Reproducible Research: Course Project 2

Analysis of the Most Dangerous and Most Expensive Weather Events using the NOAA Storm Database

Synopsis:

This analysis examines the data from the NOAA Storm Database that records severe weather events along with estimates of fatalities, injuries, property damage, and crop damage. In this analysis, the NOAA database is categorized by the Weather Event Type (or, "Event Name") and examines the Event Types that are: 1) Most Dangerous in terms of number of Injuries and Fatalities; and 2) Most Expensive (economic impact) in terms of amount of Property Damage and Crop damage. Not surprisingly, Tornadoes, Ice, Wind, and Floods are represented in the top among the Most Dangerous and Most Expensive (greatest economic consequences) Weather Event Types.

Author: Andrew D. Stewart 26 August 2017

Load Libraries

First, load R libraries: data.table, dplyr, lubridate, ggplot2, datasets, xtable, knitr, and stringer.

Load The Data

Load the data from a file named **repdata_data_StormData.csv** that was provided from the Coursera website. The events in the database start in the year 1950 and end in November 2011. Additional details of the data can be found at:http://www.nws.noaa.gov/directives/

```
knitr::opts_chunk$set(fig.width=12, fig.height=8, fig.path='', echo = TRUE,
                      fig.keep = "all", results = "asis")

## Read activity.csv and format as tbl_df
actyData <- read.csv("repdata_data_StormData.csv", header = TRUE, sep =",")
actyData <- tbl_df(actyData)
```

DATA PROCESSING

The data is manipulated as a `tbl_df` dataframe table which is sorted and grouped by the Weather Event Type. Columns of variables not relevant to this analysis are removed. Columns that relate to Injuries, Fatalities, Property Damage, and Crop Damage are then manipulated in the following ways. Property and Crop Damage data were originally stored with "B/M/K" varrying exponent multiplier weightings (billions/millions/thousands); the respective weights are applied to the base data so that all damage values are in the same base ("ones") amounts.

The number of Fatalities, Injuries, Property Damage, and Crop Damage are `summarise`ed by Weather Event Type. Two new variable columns are created which represent the Total Sums of all (Injuries + Fatalities) and (Property Damage + Crop Damage) for each Weather Event Type.
The **mostDangerous** and **mostExpensive** `tbl_df` dataframe tables are created and sorted by the respective Total Sums from largest to smallest. For each dataframe table, all rows with a respective Total Sum of zero (0) are removed since these events record no injuries/fatalities or no property/crop damage, respectively.

While both of these tables can describe which type of weather events: 1) are the Most Harmful, and 2) have the Greatest Economic consequences, there is **significant inconsistencies** in how the Weather Event Types are recorded in the original data. These Event Types must be combined, collapsed and filtered for this analysis – See the **DATA FILTERING** section below.

```
## Group By Event Type
evtypeGroup <- group_by(actyData,EVTYPE)

## Select only Event Type, Fatalties, and Injuries columns
groupEvents <- select(evtypeGroup, EVTYPE, FATALITIES, INJURIES, PROPDMG, PROPDMGEXP,
                      CROPDMG, CROPDMGEXP)

## Apply B / M / K weights to Crop and Property Damage:
groupEvents[is.na(groupEvents)] <- 0  ## Adds 0s just in case NAs
groupEvents <- groupEvents %>% mutate(propDamage = ifelse(PROPDMGEXP == "B",
                1000000000*PROPDMG, ifelse(PROPDMGEXP == "M", 1000000*PROPDMG,
                ifelse(PROPDMGEXP == "K", 1000*PROPDMG, 0))))

groupEvents <- groupEvents %>% mutate(cropDamage = ifelse(CROPDMGEXP == "B",
                1000000000*CROPDMG, ifelse(CROPDMGEXP == "M", 1000000*CROPDMG,
                ifelse(CROPDMGEXP == "K", 1000*CROPDMG, 0))))

## GROUP EVENTS BY EVENT TYPE
groupEvents <- groupEvents %>% group_by(EVTYPE)

## Summarise by Grouped Events sum of: Fatalities, Injuries, Property and Crop Damage
sumGroupEvents <- groupEvents %>% summarise(sumFatal = sum(FATALITIES),
        sumInjury = sum(INJURIES),sumProp = sum(propDamage), sumCrop = sum(cropDamage))

## Add a column = sum Total of the sums of Fatalities, Injuries, Prop., and Crop Damage
sumGroupEvents <- sumGroupEvents %>% mutate(sumTotal = sumFatal + sumInjury)
sumGroupEvents <- sumGroupEvents %>% mutate(totalExpenses = sumProp + sumCrop)

## Filter out rows that have 0 in either of the total sum types
sumGroupEvents <- filter(sumGroupEvents, ((sumTotal >0) | (totalExpenses >0)))

## Most Dangerous/harmful to human health (and filter out 0's)
mostDangerous <- filter(sumGroupEvents, sumTotal > 0)
mostDangerous <- arrange(mostDangerous, desc(sumTotal))

## Greatest economic impact (and filter out 0's)
mostExpensive <- filter(sumGroupEvents, totalExpenses > 0)
mostExpensive <- arrange(mostExpensive, desc(totalExpenses))
```

Special Function called in Data Filtering to collapse two similar Activity Type Events

This helper `function` is called in the **DATA FILTERING** section to combine **two** Weather Event Types that are similar in nature, but have been recorded with very different names, into one observation (e.g.: "COLD" "LOW TEMPERATURE").

A `tbl_df` and two `strings` are passed to the function and the function returns a `tbl_df` with the two Event Type rows combined as one row.

The function works by filtering the `tbl_df` by the logical or `|` of the `grepl` of both Event Type names that are passed as `strings`. The filtered `tbl_df` is then mutated using the **first string** name that was passed to

the `function` as the new combined row name with the sum of the variable columns – combined by calling `summarise_all` with `funs(sum)` as the argument.

The old rows of the passed `tbl_df` are removed and the new row is added using `rbind`. The new `tbl_df` is returned.

```
collapseEventFunctionTwoNames <- function(groupedEvents, eventName1, eventName2) {

        processEvent <- groupedEvents %>% filter((grepl(eventName1, EVTYPE,
                ignore.case = TRUE))| (grepl(eventName2, EVTYPE, ignore.case = TRUE))) %>%
                mutate(EVTYPE = eventName1) %>% group_by(EVTYPE) %>%
                summarise_all(funs(sum))

        groupedEvents <- groupedEvents %>% filter(!(grepl(eventName1, EVTYPE,
                ignore.case = TRUE)) & !(grepl(eventName2, EVTYPE, ignore.case = TRUE)))

        groupedEvents <- rbind(groupedEvents, processEvent)

groupedEvents
}
```

Special Function called in Data Filtering to collapse multipe similar Event Function Names

This helper `function` is called in the **DATA FILTERING** section to combine **multiple** Weather Event Types that have very similar names into one observation (e.g.: "HURRICANE", "HURRICANE ANDREW", "hurricane").

A `tbl_df` and a `vector` of `strings` are passed to the function and the function returns a `tbl_df` with all the Event Type rows combined as one row for each `string` name in the passed `vector`.

The function works using a `for` loop over the passed `vector`of names (i.e.: for each name), filtering the `tbl_df` by the logical `agrepl` of the name in the `string vector`. The logically filtered `tbl_df` is then mutated using the `string` name and the data from all the variable columns combined by calling `summarise_all` with `funs(sum)` as the argument.

The old rows of passed `tbl_df` are removed by the logical opposite of filter `!(agrepl)` and the new row is added using `rbind` – this is done for each name provided in the `string vector`. The new `tbl_df` is returned.

```
collapseEventFunction <- function(groupedEvents, eventNames) {

    for (i in 1:length(eventNames)) {
        processEvent <- groupedEvents %>% filter(agrepl(eventNames[i], EVTYPE,
                ignore.case = TRUE)) %>% mutate(EVTYPE = eventNames[i]) %>%
                group_by(EVTYPE) %>% summarise_all(funs(sum))

        groupedEvents <- groupedEvents %>% filter(!(agrepl(eventNames[i], EVTYPE,
                ignore.case = TRUE)))

        groupedEvents <- rbind(groupedEvents, processEvent)
    }
groupedEvents
}
```

DATA FILTERING

This step collapses the data by combining **multiple** Weather Type Event data (rows) by combining events with very similar names (e.g.: "HURRICANE", "HURRICANE ANDREW", "hurricane") or combining **two** rows of data with similar weather type but have very different names (e.g.: "COLD" "LOW TEMPERATURE").

This data filtering is done by calls to the respective helper `function` described above. The names of the Weather Event Types that are used were created by analysis of the first filtering of the **mostDangerous** and **mostExpensive** datasets.

In some cases, weather type events were combined by considering data from more recent year as more complete. Other events that had questionable or inconsistent event names were combined with the event type named "Other." Other Event Name Types could be considered for more combinations or greater separation, as desired for the analysis; the filtering process is **completely automated and reproducable**.

```
## Collapse Event names for Fatalities and Injuries

## Vectors of strings with similar event names
similarEvents <- c("HURRICANE", "TORNADO", "HAIL", "RAIN", "FLOOD", "COLD", "DUST", "GLAZE",
        "GUSTY WIND", "HEAT WAVE", "SNOW", "HIGH SURF", "HYPOTHERMIA", "ICE", "LANDSLIDE",
        "HIGH WIND", "RIP CURRENT", "STORM SURGE", "THUNDERSTORM WIND", "FIRE",
        "WINTER STORM", "AVALANCE", "COASTAL STORM", "HEAT", "HEAVY RAIN", "THUNDERSTORM",
        "LIGHTNING", "WIND", "UNSEASONABLY WARM", "FROST", "OTHER")

similarEvents2 <- c("SURF", "SEAS", "WINTER WEATHER","TROPICAL STORM", "WINTER", "URBAN SMALL", "BLIZZA

## Apply filtering based on first two vectors of similar event names
newcollapse <- collapseEventFunction(mostDangerous, similarEvents)
newcollapse2 <- collapseEventFunction(newcollapse, similarEvents2)

## Apply filtering based on two names only
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse2, "FLOOD", "FLD")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HURRICANE", "TYPHOON")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HIGH WAVES", "HIGH SWELLS")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "FLOOD", "HIGH WATER")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "FLOOD", "RAPIDLY RISING WATER")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HIGH SEAS", "SEAS")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HIGH SURF", "SURF")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "COLD", "LOW TEMPERATURE")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "ICE", "GLAZE")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "OTHER", "HIGH")

## Arrange most dangerous events by sum of Injuries and Fatalities
mostDangerous <- arrange(newcollapse3, desc(sumTotal))
mostDangerous <- filter(mostDangerous, sumTotal > 0)  ## Remove all sums < 0

## Apply same filtering to Most Expensive:
## Collapse Event Names for Property and Crop Damage
## Apply same Event name filtering
newcollapse <- collapseEventFunction(mostExpensive, similarEvents)
newcollapse2 <- collapseEventFunction(newcollapse, similarEvents2)
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse2, "FLOOD", "FLD")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HURRICANE", "TYPHOON")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HIGH WAVES", "HIGH SWELLS")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "FLOOD", "HIGH WATER")
```

```
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "FLOOD", "RAPIDLY RISING WATER")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HIGH SEAS", "SEAS")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HIGH SURF", "SURF")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "COLD", "LOW TEMPERATURE")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "ICE", "GLAZE")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "TORNADO", "TORNDAO")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HIGH SURF", "HIGH SEAS")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "HIGH SURF", "HIGH WAVES")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "THUNDERSTORM", "TSTMW")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "COASTAL STORM", "COASTAL EROSION")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "COASTAL STORM", "BEACH EROSION")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "URBAN SMALL", "URBAN AND SMALL")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "FLOOD", "URBAN SMALL")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "RAIN", "DOWNBURST")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "OTHER", "LANDSLUMP")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "OTHER", "LANDSPOUT")
newcollapse3 <- collapseEventFunctionTwoNames(newcollapse3, "COLD", "COOL AND WET")

## Remove unknown/poorly identified weather event types:
newcollapse3 <- filter(newcollapse3, EVTYPE != "?" & EVTYPE != "APACHE COUNTY")

## Arrange most dangerous events by sum of Property and Crop Damage
mostExpensive <- arrange(newcollapse3, desc(totalExpenses))
mostExpensive <- filter(mostExpensive, totalExpenses > 0) ## Remove all sums < 0
```

RESULTS

Results are provided graphically by depicting the Event Types with the Most Dangerous and Most Expensive (Greatest Economic consequences) in respective graphs. The data is displayed in 3-dimensions: - First, on the Y-axis by Total = Injuries+Fatalities, or Total = Property+Crop Damage. - Color and Size represent the 2nd and 3rd dimensions of the data and display the sub-components of the Total Sums. This is interesting to see which events have a predominately greater number of one component-type than the other. For ease of display, all values are graphed as the log() of the associated data.

Most Dangerous Events

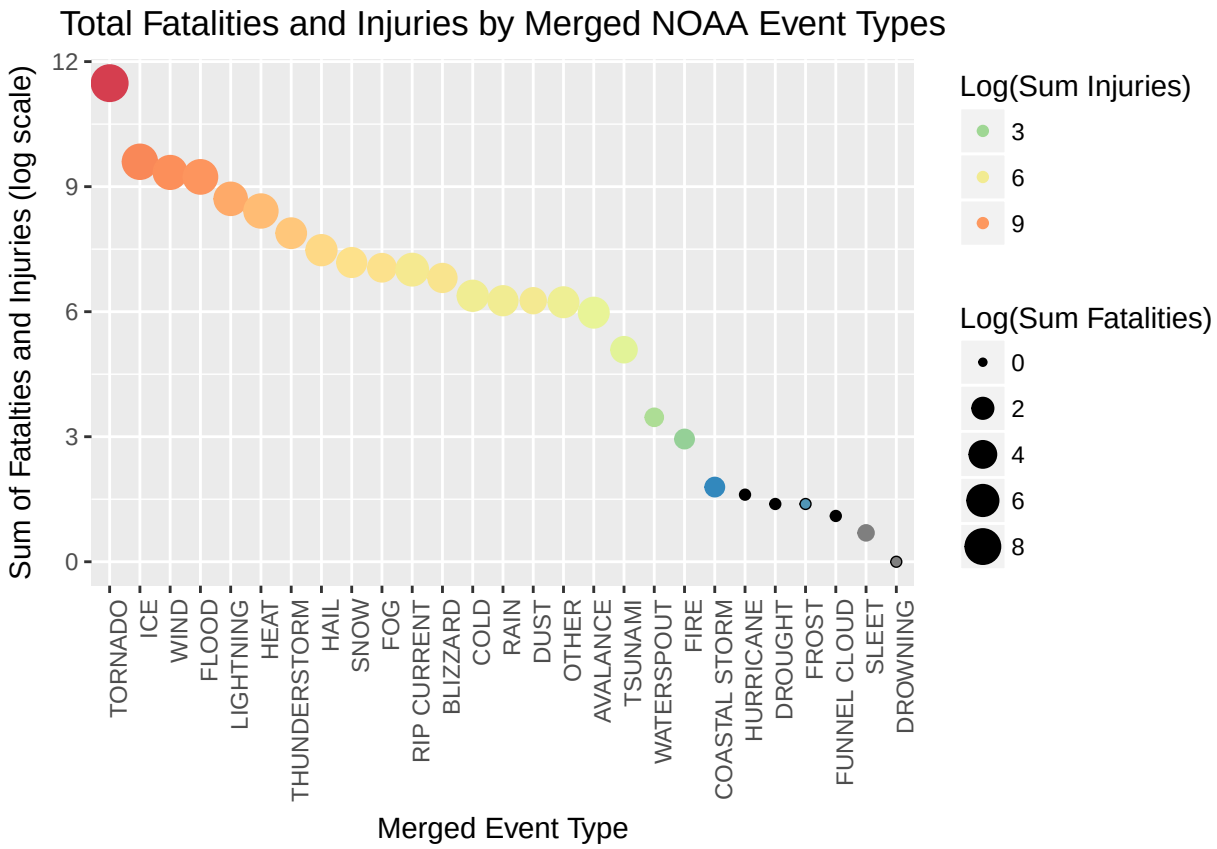

Figure 1: Events Most Harmful to Population Health

Greatest Economic Consequneces (Most Expensive)

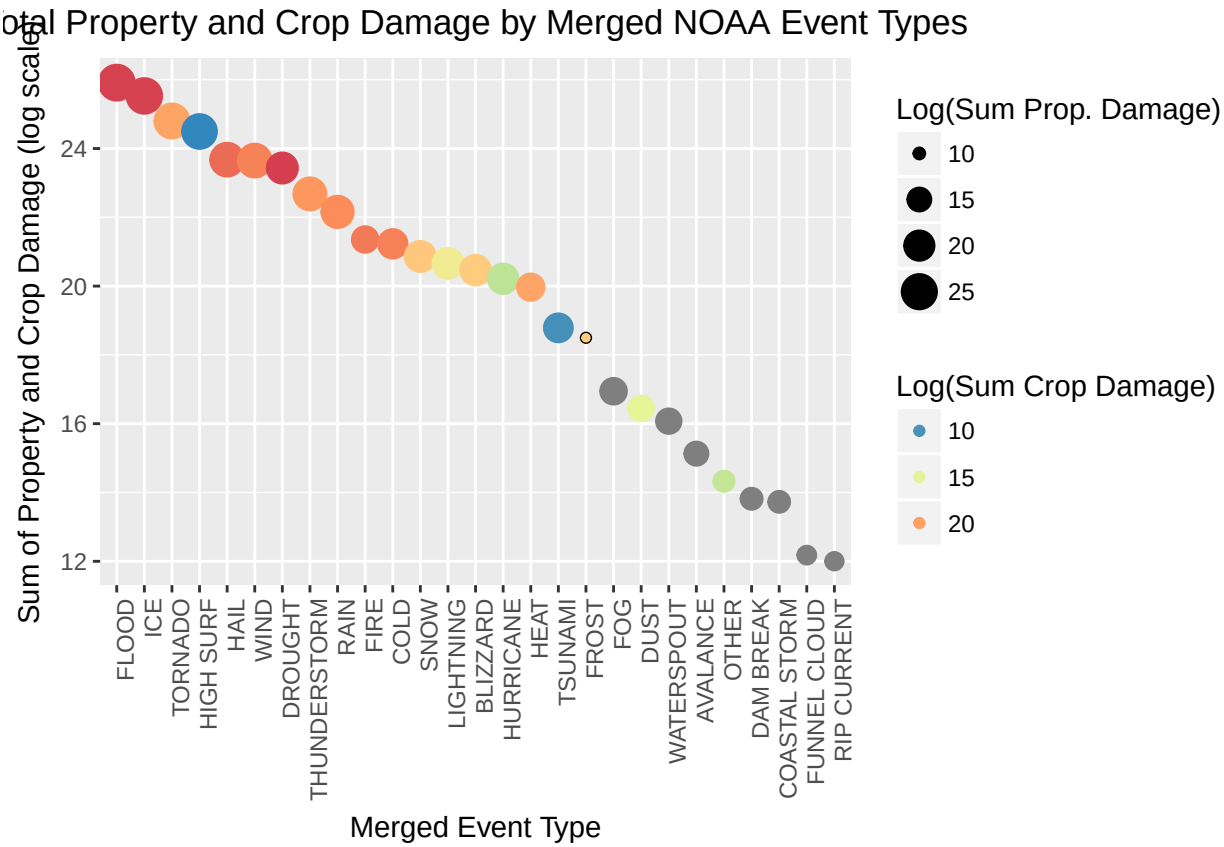

Figure 2: Events with the Greatest Economic Impact