Tornado and Flood are the most harmful storm events across the United States between 1996 and 2011

Sunday, June 21, 2015

Synopsis

In this report my aim to answer some basic questions about severe weather events across the United States between the years 1996 and 2011. My overall hypothesis is that the most harmful event with respect to population health is the Tornado and the event which has the greatest economic consequencies is the Flood. To investigate this hypothesis, I obtained the data from the Reproducible Research Coursera course website: Storm Data (47Mb) (https://d396qusza40orc.cloudfront.net /repdata%2Fdata%2FStormData.csv.bz2), the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database. From the Storm Data Documentation (https://d396qusza40orc.cloudfront.net /repdata%2Fpeer2_doc%2Fpd01016005curr.pdf) learned that the database contains only few types of events recorded before the year 1996, thus this analysis using data from 1996. During my analysis it shows that Tornadoes are the most harmful to population health and has the greatest economic impact to properties.

Data Processing

Loading and Processing the Raw Data

From the Reproducible Research Coursera course website (https://class.coursera.org/repdata-015/human_grading/view/courses/973516/assessments/4/submissions) I obtained the data on the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database, prepared for this assibnment: Storm Data (47Mb) (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2).

The file was downloaded to my working directory and unzipped, then read the raw csv file into R. During the processing:

1. required R packages were loaded,
2. only required columns were selected,
3. year data was included based on the BGN_DATE information,
4. data before 1996 were removed,
5. event types were cleared to only consider the 48 defined types,
6. expenses were counted based on PROPDMG, PROPDMGEXP, CROPDMG and CROPDMGEXP columns.

Reading the data set into R, using read.csv() functions:

```
stormRawData <- read.csv("repdata_data_StormData.csv")
```

Load the required R packages for data processing:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## install.packages("lubridate", repos = "http://cran.us.r-project.org")
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following object is masked _by_ '.GlobalEnv':
##
##     second
```

```
library(lattice)
library(stringr)
library(reshape2)
```

Subsetting the data and including the year information to the right format for analysis.

```
stormDataSelection <- select(stormRawData, REFNUM, STATE, BGN_DATE, EVTYPE, FATALI
TIES, INJURIES, PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP, REMARKS)

stormDataSelection$year <- year(as.Date(stormDataSelection$BGN_DATE, "%m/%d/%Y"))

## Removing the raw data set to free up the memory:

rm(stormRawData)
```

From 1950 through 1954, only tornado events were recorded.From 1955 through 1992, only tornado, thunderstorm wind and hail events were keyed from the paper publications into digital data. From 1993 to 1995, only tornado, thunderstorm wind and hail events have been extracted from the Unformatted Text Files. From 1996 to present, 48 event types are recorded as defined in NWS Directive 10-1605.)

Remove observations before 1996, as the 48 event types only collected from January 1996:

```
stormDataSelection <- subset(stormDataSelection, stormDataSelection$year >= 1996)
```

Cleaning the event type data:

```
## Create a function to change first character to uppercase

firstCap <- function(x) {
         s <- strsplit(x, " ")[[1]]
         paste(toupper(substring(s, 1, 1)), substring(s, 2),
               sep = "", collapse = " ")
     }

stormDataSelection$event.type <- as.factor(str_trim(stormDataSelection$EVTYPE))

event.type.levels <- levels(stormDataSelection$event.type)

event.type.levels <- tolower(event.type.levels)
for(i in 1:length(event.type.levels)){
        event.type.levels[i] <- firstCap(event.type.levels[i])
}

## Correct the event types to the defined categories, unless use 'Other'

event.type.levels[grep("Blizzard Summary", event.type.levels)] <- "Blizzard"
event.type.levels[grep("Coastal|[Cc]stl", event.type.levels)] <- "Coastal Flood"
event.type.levels[grep("Bitter Wind Chill|Bitter Wind Chill Temperatures|Cold|Cold
 And Frost|Cold And Snow|Cold Temperature|Cold Temperatures|Cold Weather|Cold Wind
 Chill Temperatures|Cold/wind Chill|Extended Cold|First Frost|Ice Jam Flood|Prolon
g Cold|Unseasonably Cool|Unseasonal Low Temp|Wind Chill", event.type.levels)] <- "
Cold/Wind Chill"
event.type.levels[grep("Patchy Dense Fog", event.type.levels)] <- "Dense Fog"
event.type.levels[grep("Smoke", event.type.levels)] <- "Dense Smoke"
event.type.levels[grep("Dust Devel", event.type.levels)] <- "Dust Devil"
event.type.levels[grep("Blowing Dust|Saharan Dust", event.type.levels)] <- "Dust S
torm"
event.type.levels[grep("Abnormal Warmth|Excessive Heat/drought|Excessively Dry|Hea
tburst|Mild And Dry Pattern|Prolong Warmth|Record Dry Month|Record Dryness|Record
Heat|Record High|Record Temperature|Record Temperatures|Record Warm|Record Warm Te
mps|Record Warmth|Temperature Record|Unseasonably Hot|Unseasonably Warm|Unseasonab
ly Warm Year|Unusual Warmth|Unusual/record Warmth|Very Dry|Very Warm|Unusually War
m", event.type.levels)] <- "Excessive Heat"
event.type.levels[grep("Excessive Cold|Extreme Cold|Extreme Cold/wind Chill|Extrem
e Wind Chill|Extreme Windchill|Extreme Windchill Temperatures|Record  Cold|Record
Cold|Record Cool|Unseasonable Cold|Unseasonably Cold|Unusually Cold", event.type.l
evels)] <- "Extreme Cold/Wind Chill"
event.type.levels[grep("Flash Flood|Flash Flood/flood|Flash Flooding|Flood/flash F
lood|Flood/flash/flood|Flood/strong Wind|Minor Flooding|River Flood|River Flooding
|Sml Stream Fld|Snowmelt Flooding|Street Flooding|Tidal Flooding|Urban Flood|Urban
 Flooding|Urban/street Flooding", event.type.levels)] <- "Flood"
event.type.levels[grep("Fog|Ice Fog", event.type.levels)] <- "Freezing Fog"
event.type.levels[grep("Agricultural Freeze|Damaging Freeze|Early Frost|Freeze|Fre
ezing Drizzle|Freezing Fog|Freezing Spray|Frost|Frost/freeze|Hard Freeze|Ice|Ice O
n Road|Ice Pellets|Ice Roads|Icy Roads|Late Freeze|Patchy Ice", event.type.levels)
] <- "Frost/Freeze"
event.type.levels[grep("Funnel Clouds|Wall Cloud", event.type.levels)] <- "Funnel
Cloud"
event.type.levels[grep("Hail(0.75)|Late Season Hail|Small Hail", event.type.levels
)] <- "Hail"
```

```
event.type.levels[grep("Abnormally Dry|Driest Month|Dry|Dry Conditions|Dry Microbu
rst|Dry Spell|Dry Weather|Dryness|Heat Wave|Hot And Dry|Hot Spell|Hot Weather|Mont
hly Temperature|Unseasonably Dry|Unseasonably Warm And Dry|Warm Weather", event.ty
pe.levels)] <- "Heat"
event.type.levels[grep("Abnormally Wet|Early Rain|Excessive Rain|Excessive Rainfal
l|Extremely Wet|Freezing Rain|Freezing Rain/sleet|Gusty Wind/hvy Rain|Gusty Wind/r
ain|Heavy Rain And Wind|Heavy Rain Effects|Heavy Rain/high Surf|Heavy Rain/wind|He
avy Rainfall|Light Freezing Rain|Locally Heavy Rain|Monthly Rainfall|Prolonged Rai
n|Rain|Rain (heavy)|Rain Damage|Record Low Rainfall|Record Rainfall|Sleet/freezing
 Rain|Thundersnow Shower|Torrential Rainfall|Tstm Heavy Rain|Unseasonably Cool & W
et|Unseasonably Warm & Wet|Unseasonably Warm/wet|Unseasonably Wet|Unseasonal Rain|
Wet Micoburst|Wet Microburst|Wet Month|Wet Year", event.type.levels)] <- "Heavy Ra
in"
event.type.levels[grep("Accumulated Snowfall|Blowing Snow|Drifting Snow|Early Snow
fall|Excessive Snow|Falling Snow/ice|First Snow|Heavy Snow Shower|Heavy Snow Squal
ls|Ice/snow|Late Season Snow|Late Season Snowfall|Late Snow|Light Snow|Light Snow/
flurries|Light Snow/freezing Precip|Light Snowfall|Moderate Snow|Moderate Snowfall
|Monthly Snowfall|Mountain Snows|Rain/snow|Record May Snow|Record Snow|Record Snow
fall|Record Winter Snow|Seasonal Snowfall|Snow|Snow Accumulation|Snow Advisory|Sno
w And Ice|Snow And Sleet|Snow Drought|Snow Showers|Snow Squall|Snow Squalls|Snow/b
lowing Snow|Snow/freezing Rain|Snow/ice|Snow/sleet|Unusually Late Snow", event.typ
e.levels)] <- "Heavy Snow"
event.type.levels[grep("Hazardous Surf|Heavy Surf|Heavy Surf/high Surf|High Surf A
dvisories|High Surf Advisory|Rough Surf", event.type.levels)] <- "High Surf"
event.type.levels[grep("Hail/wind|High Wind (g40)|High Winds|Non Tstm Wind|Non-sev
ere Wind Damage|Non-tstm Wind|Wake Low Wind|Whirlwind|Wind|Wind Advisory|Wind And
Wave|Wind Damage|Wind Gusts|Winds|Wnd", event.type.levels)] <- "High Wind"
event.type.levels[grep("Hurricane|Hurricane Edouard|Hurricane/typhoon|Hyperthermia
/exposure|Hypothermia/exposure|Typhoon", event.type.levels)] <- "Hurricane/Typhoon
"
event.type.levels[grep("Black Ice|Ice Jam|Icestorm/blizzard", event.type.levels)]
<- "Ice Storm"
event.type.levels[grep("Lake Effect Snow|Lake-effect Snow|Late-season Snowfall", e
vent.type.levels)] <- "Lake-Effect Snow"
event.type.levels[grep("Gusty Lake Wind", event.type.levels)] <- "Marine High Wind
"
event.type.levels[grep("Heavy Surf And Wind", event.type.levels)] <- "Marine Stron
g Wind"
event.type.levels[grep("Marine Tstm Wind", event.type.levels)] <- "Marine Thunders
torm Wind"
event.type.levels[grep("Beach Erosion|Cool Spell|Dam Break|Downburst|Drowning|Glaz
e|Heavy Precipitation|Heavy Seas|High  Swells|High Seas|High Swells|Landslide|Land
slides|Landslump|Landspout|Marine Accident|Microburst|Mixed Precip|Mixed Precipita
tion|Monthly Precipitation|Mud Slide|Mudslide|Mudslide/landslide|Mudslides|No Seve
re Weather|Non Severe Hail|None|Northern Lights|Record Precipitation|Red Flag Crit
eria|Red Flag Fire Wx|Remnants Of Floyd|Rock Slide|Rogue Wave|Rough Seas|^Summary|
Unusually Warm|Urban/small Strm Fldg|Urban/sml Stream Fld|Urban/sml Stream Fldg|Vo
g", event.type.levels)] <- "Other"
event.type.levels[grep("Rip Currents", event.type.levels)] <- "Rip Current"
event.type.levels[grep("Sleet Storm", event.type.levels)] <- "Sleet"
event.type.levels[grep("Astronomical High Tide|Blow-out Tide|Blow-out Tides|Metro
Storm, May 26|Storm Surge|Storm Surge/tide", event.type.levels)] <- "Storm Tide"
event.type.levels[grep("Gradient Wind|Strong Wind Gust|Strong Winds", event.type.l
evels)] <- "Strong Wind"
```

```
event.type.levels[grep("Gusty Thunderstorm Wind|Gusty Thunderstorm Winds|Gusty Win
d|Gusty Wind/hail|Gusty Winds|Severe Thunderstorm|Severe Thunderstorms|Thunderstor
m|Thunderstorm Wind (g40)|Thunderstorms|Tstm Wind|Tstm Wind  (g45)|Tstm Wind (41)|
Tstm Wind (g35)|Tstm Wind (g40)|Tstm Wind (g45)|Tstm Wind 40|Tstm Wind 45|Tstm Win
d And Lightning|Tstm Wind G45|Tstm Wind/hail|Tstm Winds|^Tstm", event.type.levels)
] <- "Thunderstorm Wind"
event.type.levels[grep("Tornado Debris", event.type.levels)] <- "Tornado"
event.type.levels[grep("Volcanic Ash Plume|Volcanic Ashfall|Volcanic Eruption", ev
ent.type.levels)] <- "Volcanic Ash"
event.type.levels[grep("High Water|Waterspouts", event.type.levels)] <- "Waterspou
t"
event.type.levels[grep("Brush Fire|Wild/forest Fire", event.type.levels)] <- "Wild
fire"
event.type.levels[grep("Winter Mix|Winter Weather Mix|Winter Weather/mix|Wintery M
ix|Wintry Mix", event.type.levels)] <- "Winter Weather"

levels(stormDataSelection$event.type) <- event.type.levels
```

Count the damage expenses:

```
## Translate letters to value:

levels(stormDataSelection$PROPDMGEXP) <- sub("0", "1", levels(stormDataSelection$P
ROPDMGEXP))
levels(stormDataSelection$PROPDMGEXP) <- sub("[Kk]", "1000", levels(stormDataSelec
tion$PROPDMGEXP))
levels(stormDataSelection$PROPDMGEXP) <- sub("[Mm]", "1000000", levels(stormDataSe
lection$PROPDMGEXP))
levels(stormDataSelection$PROPDMGEXP) <- sub("[Bb]", "1000000000", levels(stormDat
aSelection$PROPDMGEXP))
levels(stormDataSelection$PROPDMGEXP) <- sub("[Hh]", "100", levels(stormDataSelect
ion$PROPDMGEXP))

levels(stormDataSelection$CROPDMGEXP) <- sub("0", "1", levels(stormDataSelection$C
ROPDMGEXP))
levels(stormDataSelection$CROPDMGEXP) <- sub("[Kk]", "1000", levels(stormDataSelec
tion$CROPDMGEXP))
levels(stormDataSelection$CROPDMGEXP) <- sub("[Mm]", "1000000", levels(stormDataSe
lection$CROPDMGEXP))
levels(stormDataSelection$CROPDMGEXP) <- sub("[Bb]", "1000000000", levels(stormDat
aSelection$CROPDMGEXP))
levels(stormDataSelection$CROPDMGEXP) <- sub("[Hh]", "100", levels(stormDataSelect
ion$CROPDMGEXP))

stormDataSelection$PROPDMGEXP <- as.numeric(paste(stormDataSelection$PROPDMGEXP))
stormDataSelection$CROPDMGEXP <- as.numeric(paste(stormDataSelection$CROPDMGEXP))

stormDataSelection$PROPDMGEXP[is.na(stormDataSelection$PROPDMGEXP)] <- 0
stormDataSelection$CROPDMGEXP[is.na(stormDataSelection$CROPDMGEXP)] <- 0
```

Finalise the tidy data set:

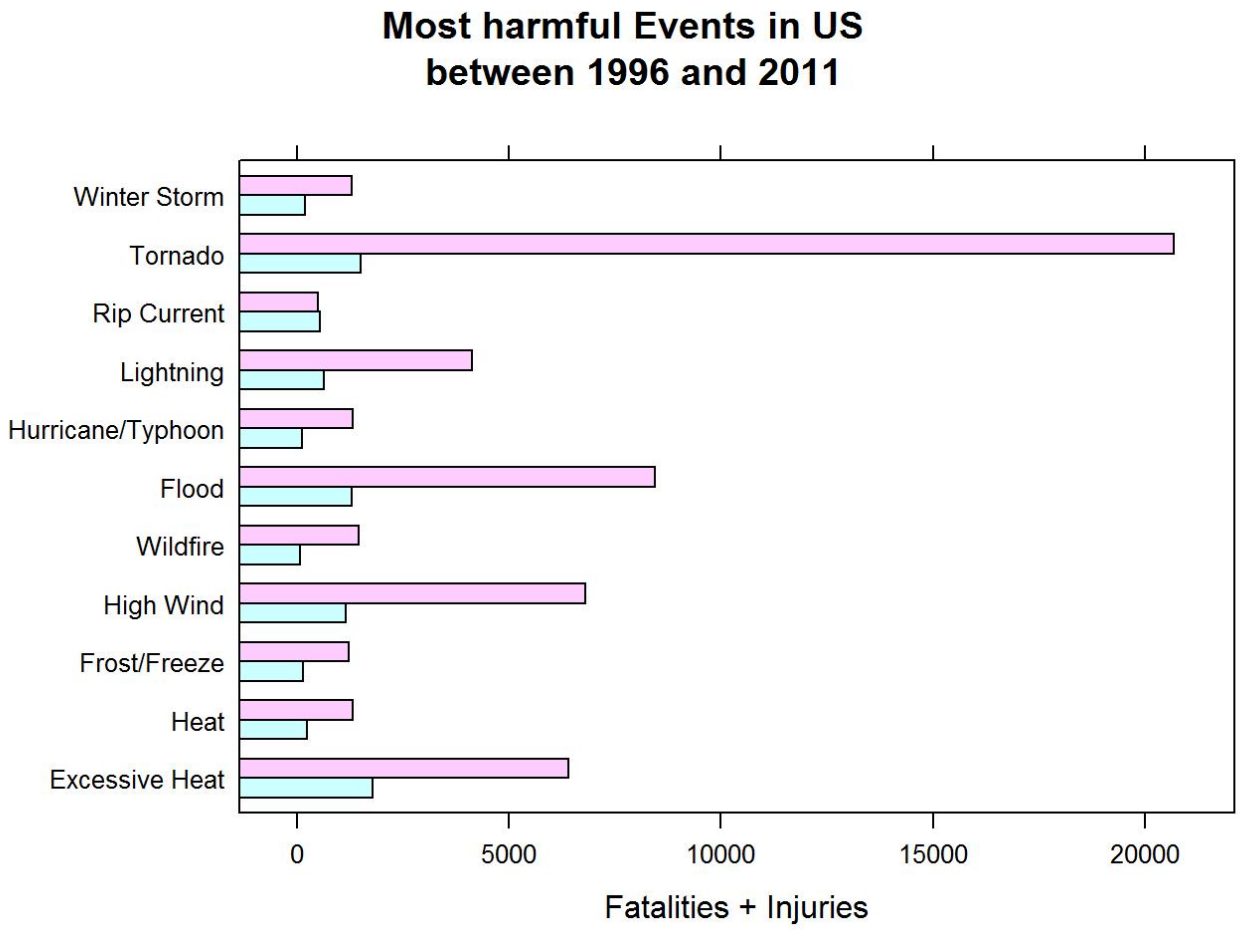
```
stormTidyData <- stormDataSelection %>%
        mutate(property.damage.expense = PROPDMGEXP * PROPDMG / 1000000, crop.dama
ge.expense = CROPDMGEXP * CROPDMG / 1000000) %>%
        select(state = STATE, year, event.type, fatalities = FATALITIES, injuries
= INJURIES, property.damage.expense, crop.damage.expense)

## Remove items without harms or damages

stormData <- subset(stormTidyData, (fatalities + injuries + property.damage.expens
e + crop.damage.expense) > 0)

## Create exploratory plots:

fatalitiesData <- as.data.frame(xtabs(fatalities ~ event.type, stormData))
injuriesData <- as.data.frame(xtabs(injuries ~ event.type, stormData))
harmfulData <- merge(fatalitiesData, injuriesData, by="event.type")
names(harmfulData)[2:3] <- c("Fatalities", "Injuries")
harmfulDataTop <- subset(arrange(mutate(harmfulData, Summ = Fatalities + Injuries)
, desc(Summ)), Summ > 1000)

p1 <- barchart(event.type ~ Fatalities + Injuries, data = harmfulDataTop, main = "
Most harmful Events in US \n between 1996 and 2011")


propertyData <- as.data.frame(xtabs(property.damage.expense ~ event.type, stormDat
a))
cropData <- as.data.frame(xtabs(crop.damage.expense ~ event.type, stormData))
expenseData <- merge(propertyData, cropData, by="event.type")
names(expenseData)[2:3] <- c("Property.Damage", "Crop.Damage")
expenseDataTop <- subset(arrange(mutate(expenseData, Total.Damage = Property.Damag
e + Crop.Damage), desc(Total.Damage)), Total.Damage > 1000)

p2 <- barchart(event.type ~ Property.Damage + Crop.Damage, data = expenseDataTop,
main = "The Greatest Economic Consequencies of Weather Events in US \n between 199
6 and 2011 (in million USD)")
```

The data processing steps are completed and the data set is ready for analysis.

Results

Events analysis across the United States causing the most harmful result

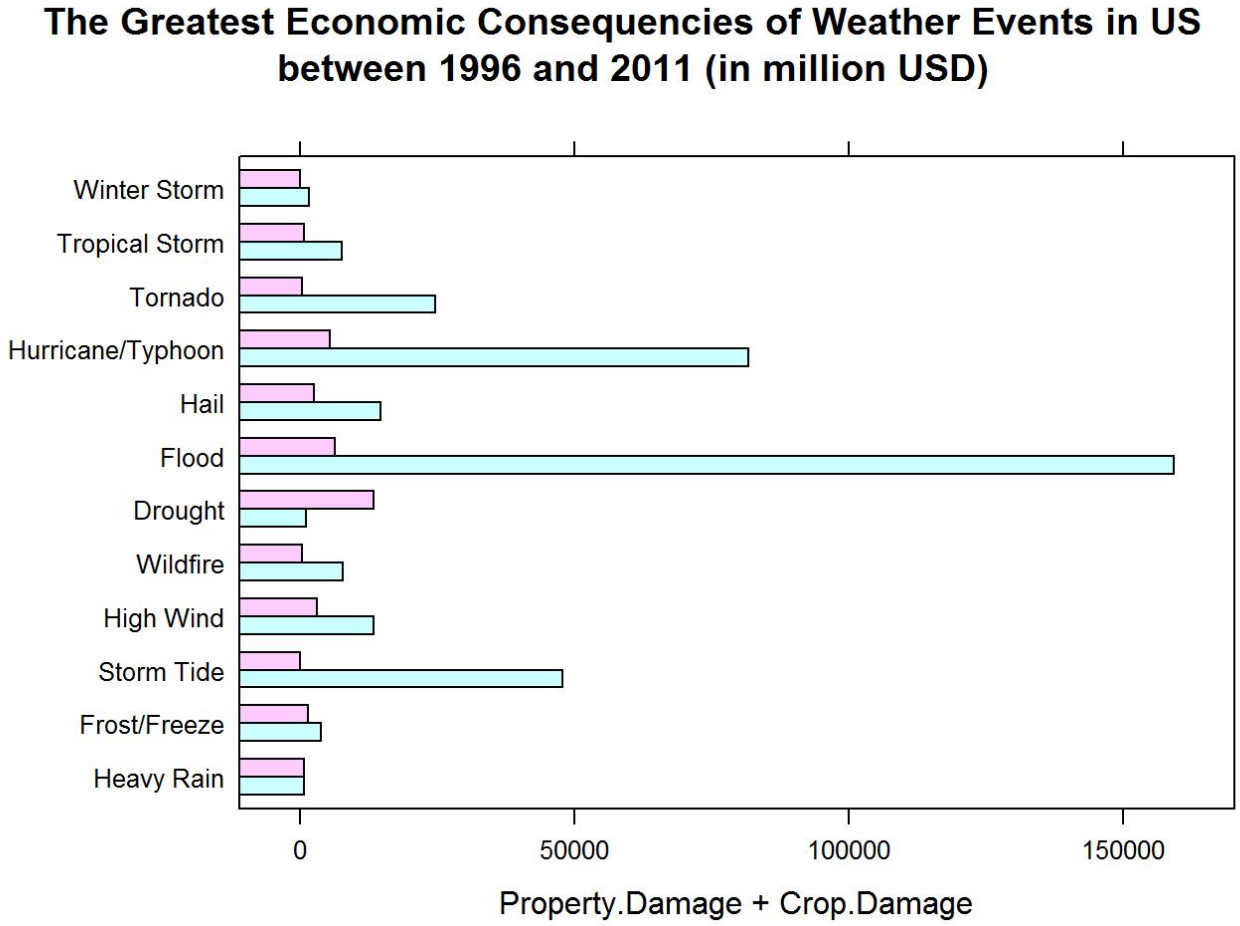
```
print(p1)
```



The Tornado is the most Harmful weather event across the United States between 1996 and 2011.

Events analysis across the United States causing the greatest damage

```
print(p2)
```



The Flood events have the greatest Economic Consequencies on properties and crops across the United States between 1996 and 2011.