need to standardize the exponents in CROPDMGEXP and PROPDMGEXP, and modify the values in PROPDMG and CROPDMG based on them. We'll also remove all rows without a multiplier, as they're either incomplete or too small to matter.

```
  eco_danger_data$PROPDMGEXP <- toupper(eco_danger_data$PROPDMGEXP)
  eco_danger_data$CROPDMGEXP <- toupper(eco_danger_data$CROPDMGEXP)
  eco_danger_data <- eco_danger_data[eco_danger_data$CROPDMGEXP != "" & eco_danger_data$P
ROPDMGEXP != "",]
  eco_danger_data$PROPDMGEXP[eco_danger_data$PROPDMGEXP == "K"] <- 1000
  eco_danger_data$PROPDMGEXP[eco_danger_data$PROPDMGEXP == "M"] <- 100000
  eco_danger_data$PROPDMGEXP[eco_danger_data$PROPDMGEXP == "B"] <- 100000000
  eco_danger_data$CROPDMGEXP[eco_danger_data$CROPDMGEXP == "K"] <- 1000
  eco_danger_data$CROPDMGEXP[eco_danger_data$CROPDMGEXP == "M"] <- 100000
  eco_danger_data$CROPDMGEXP[eco_danger_data$CROPDMGEXP == "B"] <- 100000000
  eco_danger_data$CROPDMGEXP <- as.numeric(eco_danger_data$CROPDMGEXP)
```

```
## Warning: NAs introduced by coercion
```

```
  eco_danger_data$PROPDMGEXP <- as.numeric(eco_danger_data$PROPDMGEXP)
```

Now we apply the multipliers to the CROPDMG and PROPDMG columns, and add them to get a total dollar value:

```
  eco_danger_data$TOTAL <- eco_danger_data$PROPDMG * eco_danger_data$PROPDMGEXP + eco_dan
ger_data$CROPDMG * eco_danger_data$CROPDMGEXP
```

Finally, we can use a for loop to create a new data frame that contains the total costs by EVTYPE.

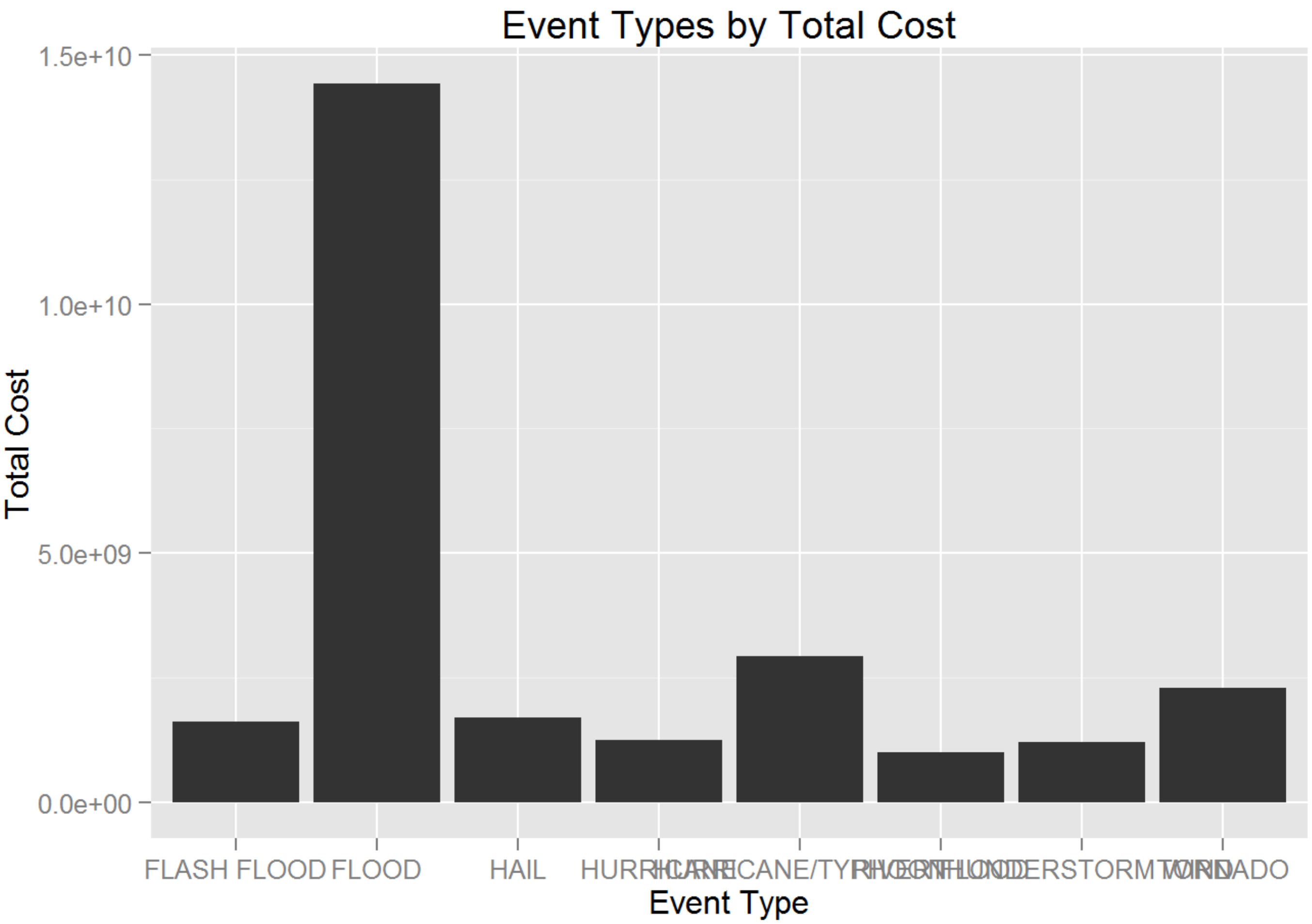
```
  eco_total_costs <- data.frame(eventType=character(), damageTotal=numeric())
  for (level in levels(pop_danger_data$EVTYPE)) {
    subset <- eco_danger_data[(eco_danger_data$EVTYPE == level),]
    total_cost <- sum(subset$TOTAL)
    eco_total_costs <- rbind(eco_total_costs, data.frame(eventType=level, damageTotal=tot
al_cost))
  }
  eco_total_costs <- eco_total_costs[complete.cases(eco_total_costs),]
```

Now we need to remove cases where the total cost is 0, and isolate the 95th percentile to only see the most damaging EVTYPEs:

```
  eco_total_costs <- eco_total_costs[eco_total_costs$damageTotal > 0,]
  final_eco_data <- eco_total_costs[eco_total_costs$damageTotal >= quantile(eco_total_cos
ts$damageTotal, c(.90))[1],]
```

A quick plot makes the answer to this problem very clear:

```
  eco_plot <- qplot(x=eventType, y=damageTotal, data=final_eco_data, geom="bar", stat="id
entity", xlab="Event Type", ylab="Total Cost", main="Event Types by Total Cost")
  eco_plot
```



The event with the greatest economic impacts are floods.