Analysis of Health and Economic Impacts of U.S. Storms

Clarence Brooks

9/14/2020

1: Synopsis

The goal of this analysis is to explore the NOAA Storm Database and examine the effects of severe weather events on both population and economy. The database covers the time period between 1950 and November 2011.

This analysis investigates which types of severe weather events are most harmful on: 1. Health (Injuries and Fatalities) 2. Property and Crops (Economic Consequences)

Documentation on the Data: Documentation

2: Data Processing

2.1: Data Loading

Read the data into a dataframe, then convert it to a data.table.

```
library("data.table")
library("ggplot2")
# NOTE: This assumes that the .csv file has already been extracted
# and is present in the current working directory.
stormDF <- read.csv("repdata_data_StormData.csv")
# Converting data.frame to data.table
stormDT <- as.data.table(stormDF)
```

2.2: Examine Column Names

```
colnames(stormDT)
```

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

2.3: Data Subsetting

Subset the dataset on the parameters of interest. Basically, remove the unwanted columns.

```
# Find the columns to remove.
cols2Remove <- colnames(stormDT[, !c("EVTYPE"
  , "FATALITIES"
```

```
  , "INJURIES"
  , "PROPDMG"
  , "PROPDMGEXP"
  , "CROPDMG"
  , "CROPDMGEXP")])
# Remove the unwanted columns.
stormDT[, c(cols2Remove) := NULL]
# Only keep the data where injuries, fatalities, property damage,
# or crop damage occurred.
stormDT <- stormDT[(EVTYPE != "?" &
             (INJURIES > 0 | FATALITIES > 0 | PROPDMG > 0 | CROPDMG > 0)), c("EVTYPE"
                                                                             , "FATALITIES"
                                                                             , "INJURIES"
                                                                             , "PROPDMG"
                                                                             , "PROPDMGEXP"
                                                                             , "CROPDMG"
                                                                             , "CROPDMGEXP") ]
```

2.4: Convert Exponent Columns into Actual Exponents instead of (-,+, H, K, etc)

Make the PROPDMGEXP and CROPDMGEXP columns cleaner so they can be used to calculate property and crop cost.

```
# Change all damage exponents to uppercase.
cols <- c("PROPDMGEXP", "CROPDMGEXP")
stormDT[,  (cols) := c(lapply(.SD, toupper)), .SDcols = cols]
# Convert property damage alphanumeric exponents to numeric values.
propDmgKey <-  c("\"\"" = 10^0,
                 "-" = 10^0,
                 "+" = 10^0,
                 "0" = 10^0,
                 "1" = 10^1,
                 "2" = 10^2,
                 "3" = 10^3,
                 "4" = 10^4,
                 "5" = 10^5,
                 "6" = 10^6,
                 "7" = 10^7,
                 "8" = 10^8,
                 "9" = 10^9,
                 "H" = 10^2,
                 "K" = 10^3,
                 "M" = 10^6,
                 "B" = 10^9)
# Convert crop damage alphanumeric exponents to numeric values.
cropDmgKey <-  c("\"\"" = 10^0,
                 "?" = 10^0,
                 "0" = 10^0,
                 "K" = 10^3,
                 "M" = 10^6,
                 "B" = 10^9)
stormDT[, PROPDMGEXP := propDmgKey[as.character(stormDT[,PROPDMGEXP])]]
stormDT[is.na(PROPDMGEXP), PROPDMGEXP := 10^0 ]
stormDT[, CROPDMGEXP := cropDmgKey[as.character(stormDT[,CROPDMGEXP])] ]
```

```
stormDT[is.na(CROPDMGEXP), CROPDMGEXP := 10^0 ]
```

2.5: Making Economic Cost Columns

```
stormDT <- stormDT[, .(EVTYPE, FATALITIES, INJURIES, PROPDMG, PROPDMGEXP, propCost = PROPDMG * PROPDMGEX
```

2.6: Calculating Total Property and Crop Cost

```
totalCostDT <- stormDT[, .(propCost = sum(propCost), cropCost = sum(cropCost), Total_Cost = sum(propCost
totalCostDT <- totalCostDT[order(-Total_Cost), ]
totalCostDT <- totalCostDT[1:10, ]
head(totalCostDT, 5)
```

```
##               EVTYPE     propCost   cropCost   Total_Cost
## 1:             FLOOD 144657709807 5661968450 150319678257
## 2: HURRICANE/TYPHOON  69305840000 2607872800  71913712800
## 3:           TORNADO  56947380676  414953270  57362333946
## 4:       STORM SURGE  43323536000       5000  43323541000
## 5:              HAIL  15735267513 3025954473  18761221986
```

2.7: Calculating Total Fatalities and Injuries

```
totalInjuriesDT <- stormDT[, .(FATALITIES = sum(FATALITIES), INJURIES = sum(INJURIES), totals = sum(FATA
totalInjuriesDT <- totalInjuriesDT[order(-FATALITIES), ]
totalInjuriesDT <- totalInjuriesDT[1:10, ]
head(totalInjuriesDT, 5)
```

```
##            EVTYPE FATALITIES INJURIES totals
## 1:        TORNADO       5633    91346  96979
## 2: EXCESSIVE HEAT       1903     6525   8428
## 3:    FLASH FLOOD        978     1777   2755
## 4:           HEAT        937     2100   3037
## 5:      LIGHTNING        816     5230   6046
```

3: Results

3.1: Events that are Most Harmful to Population Health

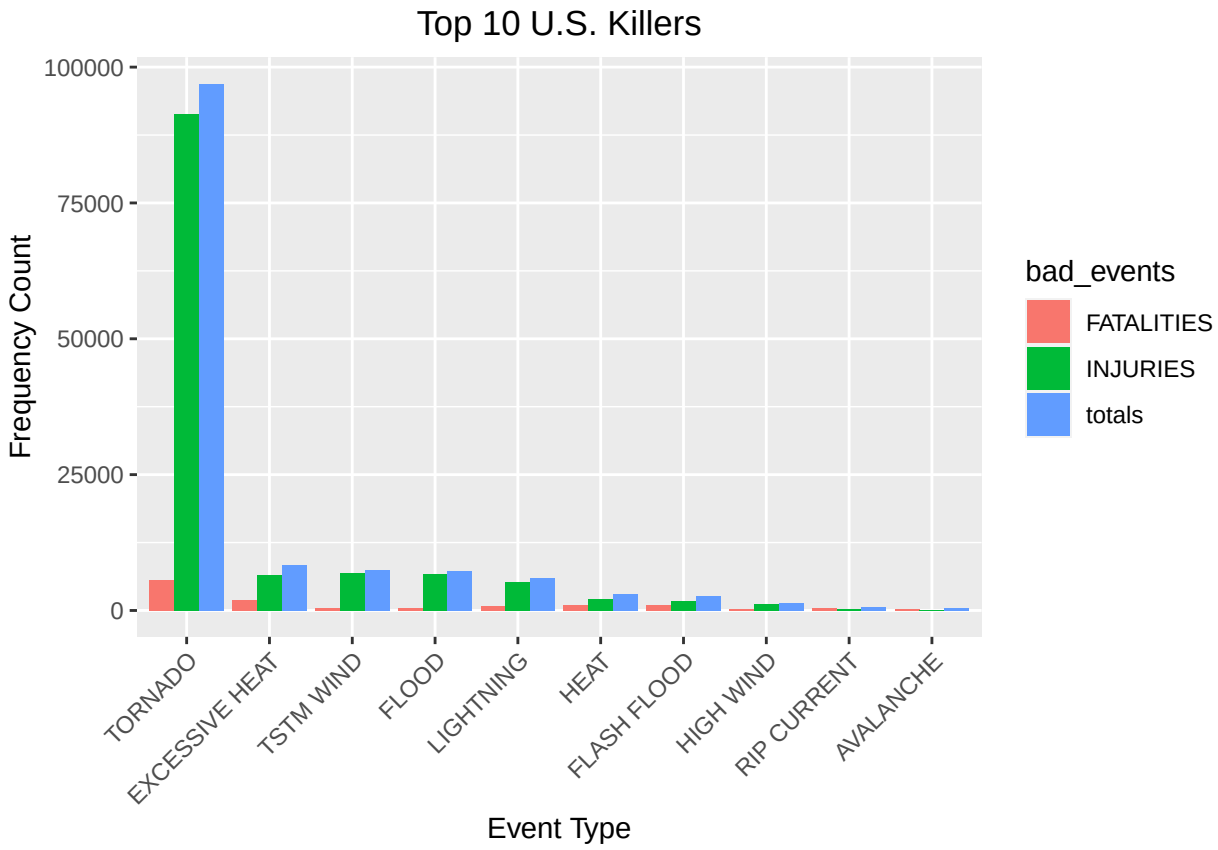
Melting data.table so that it is easier to put in bar graph format

```
harmful_events <- melt(totalInjuriesDT, id.vars="EVTYPE", variable.name = "bad_events")
head(harmful_events, 5)
```

```
##            EVTYPE bad_events value
## 1:        TORNADO FATALITIES  5633
## 2: EXCESSIVE HEAT FATALITIES  1903
## 3:    FLASH FLOOD FATALITIES   978
## 4:           HEAT FATALITIES   937
## 5:      LIGHTNING FATALITIES   816
```

```
# Create chart.
healthChart <- ggplot(harmful_events, aes(x=reorder(EVTYPE, -value), y=value))
# Plot data as bar chart.
healthChart = healthChart + geom_bar(stat="identity", aes(fill=bad_events), position="dodge")
# Format y-axis scale and set y-axis label.
```

```
healthChart = healthChart + ylab("Frequency Count")
# Set x-axis label
healthChart = healthChart + xlab("Event Type")
# Rotate x-axis labels
healthChart = healthChart + theme(axis.text.x = element_text(angle=45, hjust=1))
# Set chart title and center it
healthChart = healthChart + ggtitle("Top 10 U.S. Killers") + theme(plot.title = element_text(hjust = 0.5
healthChart
```



3.2: Events that have the Greatest Economic Consequences

Melting data.table so that it is easier to put in bar graph format.

```
econ_consequences <- melt(totalCostDT, id.vars="EVTYPE", variable.name = "Damage_Type")
head(econ_consequences, 5)
```

```
##                EVTYPE Damage_Type        value
## 1:             FLOOD    propCost 144657709807
## 2: HURRICANE/TYPHOON    propCost  69305840000
## 3:           TORNADO    propCost  56947380676
## 4:       STORM SURGE    propCost  43323536000
## 5:              HAIL    propCost  15735267513
```

```
# Create chart
econChart <- ggplot(econ_consequences, aes(x=reorder(EVTYPE, -value), y=value))
# Plot data as bar chart
econChart = econChart + geom_bar(stat="identity", aes(fill=Damage_Type), position="dodge")
# Format y-axis scale and set y-axis label
```

```
econChart = econChart + ylab("Cost (dollars)")
# Set x-axis label
econChart = econChart + xlab("Event Type")
# Rotate x-axis tick labels
econChart = econChart + theme(axis.text.x = element_text(angle=45, hjust=1))
# Set chart title and center it
econChart = econChart + ggtitle("Top 10 U.S. Storm Events causing Economic Damages") + theme(plot.title
econChart
```