Analysis: Storms and other severe weather events in public health and economic problems for communities and municipalities in US Storms.

DavidC.

31/08/2019

Github repository for the Course: Reproducible Research

1: Synopsis

The goal of the assignment is to explore the U.S. National Oceanic and Atmospheric Administration's (NOAA) Storm Database and explore the effects of severe weather events on both population and economy. The events in the database start in the year 1950 and end in November 2011.

The following analysis investigates which types of severe weather events are most harmful on:

1. Health (Injuries and Fatalities)
2. Property damage and crops (Economic Consequences)

Source of information on the data: Storm Data Documentation

2: Environment Preparation

2.1: Setup libs and download data

```
# Install needed libs if needed
#install.packages("ggplot2")
#install.packages("data.table")
#library("knitr")

# Load the library's into R Studio memory.
setwd("~/coursera/data")
library("data.table")
```

```
library("ggplot2")
library(knitr)


# Download and read the CSV files into stromDateFrame.
sourceFileUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
download.file(sourceFileUrl, destfile = paste0(getwd(), './repdata_data_StormData.csv.bz2'))
## Warning in download.file(sourceFileUrl, destfile =
## paste0(getwd(), "./repdata_data_StormData.csv.bz2")): URL https://
## d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2:
## cannot open destfile '/Users/davidecastellano/coursera/data./
## repdata_data_StormData.csv.bz2', reason 'No such file or directory'
## Warning in download.file(sourceFileUrl, destfile = paste0(getwd(), "./
## repdata_data_StormData.csv.bz2")): download had nonzero exit status
stromDateFrame <- read.csv("./repdata_data_StormData.csv.bz2")


# Converting data.frame to data.table
stromDateTable <- as.data.table(stromDateFrame)
```

2.2: Data Structure Verification (optional).

```
# Display list of column's for the data.table (stromDateTable).
colnames(stromDateTable)
##  [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 Wrangling (cleaning)

Subset the dataset on the parameters of interest. Basically, we remove the columns we don't need for clarity.

```
# Finding columns to remove
cols_to_Remove <- colnames(stromDateTable[, !c("EVTYPE"
                                               , "FATALITIES"
                                               , "INJURIES"
                                               , "PROPDMG"
                                               , "PROPDMGEXP"
                                               , "CROPDMG"
                                               , "CROPDMGEXP")])


# Removing columns
stromDateTable[, c(cols_to_Remove) := NULL]


# Only use data where fatalities or injuries occurred.
stromDateTable <- stromDateTable[(EVTYPE != "?" &
                                          (INJURIES > 0 | FATALITIES > 0 | PR
OPDMG > 0 | CROPDMG > 0)), c("EVTYPE"

, "FATALITIES"

, "INJURIES"

, "PROPDMG"

, "PROPDMGEXP"

, "CROPDMG"

, "CROPDMGEXP") ]
```

2.4: Data Wrangling (translation)

Making 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")
```

```
stromDateTable[,  (cols) := c(lapply(.SD, toupper)), .SDcols = cols]


# 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)


# Map 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)


stromDateTable[, PROPDMGEXP := propDmgKey[as.character(stromDateTable[, PROPD
MGEXP])]]
stromDateTable[is.na(PROPDMGEXP), PROPDMGEXP := 10^0 ]


stromDateTable[, CROPDMGEXP := cropDmgKey[as.character(stromDateTable[, CROPD
MGEXP])] ]
```

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


stromDateTable <- stromDateTable[, .(EVTYPE, FATALITIES, INJURIES, PROPDMG, P
ROPDMGEXP, propCost = PROPDMG * PROPDMGEXP, CROPDMG, CROPDMGEXP, cropCost = C
ROPDMG * CROPDMGEXP)]
```

3: Data Processing and Verification

3.1: Table: Total Property and Crop Cost.

```
totalCostDataTable <- stromDateTable[, .(propCost = sum(propCost), cropCost =
sum(cropCost), Total_Cost = sum(propCost) + sum(cropCost)), by = .(EVTYPE)]


totalCostDataTable <- totalCostDataTable[order(-Total_Cost), ]


totalCostDataTable <- totalCostDataTable[1:10, ]


head(totalCostDataTable, 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
```

3.2: Table: Total Fatalities and Injuries.

```
totalInjuriesDataTable <- stromDateTable[, .(FATALITIES = sum(FATALITIES), IN
JURIES = sum(INJURIES), totals = sum(FATALITIES) + sum(INJURIES)), by = .(EVT
YPE)]


totalInjuriesDataTable <- totalInjuriesDataTable[order(-FATALITIES), ]


totalInjuriesDataTable <- totalInjuriesDataTable[1:10, ]


head(totalInjuriesDataTable, 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
```

4: Results

4.1: Events that are Most Harmful to Population Health

4.1.1: Table: Top 10 US Killers

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

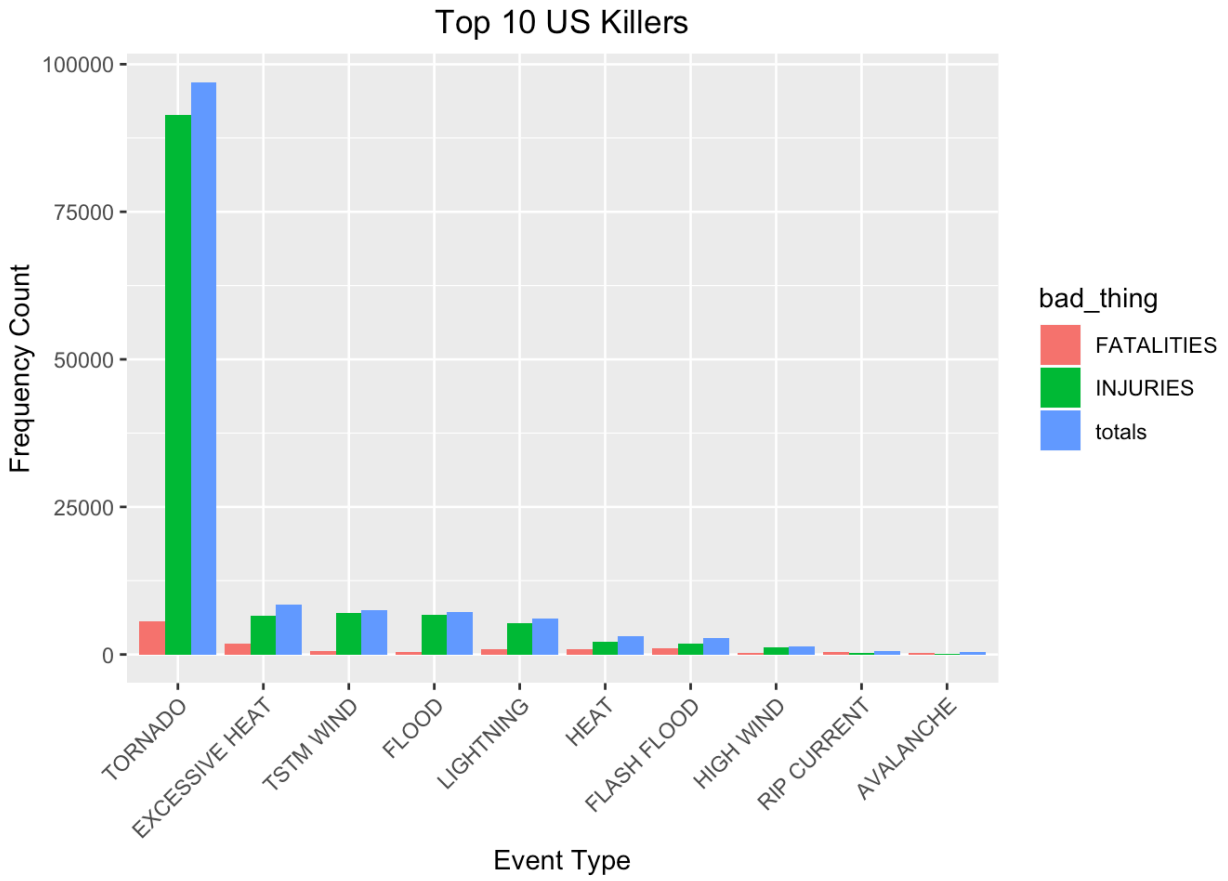
```
bad_stuff <- melt(totalInjuriesDataTable, id.vars="EVTYPE", variable.name = "
bad_thing")
head(bad_stuff, 5)
```

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

4.1.1: Chart: Top 10 US Killers

```
# Create chart
healthChart <- ggplot(bad_stuff, aes(x = reorder(EVTYPE, -value), y = value))


# Plot data as bar chart
healthChart = healthChart + geom_bar(stat = "identity", aes(fill = bad_thing)
, 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, hjus
t = 1))


# Set chart title and center it
healthChart = healthChart + ggtitle("Top 10 US Killers") + theme(plot.title =
element_text(hjust = 0.5))


healthChart
```



4.2: Events that have the Greatest Economic Consequences

4.2.1: Table: Top 10 US Storm Events causing Economic Consequences

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

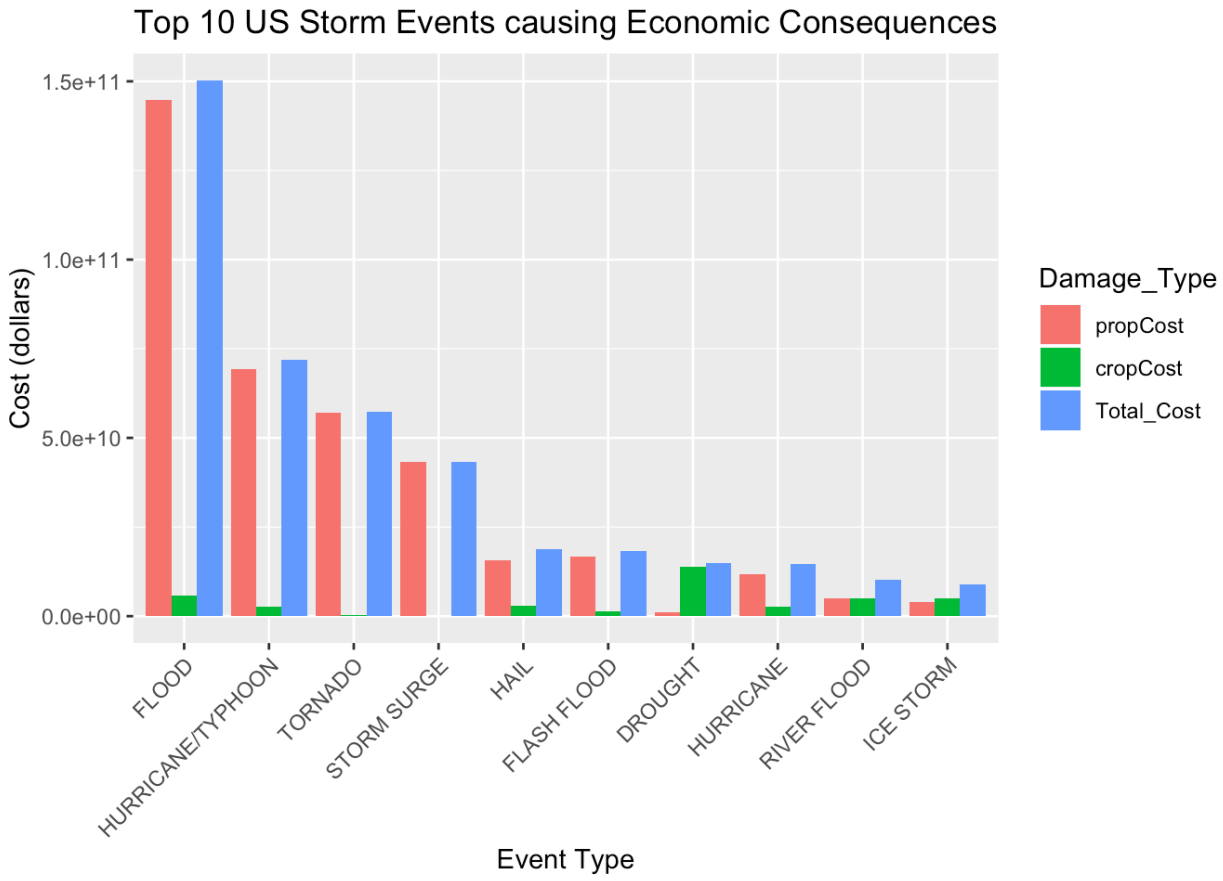
```
econ_consequences <- melt(totalCostDataTable, id.vars="EVTYPE", variable.name
= "Damage_Type")
```

```
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
```

4.2.2: Chart: Top 10 US Storm Events causing Economic Consequences

```
# Create chart

econChart <- ggplot(econ_consequences, aes(x = reorder(EVTYPE, -value), y = v
alue))


# 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 causing Economic Cons
equences") + theme(plot.title = element_text(hjust = 0.5))


econChart
```


Top 10 US Storm Events causing Economic Consequences
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