Storm Effects on Communities, Analysis

Anandu R

8/6/2020

Storms and other severe weather events can cause both public health and economic problems for communities and municipalities. Many severe events can result in fatalities, injuries, and property damage, and preventing such outcomes to the extent possible is a key concern.

This project involves exploring the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database. This 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.

Data Processing

There is also some documentation of the database available. Details on how some of the variables are constructed/defined is available on this website by National Weather Service : Storm Data Documentation

Getting the data

```
fileUrl = "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
if(!file.exists("./data/data.csv.bz2")){
  download.file(fileUrl,"./data/data.csv.bz2")
}
```

Reading the data

```
suppressMessages(library(dplyr))
data_raw <- read.csv("./data/data.csv.bz2", sep =",", header = T)
```

```
head(data_raw)
```

Preliminary analysis of data

```
##   STATE__           BGN_DATE BGN_TIME TIME_ZONE COUNTY COUNTYNAME STATE  EVTYPE
## 1       1  4/18/1950 0:00:00     0130       CST     97     MOBILE    AL TORNADO
## 2       1  4/18/1950 0:00:00     0145       CST      3    BALDWIN    AL TORNADO
## 3       1  2/20/1951 0:00:00     1600       CST     57    FAYETTE    AL TORNADO
## 4       1   6/8/1951 0:00:00     0900       CST     89    MADISON    AL TORNADO
```

```
## 5       1 11/15/1951 0:00:00     1500       CST     43    CULLMAN    AL TORNADO
## 6       1 11/15/1951 0:00:00     2000       CST     77 LAUDERDALE    AL TORNADO
##   BGN_RANGE BGN_AZI BGN_LOCATI END_DATE END_TIME COUNTY_END COUNTYENDN
## 1         0                                               0         NA
## 2         0                                               0         NA
## 3         0                                               0         NA
## 4         0                                               0         NA
## 5         0                                               0         NA
## 6         0                                               0         NA
##   END_RANGE END_AZI END_LOCATI LENGTH WIDTH F MAG FATALITIES INJURIES PROPDMG
## 1         0                    14.0   100 3   0          0       15    25.0
## 2         0                     2.0   150 2   0          0        0     2.5
## 3         0                     0.1   123 2   0          0        2    25.0
## 4         0                     0.0   100 2   0          0        2     2.5
## 5         0                     0.0   150 2   0          0        2     2.5
## 6         0                     1.5   177 2   0          0        6     2.5
##   PROPDMGEXP CROPDMG CROPDMGEXP WFO STATEOFFIC ZONENAMES LATITUDE LONGITUDE
## 1          K       0                                         3040      8812
## 2          K       0                                         3042      8755
## 3          K       0                                         3340      8742
## 4          K       0                                         3458      8626
## 5          K       0                                         3412      8642
## 6          K       0                                         3450      8748
##   LATITUDE_E LONGITUDE_ REMARKS REFNUM
## 1       3051       8806              1
## 2          0          0              2
## 3          0          0              3
## 4          0          0              4
## 5          0          0              5
## 6          0          0              6
```

Reading column names

```
names(data_raw)
```

```
##  [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"
```

Data Cleaning

Removing unnecessary variables Since the END_DATE and END_TIME fields are same as the BGN_DATA and BGN_TIME, we also remove those columns from the data.

Furthermore, since the COUNTY_END field has only the value 0 and would serve no purpose to the analysis, it too is removed

The "REFNUM" and "REMARKS" fields don't serve any purpose to our analysis

```
data_clean = select(data_raw,
                    STATE,
                    COUNTY,
                    BGN_DATE,
                    BGN_TIME,
                    EVTYPE,
                    FATALITIES,
                    INJURIES,
                    PROPDMG,
                    PROPDMGEXP,
                    CROPDMG,
                    CROPDMGEXP)
```

```
datatypes = as.character(sapply(data_clean, class))
character_loc = which(datatypes == "character")
arr_missing = array(dim = length(character_loc))
j=1
for(i in character_loc){
  arr_missing[j] = mean(data_clean[,i]=="")
  j = j+1
}
arr_missing_r = character_loc[which(arr_missing*100 < 2 & arr_missing > 0)]
arr_missing_c = character_loc[which(arr_missing*100 > 50)]
arr_NAs_r = which(as.numeric(colMeans(is.na(data_clean))) > 0 &
                    as.numeric(colMeans(is.na(data_clean)))*100 < 2)
arr_NAs_c = which(as.numeric(colMeans(is.na(data_clean)))*100 > 50)
```

Checking distribution of Missing data and NAs in the dataset Columns 9, and 11 represent the "PROPDMGEXP", "CROPDMGEXP" fields which are required for the analysis therefore we will keep them.

Therefore all in all, there arent any records to be removed or are their any columns that can be removed.

Checking distribution of missing value and NAs

```
as.numeric(colMeans(is.na(data_clean)))
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0
```

```
as.numeric(colMeans(data_clean==""))
```

```
##  [1] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##  [8] 0.0000000 0.5163865 0.0000000 0.6853763
```

NOTE: During analysis there may still be some fields with no value aka missing values in certain columns, but their percentages are in range 10-50% so the next suitable step would be to impute the values in the dataset, but since it is the weather data, imputing values would only create noise in the data(?)

Looking at cleaned data

```
head(data_clean)
```

```
##   STATE COUNTY           BGN_DATE BGN_TIME  EVTYPE FATALITIES INJURIES PROPDMG
## 1    AL     97  4/18/1950 0:00:00     0130 TORNADO          0       15    25.0
## 2    AL      3  4/18/1950 0:00:00     0145 TORNADO          0        0     2.5
## 3    AL     57  2/20/1951 0:00:00     1600 TORNADO          0        2    25.0
## 4    AL     89   6/8/1951 0:00:00     0900 TORNADO          0        2     2.5
## 5    AL     43 11/15/1951 0:00:00     1500 TORNADO          0        2     2.5
## 6    AL     77 11/15/1951 0:00:00     2000 TORNADO          0        6     2.5
##   PROPDMGEXP CROPDMG CROPDMGEXP
## 1          K       0
## 2          K       0
## 3          K       0
## 4          K       0
## 5          K       0
## 6          K       0
```

Fixing the datatypes and datafields

```
data_clean$BGN_DATE =
  as.POSIXct(data_clean$BGN_DATE, format = "%m/%d/%Y %H:%M:%S")

data_clean$BGN_TIME =
  format(strptime(data_clean$BGN_TIME,"%H%M"),'%H:%M')

data_clean$BGN_DATETIME =
  as.POSIXct(paste(data_clean$BGN_DATE,
                   data_clean$BGN_TIME
                   ), format="%Y-%m-%d %H:%M")
data_clean =
  select(data_clean,
         STATE, COUNTY,
         BGN_DATETIME,
         EVTYPE, FATALITIES,
         INJURIES,
         PROPDMG,
         PROPDMGEXP,
         CROPDMG,
         CROPDMGEXP)
```

Creating a datatime field

Imputing proper values in the "PROPDMGEXP", "CROPDMGEXP" fields Current values in "CROPDMGEXP"

```
unique(data_clean$CROPDMGEXP)
```

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

Current values in “PROPDMGEXP”

```
unique(data_clean$PROPDMGEXP)
```

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

Correct representations:
- “""” = 10^0,
- “-” = 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

Imputing the correct values

```
data_clean = transform(data_clean,
                       PROPDMGEXP = toupper(PROPDMGEXP),
                       CROPDMGEXP = toupper(CROPDMGEXP))
DmgExP =  c("\"\"" = 10^0,
            "-" = 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)
data_clean = transform(
  data_clean,
  PROPDMGEXP = as.numeric(DmgExP[as.character(data_clean[,"PROPDMGEXP"])]),
  CROPDMGEXP = as.numeric(DmgExP[as.character(data_clean[,"CROPDMGEXP"])])
)
```

```
data_clean = transform(
  data_clean,
  PROPDMGEXP = ifelse(is.na(PROPDMGEXP),10^0,PROPDMGEXP),
  CROPDMGEXP = ifelse(is.na(CROPDMGEXP),10^0,CROPDMGEXP)
)
```

Subsetting the data, removing EVTYPEs that have 0 impact of any sort

```
data_clean = subset(data_clean,
                    EVTYPE != "?" &
                      (INJURIES > 0 |
                         FATALITIES > 0 |
                         PROPDMG > 0 |
                         CROPDMG > 0)
                    )
```

Looking at cleaned data

```
head(data_clean)
```

```
##   STATE COUNTY        BGN_DATETIME  EVTYPE FATALITIES INJURIES PROPDMG
## 1    AL     97 1950-04-18 01:30:00 TORNADO          0       15    25.0
## 2    AL      3 1950-04-18 01:45:00 TORNADO          0        0     2.5
## 3    AL     57 1951-02-20 16:00:00 TORNADO          0        2    25.0
## 4    AL     89 1951-06-08 09:00:00 TORNADO          0        2     2.5
## 5    AL     43 1951-11-15 15:00:00 TORNADO          0        2     2.5
## 6    AL     77 1951-11-15 20:00:00 TORNADO          0        6     2.5
##   PROPDMGEXP CROPDMG CROPDMGEXP
## 1       1000       0          1
## 2       1000       0          1
## 3       1000       0          1
## 4       1000       0          1
## 5       1000       0          1
## 6       1000       0          1
```

Analysing the event types and fixing field names

```
unique(data_clean$EVTYPE)[1:10]
```

```
##  [1] "TORNADO"                    "TSTM WIND"
##  [3] "HAIL"                       "ICE STORM/FLASH FLOOD"
##  [5] "WINTER STORM"               "HURRICANE OPAL/HIGH WINDS"
##  [7] "THUNDERSTORM WINDS"         "HURRICANE ERIN"
##  [9] "HURRICANE OPAL"             "HEAVY RAIN"
```

Labeling event types correctly

```
## WIND 1
data_clean[data_clean$EVTYPE=="NON TSTM WIND","EVTYPE"] = "WIND"
data_clean[data_clean$EVTYPE=="NON-TSTM WIND","EVTYPE"] = "WIND"

## THUNDERSTORM
data_clean[grepl("thunderstorm",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "THUNDERSTORM"
data_clean[grepl("thunderestorm",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "THUNDERSTORM"
data_clean[grepl("thundeerstorm",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "THUNDERSTORM"
data_clean[grepl("thunerstorm",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "THUNDERSTORM"
data_clean[grepl("THUNDERTORM WINDS",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "THUNDERSTORM"
data_clean[grepl("TUNDERSTORM WIND",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "THUNDERSTORM"
data_clean[grepl("THUDERSTORM WINDS",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "THUNDERSTORM"
data_clean[grepl("THUNDERSTROM WIND",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "THUNDERSTORM"
data_clean[grepl("tstm",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "THUNDERSTORM"

## WATERSPOUT + TORNADO
data_clean[grepl("WATERSPOUT[\" \",/,-,+]TORNADO",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "WATERSPOUT+TORNADO"
data_clean[grepl("WATERSPOUT-TORNADO",
                 data_clean$EVTYPE),"EVTYPE"] = "WATERSPOUT+TORNADO"
data_clean[grepl("WATERSPOUT/ TORNADO",
                 data_clean$EVTYPE),"EVTYPE"] = "WATERSPOUT+TORNADO"
data_clean[grepl("WATERSPOUT",
                 data_clean$EVTYPE),"EVTYPE"] = "WATERSPOUT+TORNADO"

## TORNADO
data_clean[grepl("^TORNADO",
                 data_clean$EVTYPE),"EVTYPE"] = "TORNADO"
data_clean[grepl("TORNDAO",
                 data_clean$EVTYPE),"EVTYPE"] = "TORNADO"
data_clean[grepl("FUNNEL CLOUD",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "TORNADO"
data_clean[grepl("COLD AIR TORNADO",
```

```
                data_clean$EVTYPE),"EVTYPE"] = "TORNADO"

## LANDSLIDE
data_clean[grepl("landslide",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "LANDSLIDE"

## FLASH FLOOD
data_clean[grepl("FLASH FLOOD",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "FLASH FLOOD"
data_clean[grepl("flash*FLOOD",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "FLASH FLOOD"
data_clean[grepl("FLOOD[/,\" \"]flash",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "FLASH FLOOD"

## COASTAL FLOOD
data_clean[grepl("COASTAL FLOOD",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "COASTAL FLOOD+EROSION"
data_clean[grepl("COASTAL  FLOODING/EROSION",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "COASTAL FLOOD+EROSION"
data_clean[grepl("Erosion/Cstl Flood",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "COASTAL FLOOD+EROSION"
data_clean[grepl("Erosion",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "COASTAL FLOOD+EROSION"

## FLOODS
data_clean[grepl("FLOODING",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "FLOOD"
data_clean[grepl("FLOODS",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "FLOOD"
data_clean[grepl("RAPIDLY RISING WATER",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "FLOOD"

## OTHER FLOODS
data_clean[grepl("URBAN",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "URBAN FLOOD"
data_clean[grepl("RIVER",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "RIVER FLOOD"
data_clean[grepl("FLOOD/RAIN/WINDS",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "FLOOD"
```

```
data_clean[grepl("HEAVY RAIN AND FLOOD",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "FLOOD"
data_clean[grepl("HEAVY SNOW/HIGH WINDS & FLOOD",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "FLOOD"
data_clean[grepl("FLOOD & HEAVY RAIN",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "FLOOD"
data_clean[grepl("Ice jam flood \\(minor",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "FLOOD"
data_clean[grepl("LAKE FLOOD",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "RURAL FLOOD"
data_clean[grepl("LAKESHORE FLOOD",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "RURAL FLOOD"
data_clean[grepl("MAJOR FLOOD",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "URBAN FLOOD"
data_clean[grepl("RIVER FLOOD",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "RURAL FLOOD"
data_clean[grepl("SMALL STREAM FLOOD",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "RURAL FLOOD"

## TIDE
data_clean[grepl("TIDE",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "TIDE"

## AVALANCHE
data_clean[grepl("Avalanche",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "AVALANCHE"
data_clean[grepl("Avalance",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "AVALANCHE"

## ICE SNOW BLIZZARD
data_clean[grepl("FROST",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("FREEZE",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("COLD",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("snow",
                 data_clean$EVTYPE,
```

```
                  ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("chill",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("low temp",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("winter",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "BLIZZARD"
data_clean[grepl("blizzard",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "BLIZZARD"
data_clean[grepl("ice storm",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "BLIZZARD"
data_clean[grepl("ice",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("wintr",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("freez",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("sleet",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "FROST+SNOW"
data_clean[grepl("^glaze",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "FROST+SNOW"

## HEAT
data_clean[grepl("HEAT",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "HEAT+DROUGHT"
data_clean[grepl("WARM",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "HEAT+DROUGHT"
data_clean[grepl("DROUGHT",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "HEAT+DROUGHT"

## DUST
data_clean[grepl("DUST",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "DUST"

## WILDFIRE
data_clean[grepl("FIRE",
                  data_clean$EVTYPE,
                  ignore.case = T),"EVTYPE"] = "WILDFIRE"
```

```
## HURRICANE
data_clean[grepl("hurricane",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "HURRICANE"

## HAIL
data_clean[grepl("HAIL",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "HAIL"

## SURF
data_clean[grepl("surf",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "SURF"

## WIND 2
data_clean[grepl("wind",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "WIND"
data_clean[grepl("burst",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "WIND"

## MUDSLIDES
data_clean[grepl("mud",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "MUD+LAND SLIDES"
data_clean[grepl("land",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "MUD+LAND SLIDES"
data_clean[grepl("rock",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "MUD+LAND SLIDES"

## RAINFALL
data_clean[grepl("rain",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "RAINFALL"
data_clean[grepl("precip",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "RAINFALL"
data_clean[grepl("show",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "RAINFALL"

## LIGHTNING
data_clean[grepl("light",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "LIGHTNING"
data_clean[grepl("ligntning",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "LIGHTNING"
```

```
## TROPICAL STORM
data_clean[grepl("tropical",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "TROPICAL CYCLONE"

## SURGE
data_clean[grepl("surge",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "STORM SURGE"

## SLEET
data_clean[grepl("sleet",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "SLEET"

## RIP CURRENT
data_clean[grepl("rip",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "RIP CURRENT"

## GUST
data_clean[grepl("gust",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "WIND"

## HYPOTHERMIA
data_clean[grepl("hypo",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "HYPOTHERMIA"

## HYPERTHERMIA
data_clean[grepl("hyper",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "HYPERTHERMIA"

## SWELLS + SEAS + MIX + HIGH WATER + WAVES
## WET
data_clean[grepl("wet",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "WETNESS"

## TIDES
temp = unique(data_clean$EVTYPE)[c(20,27,39,40,42,43,44,54,57,58)]
data_clean[data_clean$EVTYPE %in% temp,"EVTYPE"] = "TIDE/ROUGH SEAS"

## SEICHE
data_clean[grepl("seiche",
                 data_clean$EVTYPE,
                 ignore.case = T),"EVTYPE"] = "TIDE/ROUGH SEAS"

## COASTAL STORM
data_clean[grepl("coastal storm",
                 data_clean$EVTYPE,
```

```
                ignore.case = T),"EVTYPE"] = "COASTAL STORM"
data_clean[grepl("coastalstorm",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "COASTAL STORM"

## MARINE MISHAP/ACCIDENT
data_clean[grepl("marine",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "MARINE ACCIDENT"

## OTHER
data_clean[grepl("^other",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "OTHER"

## APACHE COUNTY to WIND since mentioned in REMARKS
data_clean[grepl("APACHE COUNTY",
                data_clean$EVTYPE,
                ignore.case = T),"EVTYPE"] = "WIND"
```

```
unique(data_clean$EVTYPE)
```

```
##  [1] "TORNADO"               "THUNDERSTORM"          "HAIL"
##  [4] "FLASH FLOOD"           "BLIZZARD"              "HURRICANE"
##  [7] "RAINFALL"              "LIGHTNING"             "DENSE FOG"
## [10] "RIP CURRENT"           "HEAT+DROUGHT"          "WIND"
## [13] "FROST+SNOW"            "FLOOD"                 "WATERSPOUT+TORNADO"
## [16] "RURAL FLOOD"           "AVALANCHE"             "MARINE ACCIDENT"
## [19] "TIDE"                  "TIDE/ROUGH SEAS"       "COASTAL FLOOD+EROSION"
## [22] "SEVERE TURBULENCE"     "DUST"                  "SURF"
## [25] "WILDFIRE"              "MUD+LAND SLIDES"       "URBAN FLOOD"
## [28] "STORM SURGE"           "TROPICAL CYCLONE"      "WETNESS"
## [31] "FOG"                   "ICY ROADS"             "HEAVY MIX"
## [34] "HIGH WAVES"            "HYPOTHERMIA"           "HEAVY SEAS"
## [37] "OTHER"                 "COASTAL STORM"         "DAM BREAK"
## [40] "TYPHOON"               "HIGH SWELLS"           "HYPERTHERMIA"
## [43] "ROUGH SEAS"            "ROGUE WAVE"            "DROWNING"
## [46] "TSUNAMI"
```

Creating new fields CROPDMGPRICE and PROPDMGPRICE

```
data_clean = transform(data_clean,
                       CROPDMGPRICE = CROPDMG*CROPDMGEXP,
                       PROPDMGPRICE = PROPDMG*PROPDMGEXP)
```

Aggregating the data based on event type

```
library(dplyr)
suppressMessages(
```

```
  {
  data_aggr = data_clean %>%
    group_by(EVTYPE) %>%
    summarise(
      FATALITIES = sum(FATALITIES, na.rm = T),
      INJURIES = sum(INJURIES,na.rm = T),
      CROPDMGPRICE = sum(CROPDMGPRICE, na.rm = T),
      PROPDMGPRICE = sum(PROPDMGPRICE, na.rm = T)
    )
  }
)
head(data_aggr[order(-data_aggr[,"FATALITIES"],
                     -data_aggr[,"INJURIES"],
                     -data_aggr[,"CROPDMGPRICE"],
                     -data_aggr[,"PROPDMGPRICE"]),])
```

```
## # A tibble: 6 x 5
##   EVTYPE       FATALITIES INJURIES CROPDMGPRICE PROPDMGPRICE
##   <chr>             <dbl>    <dbl>        <dbl>        <dbl>
## 1 TORNADO            5633    91367    414961520 56952347026.
## 2 HEAT+DROUGHT       3178     9247  14877045280  1066431750
## 3 FLASH FLOOD        1035     1802   1532197150 17589261096.
## 4 LIGHTNING           817     5231     12092090   930419430.
## 5 THUNDERSTORM        755     9543   1274213988 12785456700.
## 6 FROST+SNOW          659     1986   3565490400  1315567650
```

Exploratory Analysis

```
names(data_clean)
```

```
##  [1] "STATE"        "COUNTY"       "BGN_DATETIME" "EVTYPE"       "FATALITIES"
##  [6] "INJURIES"     "PROPDMG"      "PROPDMGEXP"   "CROPDMG"      "CROPDMGEXP"
## [11] "CROPDMGPRICE" "PROPDMGPRICE"
```

Analyis to find events most harmful with respect to population health

Looking at data in relavant columns “FATALITIES” and “INJURIES”

```
head(data_clean[,c("EVTYPE","FATALITIES","INJURIES")])
```

```
##    EVTYPE FATALITIES INJURIES
## 1 TORNADO          0       15
## 2 TORNADO          0        0
## 3 TORNADO          0        2
## 4 TORNADO          0        2
## 5 TORNADO          0        2
## 6 TORNADO          0        6
```

Removing data file after analysis

```
unlink("./data/data.csv.bz2",recursive = T)
#unlink("./analysis_cache", recursive = T)
```