Storm Analysis

Nick Daughenbaugh

August 6, 2020

1: Assignment

The basic goal of this assignment is to explore the NOAA Storm Database and answer some basic questions about severe weather events. You must use the database to answer the questions below and show the code for your entire analysis. Your analysis can consist of tables, figures, or other summaries. You may use any R package you want to support your analysis.

Questions Your data analysis must address the following questions:

1. Across the United States, which types of events are most harmful with respect to population health?
2. Across the United States, which types of events have the greatest economic consequences?

2: Data Analysis

2.1: Downloading & Reading-in Data

Download the raw data file and extract the data into a dataframe.Then convert to a data.table

```
library("data.table")
library("ggplot2")
library("tinytex")
fileUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
download.file(fileUrl, destfile = paste0("C:/Users/nrdau/Documents/R/data/StormData.csv.bz2"))
stormDF <- read.csv("C:/Users/nrdau/Documents/R/data/StormData.csv.bz2")
# Converting data.frame to data.table
stormDT <- as.data.table(stormDF)
```

2.2: Expoling Data Fields

```
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

Remove the fields we don't need.

```
# Finding fields to (not) remove
Remove <- colnames(stormDT[, !c("EVTYPE"
  , "FATALITIES"
  , "INJURIES"
  , "PROPDMG"
  , "PROPDMGEXP"
  , "CROPDMG"
  , "CROPDMGEXP")])
# Removing columns
stormDT[, c(Remove) := NULL]
# Only use data where fatalities or injuries occurred.
stormDT <- stormDT[(EVTYPE != "?" &
             (INJURIES > 0 | FATALITIES > 0 | PROPDMG > 0 | CROPDMG > 0)), c(
               "EVTYPE",
               "FATALITIES",
               "INJURIES",
               "PROPDMG",
               "PROPDMGEXP",
               "CROPDMG",
               "CROPDMGEXP") ]
```

2.4: Converting Exponent Prefixes

Covert the PROPDMGEXP and CROPDMGEXP fields to something useful.

```
# Change all damage exponents to uppercase.
EXPcols <- c("PROPDMGEXP", "CROPDMGEXP")
stormDT[,  (EXPcols) := c(lapply(.SD, toupper)), .SDcols = EXPcols]
# Map 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 prefix 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: Economic Damage

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

2.6: Total Property and Crop Cost

```
totalCostDT <- stormDT[, .(propCost = sum(propCost), cropCost = sum(cropCost), Total_Cost = sum(propCos
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  56947380677  414953270  57362333947
## 4:       STORM SURGE  43323536000       5000  43323541000
## 5:              HAIL  15735267513 3025954473  18761221986
```

2.7: Total Fatalities and Injuries

```
totalInjuriesDT <- stormDT[, .(FATALITIES = sum(FATALITIES), INJURIES = sum(INJURIES), totals = sum(FAT
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: Conclusion

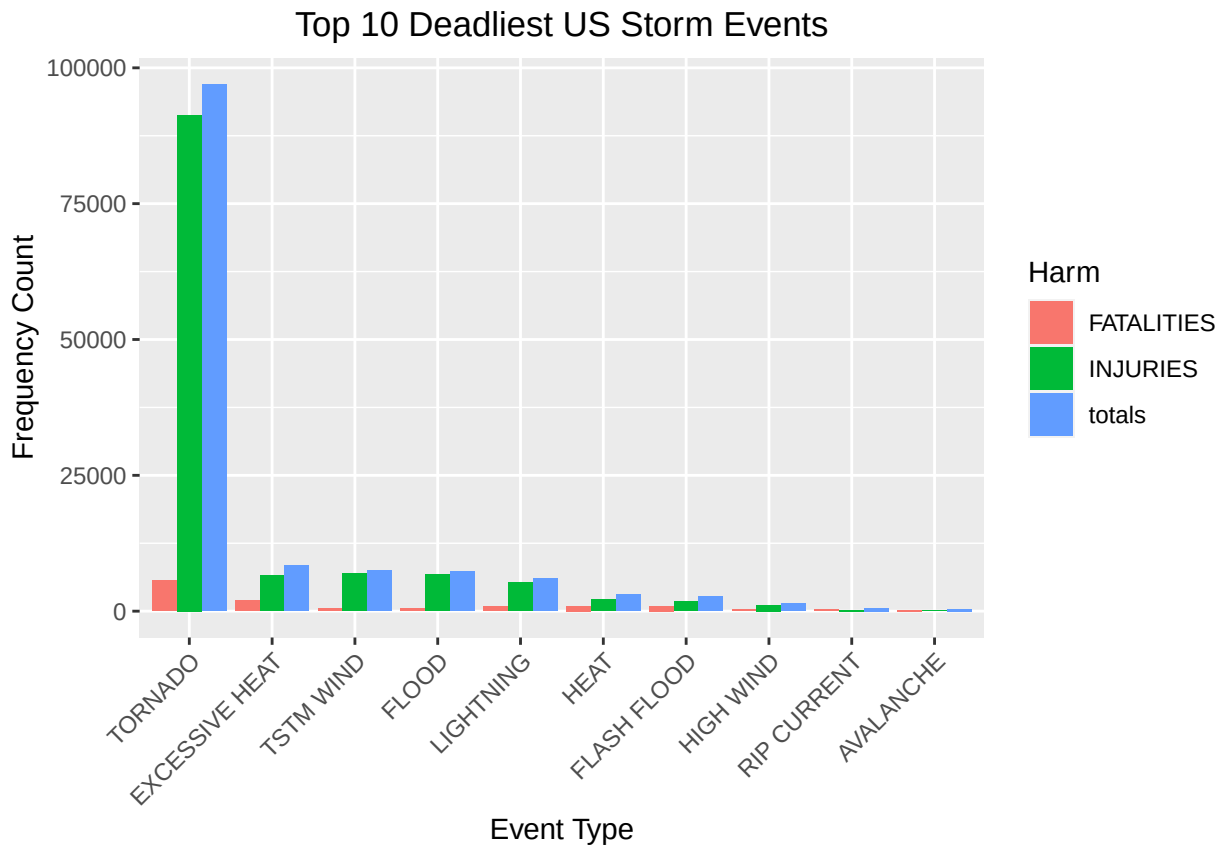
3.1: The Most Harmful Events to Population

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

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

```
##            EVTYPE       Harm 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, aes(x=reorder(EVTYPE, -value), y=value))
# Plot data as bar chart
healthChart = healthChart + geom_bar(stat="identity", aes(fill=Harm), 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 tick labels
healthChart = healthChart + theme(axis.text.x = element_text(angle=45, hjust=1))
# Set chart title and center it
healthChart = healthChart + ggtitle("Top 10 Deadliest US Storm Events") + theme(plot.title = element_te
healthChart
```



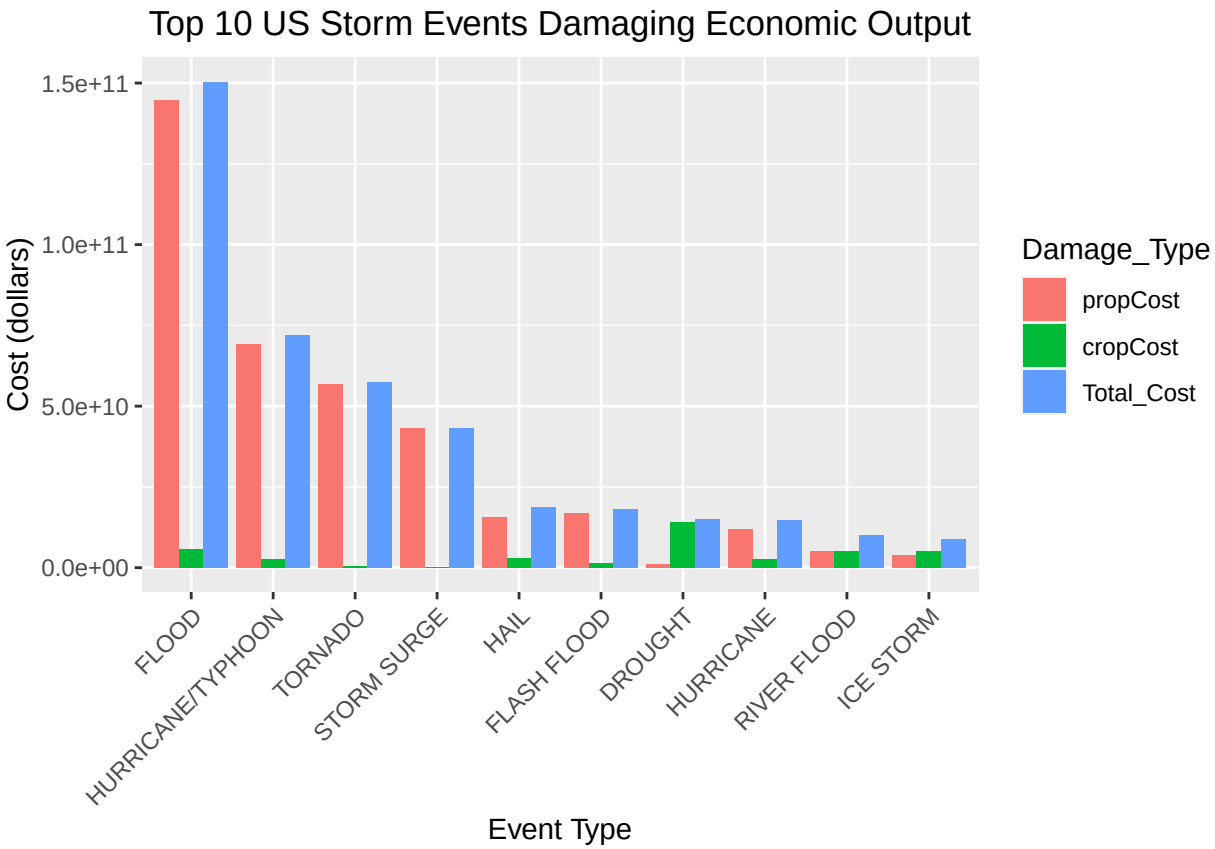
3.2: The Most Harmful Events to Economic Output

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

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

```
##                EVTYPE Damage_Type        value
## 1:              FLOOD    propCost 144657709807
## 2: HURRICANE/TYPHOON    propCost  69305840000
## 3:            TORNADO    propCost  56947380677
## 4:        STORM SURGE    propCost  43323536000
## 5:               HAIL    propCost  15735267513
```

```
# Create chart
econChart <- ggplot(econ_output, 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 US Storm Events Damaging Economic Output") + theme(plot.title =
econChart
```


Top 10 US Storm Events Damaging Economic Output
Cost (dollars)
1.5e+11
1.0e+11
5.0e+10
0.0e+00
FLOOD
HURRICANE/TYPHOON
TORNADO
STORM SURGE
HAIL
FLASH FLOOD
DROUGHT
HURRICANE
RIVER FLOOD
ICE STORM
Event Type
Damage_Type
propCost
cropCost
Total_Cost