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

```
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

```


datatypes = as.character(sapply(data_raw, 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_raw[,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_raw))) > 0 & as.numeric(colMeans(is.na(data_raw)))*100
arr_NAs_c = which(as.numeric(colMeans(is.na(data_raw)))*100 > 50)
arr_missing
```

Checking distrobution of Missing data and NAs in the dataset

```
##  [1] 0.000000000 0.000000000 0.000000000 0.001761061 0.000000000 0.000000000
##  [7] 0.606598493 0.318900539 0.269768158 0.264855142 0.803324183 0.553282345
## [13] 0.516386511 0.685376323 0.157452590 0.275706336 0.658351962 0.318556972
```

Removing columns that have more that 50% data misssing Columns 26, and 28 represent the "PROPDMGEXP", "CROPDMGEXP" fields which are required for the analysis therefore we will keep them.

```
arr_missing_c = arr_missing_c[-c(4,5)]
data_clean = data_raw[,-c(arr_missing_c,arr_NAs_c)]
```

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

```
data_clean = data_clean[,-c(11:13)]
```

```


with_NAs = complete.cases(data_raw[,arr_NAs_r])
data_clean = subset(data_clean, with_NAs)
data_clean = data_clean[!(data_clean[,arr_missing_r]==""),]
```

Removing records with missing or NA data 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 0 0 0 0 0 0 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 0.0000000 0.3177455 0.0000000 0.0000000 0.0000000 0.0000000
## [15] 0.0000000 0.0000000 0.0000000 0.5155691 0.0000000 0.6848570 0.1560898
## [22] 0.2744762 0.0000000 0.0000000 0.0000000 0.0000000 0.3174058 0.0000000
```

There are still 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__           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_LOCATI END_RANGE LENGTH WIDTH MAG FATALITIES INJURIES PROPDMG
## 1         0                    0   14.0   100   0          0       15    25.0
## 2         0                    0    2.0   150   0          0        0     2.5
## 3         0                    0    0.1   123   0          0        2    25.0
## 4         0                    0    0.0   100   0          0        2     2.5
## 5         0                    0    0.0   150   0          0        2     2.5
## 6         0                    0    1.5   177   0          0        6     2.5
##   PROPDMGEXP CROPDMG CROPDMGEXP WFO STATEOFFIC LATITUDE LONGITUDE LATITUDE_E
## 1          K       0                               3040      8812       3051
## 2          K       0                               3042      8755          0
## 3          K       0                               3340      8742          0
## 4          K       0                               3458      8626          0
## 5          K       0                               3412      8642          0
## 6          K       0                               3450      8748          0
##   LONGITUDE_ REMARKS REFNUM
## 1       8806              1
## 2          0              2
## 3          0              3
## 4          0              4
## 5          0              5
## 6          0              6
```

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
```

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: - “""” = 10ˆ0, - “-” = 10ˆ0, - “?” = 10ˆ0, - “+” = 10ˆ0, - “0” = 10ˆ0, - “1” = 10ˆ1, - “2” = 10ˆ2, - “3” = 10ˆ3, - “4” = 10ˆ4, - “5” = 10ˆ5, - “6” = 10ˆ6, - “7” = 10ˆ7, - “8” = 10ˆ8, - “9” = 10ˆ9, - “H” = 10ˆ2, - “K” = 10ˆ3, - “M” = 10ˆ6, - “B” = 10ˆ9

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
```

Aggregating the results

```
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
```

```
data_ = data_clean %>%
  group_by(EVTYPE) %>%
  summarise(
    FATALITIES = sum(FATALITIES, na.rm = T),
    INJURIES = sum(INJURIES,na.rm = T),
    CROPDMG = sum(CROPDMG, na.rm = T),
    PROPDMG = sum(PROPDMG, na.rm = T)
  )
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
data_
```

```
## # A tibble: 487 x 5
##    EVTYPE                   FATALITIES INJURIES CROPDMG PROPDMG
##    <chr>                         <dbl>    <dbl>   <dbl>   <dbl>
##  1 "   HIGH SURF ADVISORY"           0        0     0       200
##  2 " FLASH FLOOD"                    0        0     0        50
##  3 " TSTM WIND"                      0        0     0       108
##  4 " TSTM WIND (G45)"                0        0     0         8
##  5 "AGRICULTURAL FREEZE"             0        0    28.8       0
##  6 "APACHE COUNTY"                   0        0     0         5
##  7 "ASTRONOMICAL HIGH TIDE"          0        0     0       934.
##  8 "ASTRONOMICAL LOW TIDE"           0        0     0       320
##  9 "AVALANCE"                        1        0     0         0
## 10 "AVALANCHE"                     224      170     0      1624.
## # ... with 477 more rows
```

Exploratory Analysis

```
names(data_clean)
```

```
##  [1] "STATE__"      "BGN_DATE"     "BGN_TIME"     "TIME_ZONE"    "COUNTY"
##  [6] "COUNTYNAME"   "STATE"        "EVTYPE"       "BGN_RANGE"    "BGN_LOCATI"
## [11] "END_RANGE"    "LENGTH"       "WIDTH"        "MAG"          "FATALITIES"
## [16] "INJURIES"     "PROPDMG"      "PROPDMGEXP"   "CROPDMG"      "CROPDMGEXP"
## [21] "WFO"          "STATEOFFIC"   "LATITUDE"     "LONGITUDE"    "LATITUDE_E"
## [26] "LONGITUDE_"   "REMARKS"      "REFNUM"       "BGN_DATETIME"
```

Analyis to find events that are 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)
```