Tornados and excessive heats are the most harmful to population health while floods and hurricanes have the greatest economic consequences in the US

Olga Larina

4/15/2019

Synopsis

The U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database tracks characteristics of major storms and weather events in the United States, including when and where they occur, as well as estimates of any fatalities, injuries, and property damage. This data analysis addresses 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?

I cleaned data to get real types of events (according to categories from documentation), analized and summarized all fatalities, injuries, and property damages that have been recorded since 1993, because earlier only few types of events have been recorded. As soon as I want to compare events, I have to to take years for which there is more information. Most harmful with respect to population health are tornados, excessive heats, floods, flash floods, lightnings and thunderstorm winds. Floods, hurricanes, tornados and hails have the greatest economic consequences.

Data Processing

I downloaded the data from the project task repository and filtered only variables that are important for the project. Also, convert dates to Data type.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(stringdist)
library(ggplot2)
if (!file.exists("./data"))
    dir.create("./data")
if (!file.exists("./data/repdata_data_StormData.csv.bz2"))
{
    fileURL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
    download.file(fileURL, "./data/repdata_data_StormData.csv.bz2",method = "curl")
}
```

```
data <- read.csv("./data/repdata_data_StormData.csv.bz2",na.strings = "")
subdata <- data[,c("BGN_DATE","EVTYPE","FATALITIES","INJURIES","PROPDMG", "PROPDMGEXP","CROPDMG","CROPDM
dates <- subdata$BGN_DATE
dates <- as.Date(as.character(dates),"%m/%d/%Y")
subdata$BGN_DATE <- dates
```

EVTYPE contains types of events.

```
length(unique(subdata$EVTYPE))
```

```
## [1] 985
```

Are there so many categories for all years? Let's see.

```
library(dplyr)
subdata2 <- filter(subdata, BGN_DATE < as.Date("01/01/1993","%m/%d/%Y"))
unique(subdata2$EVTYPE)
```

```
## [1] TORNADO   TSTM WIND HAIL
## 985 Levels:    HIGH SURF ADVISORY  COASTAL FLOOD ... WND
```

```
subdata3 <- filter(subdata, BGN_DATE < as.Date("01/01/1994","%m/%d/%Y"))
length(unique(subdata3$EVTYPE))
```

```
## [1] 160
```

So, for 1993 we have a lot of data. Let's take 1993 and futher.

```
subdata <- filter(subdata, BGN_DATE >= as.Date("01/01/1993","%m/%d/%Y"))
```

From the documentation I know that there have to be 48 categories, let's create variable for them.

```
evTypes <- c("Astronomical Low Tide", "Avalanche", "Blizzard", "Coastal Flood", "Cold/Wind Chill","Debri
evTypes <- tolower(evTypes)
```

So, I have a lot of typos in the data. I have to convert subdata$EVTYPE to this 48 categories. An easy step is to remove spaces and convert to lower case.

```
library(dplyr)
library(stringr)
subdata <- mutate(subdata, EVTYPE = tolower(EVTYPE))
subdata <- mutate(subdata, EVTYPE = str_trim(EVTYPE))
```

I noticed, that there are abbreviations of words "wind" and "thunderstorm", let's fix it.

```
subdata$EVTYPE<-gsub("tstm","thunderstorm",subdata$EVTYPE,fixed = TRUE)
subdata$EVTYPE<-gsub("wnd","wind",subdata$EVTYPE,fixed = TRUE)
```

The main idea of data cleaning is to take each category from evTypes and try to find it in subdata$EVTYPE.

```
grep("hurricane|typhoon",evTypes,value=T)
```

```
## [1] "hurricane (typhoon)"
```

```
unique(grep("hurricane|typhoon",subdata$EVTYPE,value=T))
```

```
##  [1] "hurricane opal/high winds"  "hurricane erin"
##  [3] "hurricane opal"             "hurricane"
##  [5] "hurricane-generated swells" "hurricane emily"
##  [7] "hurricane gordon"           "hurricane felix"
##  [9] "hurricane edouard"          "typhoon"
```

```
## [11] "hurricane/typhoon"
```

```
subdata$EVTYPE[grep("hurricane|typhoon",subdata$EVTYPE)] <- "hurricane (typhoon)"

grep("tide",evTypes,value=T)
```

```
## [1] "astronomical low tide" "storm surge/tide"
```

```
unique(grep("tide",subdata$EVTYPE,value=T))
```

```
## [1] "high wind and high tides" "high tides"
## [3] "blow-out tides"           "blow-out tide"
## [5] "astronomical high tide"   "storm surge/tide"
## [7] "astronomical low tide"
```

```
subdata$EVTYPE[grep("blow-out tides|blow-out tide",subdata$EVTYPE)] <- "astronomical low tide"
subdata$EVTYPE[grep("astronomical high tide|high wind and high tides|high tides|storm surge",subdata$EV

grep("flood",evTypes,value=T)
```

```
## [1] "coastal flood"   "flash flood"     "flood"           "lakeshore flood"
```

```
unique(grep("flood",subdata$EVTYPE,value=T))
```

```
##  [1] "ice storm/flash flood"          "flash flood"
##  [3] "flash flooding"                 "flooding"
##  [5] "flood"                          "flash flooding/thunderstorm wi"
##  [7] "breakup flooding"               "river flood"
##  [9] "coastal flood"                  "flood watch/"
## [11] "flash floods"                   "flooding/heavy rain"
## [13] "heavy surf coastal flooding"    "urban flooding"
## [15] "urban/small flooding"           "local flood"
## [17] "flood/flash flood"              "flood/rain/winds"
## [19] "flash flood winds"              "urban/small stream flooding"
## [21] "stream flooding"                "flash flood/"
## [23] "flood/rain/wind"                "small stream urban flood"
## [25] "urban flood"                    "heavy rain/flooding"
## [27] "coastal flooding"               "high winds/flooding"
## [29] "urban/small stream flood"       "minor flooding"
## [31] "urban/small stream  flood"      "urban and small stream flood"
## [33] "small stream flooding"          "floods"
## [35] "small stream and urban floodin" "small stream/urban flood"
## [37] "small stream and urban flood"   "rural flood"
## [39] "thunderstorm winds urban flood" "major flood"
## [41] "ice jam flooding"               "street flood"
## [43] "small stream flood"             "lake flood"
## [45] "urban and small stream floodin" "river and stream flood"
## [47] "minor flood"                    "high winds/coastal flood"
## [49] "river flooding"                 "flood/river flood"
## [51] "mud slides urban flooding"      "heavy snow/high winds & flood"
## [53] "hail flooding"                  "thunderstorm winds/flash flood"
## [55] "heavy rain and flood"           "local flash flood"
## [57] "flood/flash flooding"           "coastal/tidal flood"
## [59] "flash flood/flood"              "flash flood from ice jams"
## [61] "flash flood - heavy rain"       "flash flood/ street"
## [63] "flash flood/heavy rain"         "heavy rain; urban flood winds;"
## [65] "flood flash"                    "flood flood/flash"
```

```
## [67] "tidal flood"                    "flood/flash"
## [69] "heavy rains/flooding"           "thunderstorm winds/flooding"
## [71] "highway flooding"               "flash flood/ flood"
## [73] "heavy rain/mudslides/flood"     "beach erosion/coastal flood"
## [75] "snowmelt flooding"              "flash flooding/flood"
## [77] "beach flood"                    "thunderstorm winds/ flood"
## [79] "flood & heavy rain"             "flood/flashflood"
## [81] "urban small stream flood"       "urban flood landslide"
## [83] "urban floods"                   "heavy rain/urban flood"
## [85] "flash flood/landslide"          "landslide/urban flood"
## [87] "flash flood landslides"         "ice jam flood (minor"
## [89] "coastalflood"                   "erosion/cstl flood"
## [91] "tidal flooding"                 "street flooding"
## [93] "flood/strong wind"              "coastal flooding/erosion"
## [95] "urban/street flooding"          "coastal  flooding/erosion"
## [97] "flood/flash/flood"              "cstl flooding/erosion"
## [99] "lakeshore flood"
```

```
s<-subdata$EVTYPE
subdata$EVTYPE[grep("(flash.*flood(ing)?|flood(ing)?.*flash)",subdata$EVTYPE)] <- "flash flood"
subdata$EVTYPE[grep("(coastal.*flood(ing)?|flood(ing)?.*coastal|cstl.*flood(ing)?|flood(ing)?.*cstl)",su
subdata$EVTYPE[grep("(lake(shore)?.*flood(ing)?|flood(ing)?.*lake(shore)?)",subdata$EVTYPE)] <- "lakesho
floodPos <- grepl("flood",subdata$EVTYPE) & !grepl("flash",subdata$EVTYPE) &
        !grepl("coastal",subdata$EVTYPE) & !grepl("lakeshore",subdata$EVTYPE)
subdata$EVTYPE[floodPos] <- "flood"


unique(grep("extreme cold/wind chill",subdata$EVTYPE,value=T))
```

```
## [1] "extreme cold/wind chill"
```

```
subdata$EVTYPE[grep("cold",subdata$EVTYPE)] <- "extreme cold/wind chill"

subdata$EVTYPE[grep("avalanche",subdata$EVTYPE)] <- "avalanche"
subdata$EVTYPE[grep("blizzard",subdata$EVTYPE)] <- "blizzard"

subdata$EVTYPE[grep("dense fog|fog|patchy dense fog",subdata$EVTYPE)] <- "dense fog"
subdata$EVTYPE[grep("freezing fog|ice fog",subdata$EVTYPE)] <- "freezing fog"

subdata$EVTYPE[grep("dense smoke|smoke",subdata$EVTYPE)] <- "dense smoke"
subdata$EVTYPE[grep("drought",subdata$EVTYPE)] <- "drought"

subdata$EVTYPE[grep("dust",subdata$EVTYPE)] <- "dust storm"

subdata$EVTYPE[grep("heat",subdata$EVTYPE)] <- "excessive heat"

subdata$EVTYPE[grep("frost|freeze",subdata$EVTYPE)] <- "frost/freeze"

subdata$EVTYPE[grep("cloud",subdata$EVTYPE)] <- "funnel cloud"

hailPos <- grepl("hail",subdata$EVTYPE) & !grepl("marine",subdata$EVTYPE)

subdata$EVTYPE[hailPos] <- "hail"

subdata$EVTYPE[grep("rain|wet",subdata$EVTYPE)] <- "heavy rain"
```

```
lakeSnowPos <- grepl("(lake.*snow|snow.*lake)",subdata$EVTYPE)
subdata$EVTYPE[lakeSnowPos] <- "lake-effect snow"
subdata$EVTYPE[grepl("snow",subdata$EVTYPE) & ! lakeSnowPos] <- "heavy snow"

subdata$EVTYPE[grep("surf",subdata$EVTYPE)] <- "high surf"

## wind
nonColdMarine <- !grepl("extreme cold/wind chill",subdata$EVTYPE)&!grepl("marine",subdata$EVTYPE)
wind <- unique(grep("wind",subdata$EVTYPE,value=T))

subdata$EVTYPE[grepl("(thunderstorm.*wind|wind.*thunderstorm)",subdata$EVTYPE)&nonColdMarine] <- "thund
subdata$EVTYPE[grepl("(high.*wind|wind.*high)",subdata$EVTYPE)&nonColdMarine] <- "high wind"
subdata$EVTYPE[grepl("(strong.*wind|wind.*strong)",subdata$EVTYPE)&nonColdMarine] <- "strong wind"

subdata$EVTYPE[grepl("(chill.*wind|wind.*chill)",subdata$EVTYPE)] <- "extreme cold/wind chill"
unique(subdata$EVTYPE[grep("wind",subdata$EVTYPE)])
```

```
##  [1] "thunderstorm wind"         "extreme cold/wind chill"
##  [3] "high wind"                 "wind"
##  [5] "wind damage"               "gusty winds"
##  [7] "strong wind"               "winds"
##  [9] "downburst winds"           "dry microburst winds"
## [11] "dry mircoburst winds"      "microburst winds"
## [13] "gradient winds"            "thundertorm winds"
## [15] "wind storm"                "tunderstorm wind"
## [17] "thundertsorm wind"         "thundestorm winds"
## [19] "thunderstrom winds"        "lightning and winds"
## [21] "thuderstorm winds"         "storm force winds"
## [23] "thunderestorm winds"       "thundeerstorm winds"
## [25] "thunerstorm winds"         "thunderstrom wind"
## [27] "whirlwind"                 "gusty wind"
## [29] "gradient wind"             "wake low wind"
## [31] "wind advisory"             "wind and wave"
## [33] "non-severe wind damage"    "wind gusts"
## [35] "gusty lake wind"           "marine thunderstorm wind"
## [37] "marine high wind"          "marine strong wind"
```

```
subdata$EVTYPE<-gsub("winds","wind",subdata$EVTYPE,fixed = TRUE)
subdata$EVTYPE<-gsub("thundertorm|tunderstorm|thundertsorm|thundeerstorm|thuderstorm|thunerstorm|thundes

nonTH <- !grepl("thunderstorm",subdata$EVTYPE) & !grepl("high",subdata$EVTYPE)
subdata$EVTYPE[grepl("wind",subdata$EVTYPE) & nonColdMarine & nonTH] <- "strong wind"

subdata$EVTYPE[grep("ice",subdata$EVTYPE)] <- "ice storm"
subdata$EVTYPE[grep("lightning",subdata$EVTYPE)] <- "lightning"
subdata$EVTYPE[grep("rip current",subdata$EVTYPE)] <- "rip current"
subdata$EVTYPE[grep("sleet",subdata$EVTYPE)] <- "sleet"
s<-subdata$EVTYPE

subdata$EVTYPE[grep("tornado",subdata$EVTYPE)] <- "tornado"
subdata$EVTYPE[grep("tropical storm",subdata$EVTYPE)] <- "tropical storm"
subdata$EVTYPE[grep("volcanic",subdata$EVTYPE)] <- "volcanic ash"
subdata$EVTYPE[grep("wildfire",subdata$EVTYPE)] <- "wildfire"
```

```
subdata$EVTYPE[grep("winter storm",subdata$EVTYPE)] <- "winter storm"
subdata$EVTYPE[grep("winter weather|winter mix|wintery mix|winter weather mix|winter weather/mix",subdata
subdata$EVTYPE[grep("warm|hot|record high temperatures|record temperatures",subdata$EVTYPE)] <- "heat"
subdata$EVTYPE[grep("dry",subdata$EVTYPE)] <- "drought"
subdata$EVTYPE[grep("summary|none|\\?",subdata$EVTYPE)] <- "others"

subdata$EVTYPE[grepl("thunderstor",subdata$EVTYPE) & !grepl("marine",subdata$EVTYPE)] <- "thunderstorm
subdata$EVTYPE[grep("coastalstorm|coastal storm",subdata$EVTYPE)] <- "storm surge/tide"
subdata$EVTYPE[grep("cool",subdata$EVTYPE)] <- "extreme cold/wind chill"
```

So, now we have categories.

```
length(unique(subdata$EVTYPE))
```

```
## [1] 155
```

Let's try to parse it one more time and if it doesn't work - replace with empty string.

```
library(stringdist)
fixedType <- character(length(subdata$EVTYPE))
for(i in 1:length(subdata$EVTYPE))
{
    type <- subdata$EVTYPE[i]
    typeSplit <- strsplit(type,"/")
    for (j in 1:length(typeSplit[[1]]))
        num <- sum(which(amatch(evTypes,typeSplit[[1]][j],maxDist=2)==1),na.rm = TRUE)
        if (num > 0)
            fixedType[i] <- evTypes[num]
}
```

Results

It's time to answer the questions. Let's take a look on fatalities and injuries.

```
library(ggplot2)
populationHealth <- data.frame(Event.Type = fixedType, Fatalities = subdata$FATALITIES,
                               Injuries = subdata$INJURIES,stringsAsFactors = FALSE)
popH <- aggregate(.~Event.Type, data = populationHealth, sum)
popH$Event.Type[grep("^$",popH$Event.Type)] <- "others"

topF <- quantile(popH$Fatalities, probs = 0.9)
popHF <- filter(popH, popH$Fatalities> topF)
print("The most harmful events: Fatalities")
```

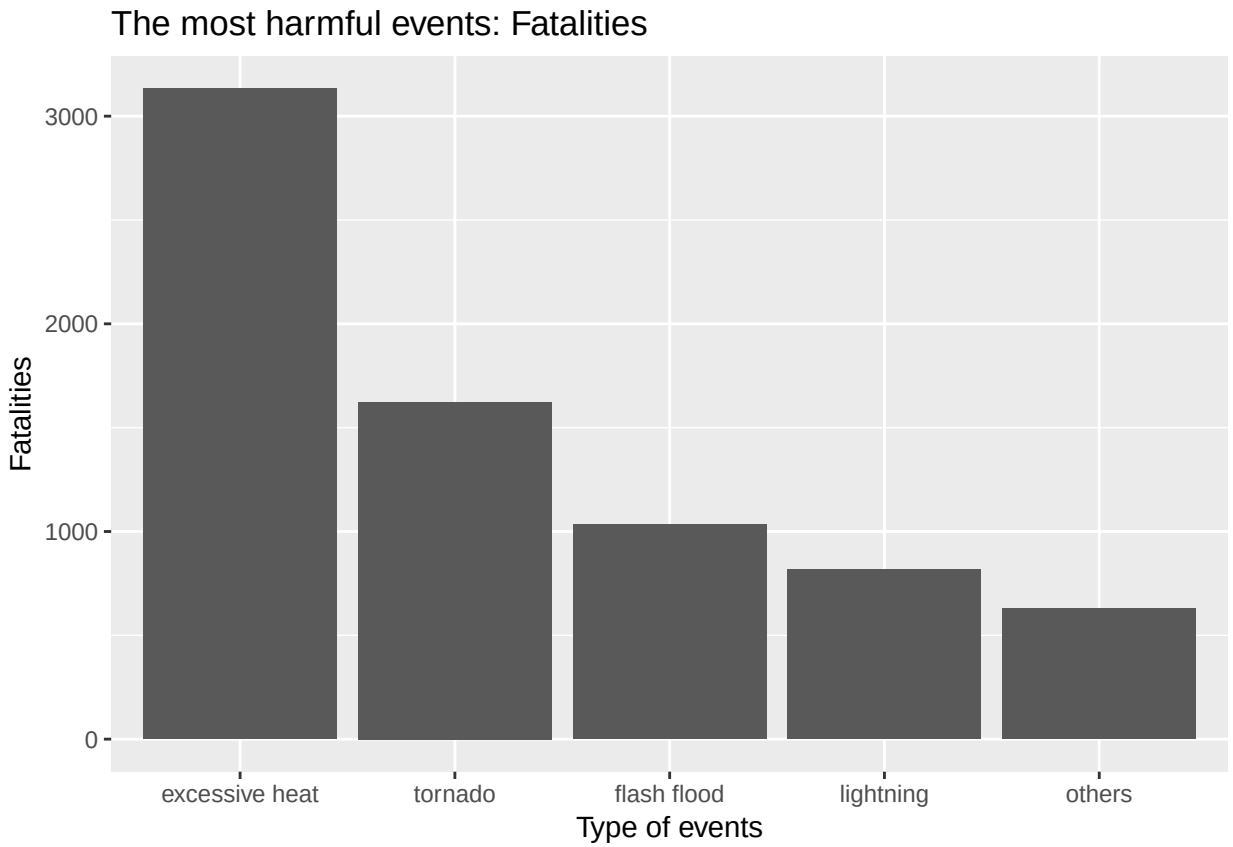
```
## [1] "The most harmful events: Fatalities"
```

```
popHF
```

```
##       Event.Type Fatalities Injuries
## 1         others        630     1441
## 2 excessive heat       3132     9209
## 3    flash flood       1035     1802
## 4      lightning        817     5232
## 5        tornado       1624    23371
```

```
gg<- ggplot(popHF,aes(x=reorder(Event.Type,-Fatalities),y = Fatalities)) +
    geom_col() +
```

```
    labs(x="Type of events", title = "The most harmful events: Fatalities")
print(gg)
```



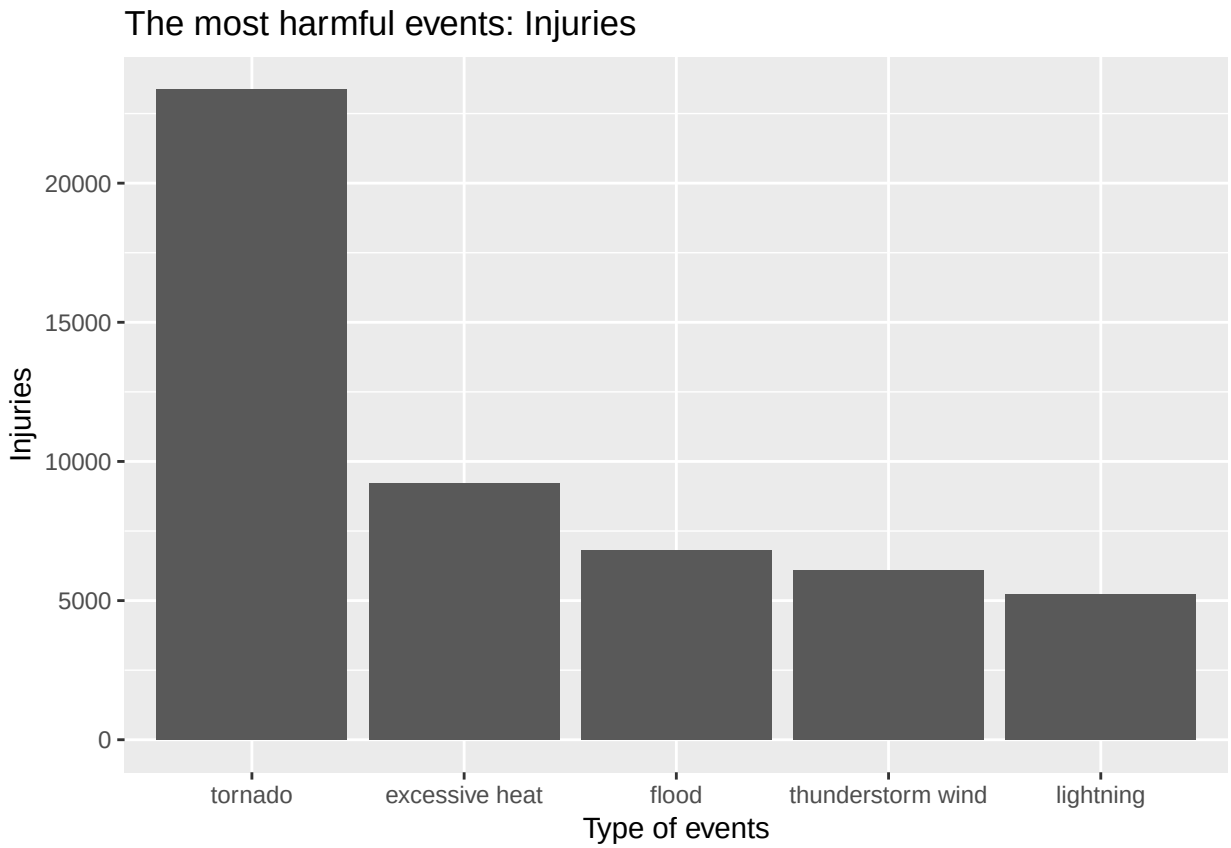

```
topI <- quantile(popH$Injuries, probs = 0.9)
popHI <- filter(popH, popH$Injuries> topI)
print("The most harmful events: Injuries")
```

```
## [1] "The most harmful events: Injuries"
```

```
popHI
```

```
##          Event.Type Fatalities Injuries
## 1    excessive heat       3132     9209
## 2             flood        484     6795
## 3         lightning        817     5232
## 4 thunderstorm wind        443     6087
## 5           tornado       1624    23371
```

```
gg2<- ggplot(popHI,aes(x=reorder(Event.Type,-Injuries),y = Injuries)) +
    geom_col() +
    labs(x="Type of events", title = "The most harmful events: Injuries")
print(gg2)
```



To calculate the cost of loss, I have to take a look on "PROPDMG", "PROPDMGEXP", "CROPDMG", "CROPDMGEXP". The "CROPDMGEXP" is the exponent values for "CROPDMG" (crop damage). In the sam way, "PROPDMG" and "PROPDMGEXP".

```
rnum <- length(subdata$PROPDMGEXP)
prop <- numeric(rnum)
propExp <- as.character(subdata$PROPDMGEXP)
propDmg <- subdata$PROPDMG
for (i in 1:rnum)
{
    exp <- propExp[i]
    pr <- propDmg[i]
    if (!is.na(exp))
    {
        if (exp=="0")
            prop[i] <- pr*10^0
        if (exp=="1")
            prop[i] <- pr*10^1
        if (exp=="2")
            prop[i] <- pr*10^2
        if (exp=="3")
            prop[i] <- pr*10^3
        if (exp=="4")
            prop[i] <- pr*10^4
        if (exp=="5")
            prop[i] <- pr*10^5
        if (exp=="6")
```

```
            prop[i] <- pr*10^6
        if (exp=="7")
            prop[i] <- pr*10^7
        if (exp=="8")
            prop[i] <- pr*10^8
        if (exp == "B")
            prop[i] <- pr*10^9
        if (exp == "h" | exp == "H")
            prop[i] <- pr*10^2
        if (exp == "K")
            prop[i] <- pr*10^3
        if (exp == "m" | exp == "M")
            prop[i] <- pr*10^6
    }
}
```

In the same way, I count crop.

Let's take a look on the damage.

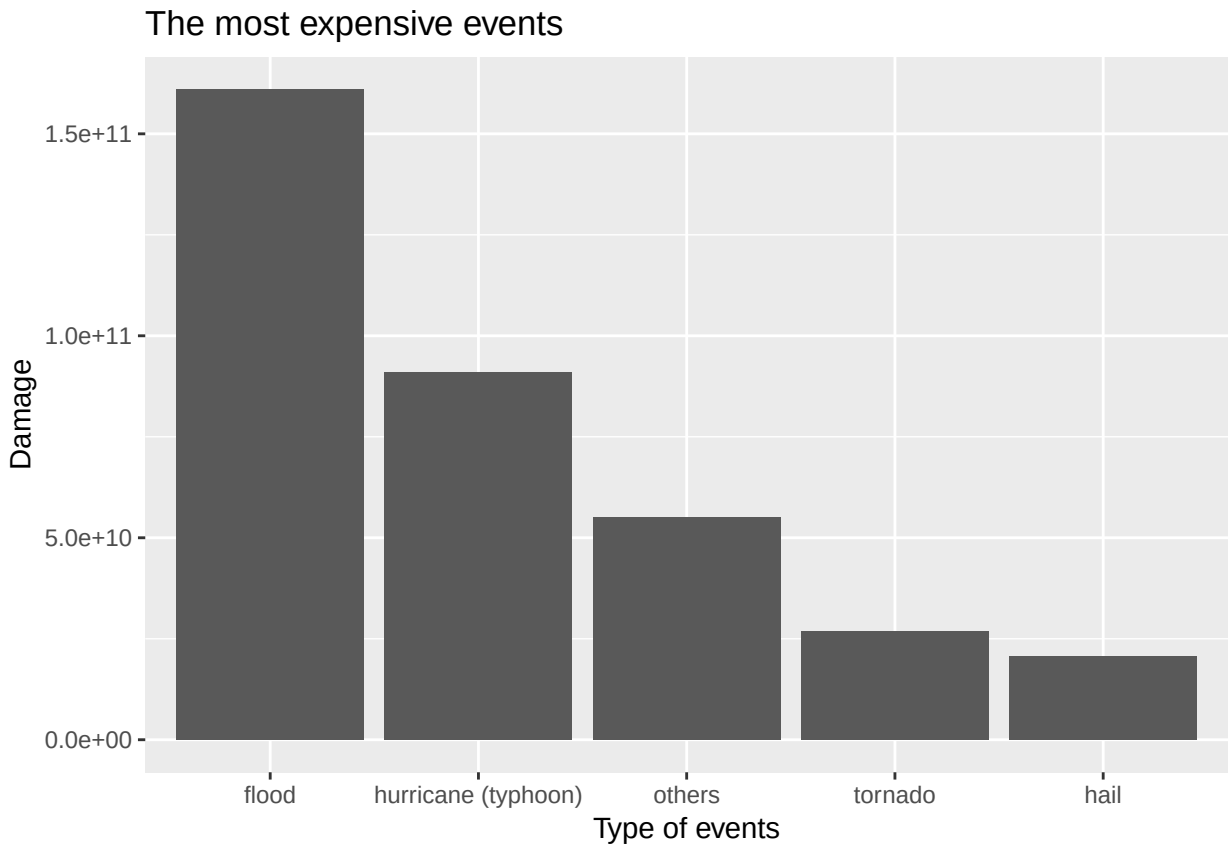
```
econom Cons <- data.frame(Event.Type = fixedType, Damage = prop + crop,stringsAsFactors = FALSE)
ec <- aggregate(Damage~Event.Type, data = econom Cons, sum)
ec$Event.Type[grep("^$",ec$Event.Type)] <- "others"

topD <- quantile(ec$Damage, probs = 0.9)
ecD <- filter(ec, ec$Damage > topD)
print("The most expensive events")
```

```
## [1] "The most expensive events"
```

```
ecD
```

```
##            Event.Type       Damage
## 1              others  55191499580
## 2               flood 161013873600
## 3                hail  20737200326
## 4 hurricane (typhoon)  90872527810
## 5             tornado  26820081214
```

```
gg3<- ggplot(ecD,aes(x=reorder(Event.Type,-Damage),y = Damage)) +
geom_col() +
labs(x="Type of events", title = "The most expensive events")
print(gg3)
```


The most expensive events
1.5e+11
1.0e+11
5.0e+10
0.0e+00
Damage
flood
hurricane (typhoon)
others
tornado
hail
Type of events