Storm

Gorokhov Sergey

Friday, August 21, 2015

## Synopsys

The goal of this report is to analyse the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database and find which kind of events can case most harmful effect on health and economics.

# Data Processing

if(!"repdata-data-StormData.csv.bz2" %in% dir("./")){  
 download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2", destfile = "repdata-data-StormData.csv.bz2")  
}  
if(!"repdata-data-StormData.csv" %in% dir("./")){  
 unzip("repdata-data-StormData.csv.bz2")  
}

In this analyse we need only several columns in our data set. There are EVTYPE, FATALITIES, INJURIES, PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP. We can remove other columns.

if(!"ds" %in% ls() | !"dsm" %in% ls()){  
 ds<-read.csv("repdata-data-StormData.csv")  
 dsm<-ds[, c("EVTYPE", "FATALITIES", "INJURIES", "PROPDMG", "PROPDMGEXP", "CROPDMG", "CROPDMGEXP")]  
 remove(ds)  
}

As we know, in PROPDMGEXP and CROPDMGEXP columns stored level of values. Let transform it in real numbers. At first look at all possible levels

levels(dsm$PROPDMGEXP)

## [1] "" "-" "?" "+" "0" "1" "2" "3" "4" "5" "6" "7" "8" "B" "h" "H" "K"  
## [18] "m" "M"

And now replace with proper numeric values

dsm$PROPEXP[dsm$PROPDMGEXP == "K"] <- 1000  
dsm$PROPEXP[dsm$PROPDMGEXP == "k"] <- 1000  
dsm$PROPEXP[dsm$PROPDMGEXP == "M"] <- 1e+06  
dsm$PROPEXP[dsm$PROPDMGEXP == ""] <- 1  
dsm$PROPEXP[dsm$PROPDMGEXP == "B"] <- 1e+09  
dsm$PROPEXP[dsm$PROPDMGEXP == "m"] <- 1e+06  
dsm$PROPEXP[dsm$PROPDMGEXP == "0"] <- 1  
dsm$PROPEXP[dsm$PROPDMGEXP == "5"] <- 1e+05  
dsm$PROPEXP[dsm$PROPDMGEXP == "6"] <- 1e+06  
dsm$PROPEXP[dsm$PROPDMGEXP == "4"] <- 10000  
dsm$PROPEXP[dsm$PROPDMGEXP == "2"] <- 100  
dsm$PROPEXP[dsm$PROPDMGEXP == "3"] <- 1000  
dsm$PROPEXP[dsm$PROPDMGEXP == "h"] <- 100  
dsm$PROPEXP[dsm$PROPDMGEXP == "7"] <- 1e+07  
dsm$PROPEXP[dsm$PROPDMGEXP == "H"] <- 100  
dsm$PROPEXP[dsm$PROPDMGEXP == "1"] <- 10  
dsm$PROPEXP[dsm$PROPDMGEXP == "8"] <- 1e+08   
# give 0 to invalid exponent data, so they not count in  
dsm$PROPEXP[dsm$PROPDMGEXP == "+"] <- 0  
dsm$PROPEXP[dsm$PROPDMGEXP == "-"] <- 0  
dsm$PROPEXP[dsm$PROPDMGEXP == "?"] <- 0

And then do the same things with CROPDMGEXP column

levels(dsm$CROPDMGEXP)

## [1] "" "?" "0" "2" "B" "k" "K" "m" "M"

dsm$CROPEXP[dsm$CROPDMGEXP == "M"] <- 1e+06  
dsm$CROPEXP[dsm$CROPDMGEXP == "K"] <- 1000  
dsm$CROPEXP[dsm$CROPDMGEXP == "m"] <- 1e+06  
dsm$CROPEXP[dsm$CROPDMGEXP == "B"] <- 1e+09  
dsm$CROPEXP[dsm$CROPDMGEXP == "0"] <- 1  
dsm$CROPEXP[dsm$CROPDMGEXP == "k"] <- 1000  
dsm$CROPEXP[dsm$CROPDMGEXP == "2"] <- 100  
dsm$CROPEXP[dsm$CROPDMGEXP == ""] <- 1   
# give 0 to invalid exponent data, so they not count in  
dsm$CROPEXP[dsm$CROPDMGEXP == "?"] <- 0

Then compute the property damage value and crop damage value

dsm$PROPDMGVAL <- dsm$PROPDMG \* dsm$PROPEXP  
 dsm$CROPDMGVAL <- dsm$CROPDMG \* dsm$CROPEXP

After lets look at EVTYPE column. There are a lot of similar values. For example: WINDS and WIND, FLASH FLOOD and FLOOD FLASH FLOOD etc. Just try to replace similar values to reduce levels in EVTYPE

library(stringi)  
initlevels<-nlevels(dsm$EVTYPE)  
dsm$EVTYPE<-stri\_trim\_both(dsm$EVTYPE)  
dsm$EVTYPE<-stri\_replace\_all(dsm$EVTYPE, regex = "[/|\\(|\\)|\\\\|&|\\.]", " ")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "-", " ")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "AND", "")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "SLIDES", "SLIDE")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FLOODS", "FLOOD")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNDERSTORMS", "THUNDERSTORM")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNDERESTORM", "THUNDERSTORM")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNDEERSTORM", "THUNDERSTORM")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUDERSTORM", "THUNDERSTORM")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNDERSTORMW", "THUNDERSTORM")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNDERTORM", "THUNDERSTORM")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNDERSTORMWIND", "THUNDERSTORM WIND")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNERSTORM", "THUNDERSTORM")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNDERSTROM", "THUNDERSTORM")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNDERSTORMINDS", "THUNDERSTORM WIND")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "WINDS", "WIND")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "WINS", "WIND")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FLOODING", "FLOOD")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, " "," ")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FLD", "FLOOD")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "THUNDERSTORM WINDS", "THUNDERSTORM WIND")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FLOOD FLASHFLOOD", "FLASHFLOOD")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FLASH FLOOD FLOOD", "FLASHFLOOD")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FLASH FLOOD", "FLASHFLOOD")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FLOOD FLASH FLOOD", "FLASHFLOOD")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FLOOD FLASHFLOOD", "FLASHFLOOD")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FLOOD FLASH", "FLASHFLOOD")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "RAINS", "RAIN")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "CURRENTS", "CURRENT")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "CURRENTS", "CURRENT")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "LIGNTNING", "LIGHTNING")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "LIGHTING", "LIGHTNING")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "MUDSLIDE", "MUD SLIDE")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "STORMS", "STORM")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "FIRES", "FIRE")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "WILDFIRE", "WILD FIRE")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "SML", "SMALL")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, "ICY", "ICE")  
dsm$EVTYPE<-stri\_replace\_all\_fixed(dsm$EVTYPE, " "," ")  
dsm$EVTYPE<-stri\_trim\_both(dsm$EVTYPE)  
dsm$EVTYPE<-as.factor(dsm$EVTYPE)  
finlevels<-nlevels(dsm$EVTYPE)

So we have reduced the number of levels from 985 to 823

Then we can summarize damage and fatalities by event type

prdmg<-aggregate(PROPDMGVAL ~ EVTYPE, data = dsm, FUN = sum)  
 cropdmg<-aggregate(CROPDMGVAL ~ EVTYPE, data = dsm, FUN = sum)  
 fat<-aggregate(FATALITIES ~ EVTYPE, data = dsm, FUN = sum)  
 inj<-aggregate(INJURIES ~ EVTYPE, data = dsm, FUN = sum)

Since we need to find the maximum value of the damage, we can remove all the zero values

prdmg<-subset(prdmg, PROPDMGVAL > 0)  
 cropdmg<-subset(cropdmg, CROPDMGVAL > 0)  
 fat<-subset(fat, FATALITIES > 0)  
 inj<-subset(inj, INJURIES > 0)

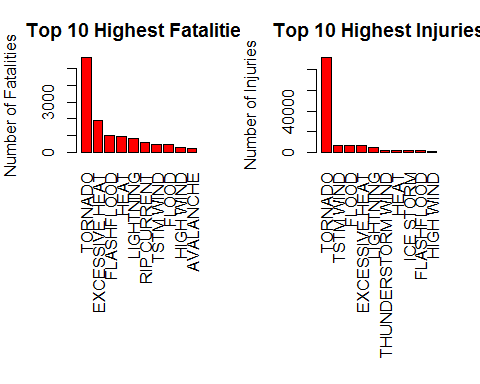
Sort our results and take top 10 values

fat<-fat[order(-fat$FATALITIES),]  
 inj<-inj[order(-inj$INJURIES),]  
 prdmg<-prdmg[order(-prdmg$PROPDMGVAL),]  
 cropdmg<-cropdmg[order(-cropdmg$CROPDMGVAL),]  
 topfat<-fat[1:10,]  
 topinj<-inj[1:10,]  
 topprdmg<-prdmg[1:10,]  
 topcropdmg<-cropdmg[1:10,]

# Results

### Across the United States, which types of events are most harmful with respect to population health?

par(mfrow = c(1, 2), mar = c(12, 4, 3, 2))  
barplot(topfat$FATALITIES, las = 3, names.arg = topfat$EVTYPE, main = "Top 10 Highest Fatalities", ylab = "Number of Fatalities", col = "red")  
barplot(topinj$INJURIES, las = 3, names.arg = topinj$EVTYPE, main = "Top 10 Highest Injuries", ylab = "Number of Injuries", col = "red")



### Across the United States, which types of events have the greatest economic consequences?

par(mfrow = c(1, 2), mar = c(12, 4, 3, 2))  
barplot(topprdmg$PROPDMGVAL/(10^9), las = 3, names.arg = topprdmg$EVTYPE, main = "Top 10 Property Damages", ylab = "Cost of damages ($ billions)", col = "red")  
barplot(topcropdmg$CROPDMGVAL/(10^9), las = 3, names.arg = topcropdmg$EVTYPE, main = "Top 10 Crop Damages", ylab = "Cost of damages ($ billions)", col = "red")

