Economic and Health Consequences of Various Weather Events

Nick Newsome

March 17, 2020

1. Synopsis

This is an analysis of weather events contained within the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database. This data set can be downloaded here: https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2 (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2). Documentation on the data set can be found here: https://d396qusza40orc.cloudfront.net /repdata%2Fpeer2_doc%2FNCDC%20Storm%20Events-FAQ%20Page.pdf (https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2FNCDC%20Storm%20Events-FAQ%20Page.pdf) and https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf (https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf). I took the data set and grouped the event types into similar groups to limit and structure the output. Once cleaned, I was able to determine which events had the greatest impact on the health of the population and which had the greatest economic costs. The health consequences were determined by number of fatalities, injuries, and by both fatalities and injuries. These were calculated by using a single event, the total over all events of each type, and by the average number within each type. The economic costs were determined by combining property damage and crop damage into one economic category. The results of economic costs were calculated by using the single event with the greatest economic cost, the type of event that has cost the most economically overall, and the type that has the highest average cost per occurrence.

2. Data Processing

2.A. Initialize libraries

```
library(dplyr)
library(stringr)
library(data.table)
library(ggplot2)
library(cowplot)
```

2.B. Import dataset

The package R.utils may need to be installed before reading in the original data. It allows 'fread' to read compressed data. It may take a minute or more to import this data set.

```
data <- fread('repdata_data_StormData.csv.bz2')
summary(data)
```

```
##     STATE__       BGN_DATE           BGN_TIME          TIME_ZONE
##  Min.   : 1.0   Length:902297      Length:902297      Length:902297
##  1st Qu.:19.0   Class :character   Class :character   Class :character
##  Median :30.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :31.2
##  3rd Qu.:45.0
##  Max.   :95.0
##
##      COUNTY       COUNTYNAME           STATE              EVTYPE
##  Min.   :  0.0   Length:902297      Length:902297      Length:902297
##  1st Qu.: 31.0   Class :character   Class :character   Class :character
##  Median : 75.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :100.6
##  3rd Qu.:131.0
##  Max.   :873.0
##
##    BGN_RANGE          BGN_AZI           BGN_LOCATI
##  Min.   :   0.000   Length:902297      Length:902297
##  1st Qu.:   0.000   Class :character   Class :character
##  Median :   0.000   Mode  :character   Mode  :character
##  Mean   :   1.484
##  3rd Qu.:   1.000
##  Max.   :3749.000
##
##    END_DATE           END_TIME           COUNTY_END COUNTYENDN
##  Length:902297      Length:902297      Min.   :0    Mode:logical
##  Class :character   Class :character   1st Qu.:0    NA's:902297
##  Mode  :character   Mode  :character   Median :0
##                                        Mean   :0
##                                        3rd Qu.:0
##                                        Max.   :0
##
##    END_RANGE          END_AZI           END_LOCATI
##  Min.   :  0.0000   Length:902297      Length:902297
##  1st Qu.:  0.0000   Class :character   Class :character
##  Median :  0.0000   Mode  :character   Mode  :character
##  Mean   :  0.9862
##  3rd Qu.:  0.0000
##  Max.   :925.0000
##
##      LENGTH              WIDTH                F               MAG
##  Min.   :   0.0000   Min.   :   0.000   Min.   :0.0      Min.   :    0.0
##  1st Qu.:   0.0000   1st Qu.:   0.000   1st Qu.:0.0      1st Qu.:    0.0
##  Median :   0.0000   Median :   0.000   Median :1.0      Median :   50.0
##  Mean   :   0.2301   Mean   :   7.503   Mean   :0.9      Mean   :   46.9
##  3rd Qu.:   0.0000   3rd Qu.:   0.000   3rd Qu.:1.0      3rd Qu.:   75.0
##  Max.   :2315.0000   Max.   :4400.000   Max.   :5.0      Max.   :22000.0
##                                         NA's   :843563
##    FATALITIES          INJURIES            PROPDMG
##  Min.   :  0.0000   Min.   :   0.0000   Min.   :   0.00
```

```
##  1st Qu.:  0.0000   1st Qu.:   0.0000   1st Qu.:   0.00  
##  Median :  0.0000   Median :   0.0000   Median :   0.00  
##  Mean   :  0.0168   Mean   :   0.1557   Mean   :  12.06  
##  3rd Qu.:  0.0000   3rd Qu.:   0.0000   3rd Qu.:   0.50  
##  Max.   :583.0000   Max.   :1700.0000   Max.   :5000.00  
##                                                           
##   PROPDMGEXP           CROPDMG         CROPDMGEXP       
##  Length:902297      Min.   :  0.000   Length:902297     
##  Class :character   1st Qu.:  0.000   Class :character  
##  Mode  :character   Median :  0.000   Mode  :character  
##                     Mean   :  1.527                     
##                     3rd Qu.:  0.000                     
##                     Max.   :990.000                     
##                                                         
##      WFO             STATEOFFIC         ZONENAMES            LATITUDE   
##  Length:902297      Length:902297      Length:902297      Min.   :   0  
##  Class :character   Class :character   Class :character   1st Qu.:2802  
##  Mode  :character   Mode  :character   Mode  :character   Median :3540  
##                                                           Mean   :2875  
##                                                           3rd Qu.:4019  
##                                                           Max.   :9706  
##                                                           NA's   :47    
##    LONGITUDE        LATITUDE_E     LONGITUDE_       REMARKS         
##  Min.   :-14451   Min.   :   0   Min.   :-14455   Length:902297     
##  1st Qu.:  7247   1st Qu.:   0   1st Qu.:     0   Class :character  
##  Median :  8707   Median :   0   Median :     0   Mode  :character  
##  Mean   :  6940   Mean   :1452   Mean   :  3509                     
##  3rd Qu.:  9605   3rd Qu.:3549   3rd Qu.:  8735                     
##  Max.   : 17124   Max.   :9706   Max.   :106220                     
##                   NA's   :40                                        
##      REFNUM      
##  Min.   :     1  
##  1st Qu.:225575  
##  Median :451149  
##  Mean   :451149  
##  3rd Qu.:676723  
##  Max.   :902297  
## 
```

Verify that all rows were imported, there should be 902,297

```
cat('Number of rows imported:', NROW(data))
```

```
## Number of rows imported: 902297
```

2.C. Preliminary Cleaning

For this analysis, time and location (other than the state) are not necessary. Gather only the necessary columns and rename them to more friendly names.

```
data <- select(data, BGN_DATE, BGN_TIME, COUNTYNAME, STATE, EVTYPE, FATALITIES, INJUR
IES, PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP)
data <- rename(data, Date=BGN_DATE, Time=BGN_TIME, County=COUNTYNAME, State=STATE, Ty
pe=EVTYPE, Fatalities=FATALITIES, Injuries=INJURIES, PropDamage=PROPDMG, PDExp=PROPDM
GEXP, CropDamage=CROPDMG, CDExp=CROPDMGEXP)
```

Remove time from date

```
data$Date <- str_sub(data$Date, end=-9)
```

Glimpse the current state of data

```
head(data, 3)
```

```
##         Date Time  County State    Type Fatalities Injuries PropDamage
## 1: 4/18/1950 0130  MOBILE    AL TORNADO          0       15       25.0
## 2: 4/18/1950 0145 BALDWIN    AL TORNADO          0        0        2.5
## 3: 2/20/1951 1600 FAYETTE    AL TORNADO          0        2       25.0
##    PDExp CropDamage CDExp
## 1:     K          0
## 2:     K          0
## 3:     K          0
```

2.D. Preprocessing

2.D.1. Organize Event Types

Find out how many unique events there are.

```
uniquetype <- distinct(data, Type)
cat('There are', NROW(uniquetype), 'distinct event types')
```

```
## There are 985 distinct event types
```

There are too many categories based off the documentation. This is due to misspellings and combining two categories into one, or simply putting a name that was not one of the categories. I began by grouping some types together, such as Hurricane, Typhoon, Tropical Depression, and Tropical Storm, all into one group. I continued to do this and checked on the number of unique types ‘uniquetype’ to see what was being overlooked. I eventually settled on 30 categories and then combined anything left over into a 31st category named ‘Other’. As the plots will show, this leftover category did not play any role in the results I was looking for.

Create lists of names to search for in order to group event types.

```
data$Type <- tolower(data$Type)
hurricane <- c('hurricane', 'tropical storm', 'typhoon', 'tropical depression', 'floy
d')
tornado <- c('tornado', 'torndao')
lightning <- c('lightning', 'lighting', 'ligntning')
hail <- 'hail'
micro <- 'burst'
wind <- c('wind', 'wnd')
tstorm <- c('tstm', 'thunderstorm', 'gustnado')
cflood <- 'coastal flood'
lflood <- 'lakeshore flood'
fflood <- 'flash'
flood <- 'flood'
tsunami <- c('tsunami', 'rogue')
cloud <- c('funnel', 'cloud', 'wall')
fog <- c('fog', 'vog')
avalanche <- c('avalan', 'slide')
fire <- c('fire', 'smoke')
dust <- 'dust'
drought <- c('drough', 'dry', 'driest')
surf <- c('tide', 'surf', 'storm surge', 'coastal surge', 'beach', 'coastal erosion',
'coastal storm', 'coastalstorm')
current <- 'rip current'
seiche <- 'seiche'
volcano <- 'volcan'
waterspout <- 'spout'
lesnow <- c('effect snow', 'lake snow')
blizzard <- 'blizzard'
wstorm <- c('winter', 'snow', 'ice', 'freezing rain', 'sleet', 'frost', 'wintry', 'mi
xed precip', 'icy', 'heavy mix', 'freezing drizzle')
rain <- c('rain', 'precip', 'unseasonably wet', 'shower', 'wet')
heat <- c('high', 'heat', 'warm', 'hot')
cold <- c('cold', 'record low', 'low temp', 'freeze', 'thermia', 'cool')
sea <- c('seas', 'swells', 'marine')
other <- c('Hurricane', 'Tornado', 'Lightning', 'Hail', 'Microburst', 'Wind', 'Thunde
rstorm','Coastal Flood', 'Lakeshore Flood', 'Flash Flood', 'Flood', 'Tsunami', 'Wall
or Funnel Cloud', 'Fog', 'Avalanche or Rock/Mudslide', 'Wildfires', 'Dust Storm', 'Dr
ough', 'High Tide/Surf or Coastal Storm/Erosion', 'Rip Current', 'Seiche', 'Volcanic
Activity', 'Waterspout', 'Lake-Effect Snow', 'Blizzard', 'Winter Storm', 'Rain', 'Ext
reme Heat', 'Extreme Cold', 'Marine or Sea Weather')
```

Use these categories to change each event type into one of the 31 revised categories.

```
data$Type <- replace(data$Type, grep(paste(hurricane, collapse = '|'), data$Type), 'Hurricane')
data$Type <- replace(data$Type, grep(paste(tornado, collapse = '|'), data$Type), 'Tornado')
data$Type <- replace(data$Type, grep(paste(lightning, collapse = '|'), data$Type), 'Lightning')
data$Type <- replace(data$Type, grep(paste(hail, collapse = '|'), data$Type), 'Hail')
data$Type <- replace(data$Type, grep(paste(micro, collapse = '|'), data$Type), 'Microburst')
data$Type <- replace(data$Type, grep(paste(wind, collapse = '|'), data$Type), 'Wind')
data$Type <- replace(data$Type, grep(paste(tstorm, collapse = '|'), data$Type), 'Thunderstorm')
data$Type <- replace(data$Type, grep(paste(cflood, collapse = '|'), data$Type), 'Coastal Flood')
data$Type <- replace(data$Type, grep(paste(lflood, collapse = '|'), data$Type), 'Lakeshore Flood')
data$Type <- replace(data$Type, grep(paste(fflood, collapse = '|'), data$Type), 'Flash Flood')
data$Type <- replace(data$Type, grep(paste(flood, collapse = '|'), data$Type), 'Flood')
data$Type <- replace(data$Type, grep(paste(tsunami, collapse = '|'), data$Type), 'Tsunami')
data$Type <- replace(data$Type, grep(paste(cloud, collapse = '|'), data$Type), 'Wall or Funnel Cloud')
data$Type <- replace(data$Type, grep(paste(fog, collapse = '|'), data$Type), 'Fog')
data$Type <- replace(data$Type, grep(paste(avalanche, collapse = '|'), data$Type), 'Avalanche or Rock/Mudslide')
data$Type <- replace(data$Type, grep(paste(fire, collapse = '|'), data$Type), 'Wildfires')
data$Type <- replace(data$Type, grep(paste(dust, collapse = '|'), data$Type), 'Dust Storm')
data$Type <- replace(data$Type, grep(paste(drought, collapse = '|'), data$Type), 'Drought')
data$Type <- replace(data$Type, grep(paste(surf, collapse = '|'), data$Type), 'High Tide/Surf or Coastal Storm/Erosion')
data$Type <- replace(data$Type, grep(paste(current, collapse = '|'), data$Type), 'Rip Current')
data$Type <- replace(data$Type, grep(paste(seiche, collapse = '|'), data$Type), 'Seiche')
data$Type <- replace(data$Type, grep(paste(volcano, collapse = '|'), data$Type), 'Volcanic Activity')
data$Type <- replace(data$Type, grep(paste(waterspout, collapse = '|'), data$Type), 'Waterspout')
data$Type <- replace(data$Type, grep(paste(lesnow, collapse = '|'), data$Type), 'Lake-Effect Snow')
data$Type <- replace(data$Type, grep(paste(blizzard, collapse = '|'), data$Type), 'Blizzard')
data$Type <- replace(data$Type, grep(paste(wstorm, collapse = '|'), data$Type), 'Winter Storm')
data$Type <- replace(data$Type, grep(paste(rain, collapse = '|'), data$Type), 'Rain')
```

```
data$Type <- replace(data$Type, grep(paste(heat, collapse = '|'), data$Type), 'Extrem
e Heat')
data$Type <- replace(data$Type, grep(paste(cold, collapse = '|'), data$Type), 'Extrem
e Cold')
data$Type <- replace(data$Type, grep(paste(sea, collapse = '|'), data$Type), 'Marine
or Sea Weather')
data$Type <- replace(data$Type, !grepl(paste(other, collapse = '|'), data$Type), 'Oth
er')
```

Now check the code to verify there are only 31 event types.

```
cat('There are now', NROW(distinct(data, Type)), 'distinct event types')
```

```
## There are now 31 distinct event types
```

2.D.2. Combine Economic Damage

The Economic Damage is separated into property damage and crop damage. Each of these columns (PropDamage and CropDamage) is a number, but the factor (such as thousands or millions) is in the PDExp and CDExp columns respectively. Need to look at the factors to determine how they are used, so that they can be multiplied by the PropDamage or CropDamage column to give a number of the total damage.

```
print('These are the factors for PropDamage:')
```

```
## [1] "These are the factors for PropDamage:"
```

```
print(as.list(distinct(data,PDExp)))
```

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

```
print('These are the factors for CropDamage:')
```

```
## [1] "These are the factors for CropDamage:"
```

```
print(as.list(distinct(data,CDExp)))
```

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

From the National Climatic Data Center Storm Events FAQ:
“Estimates should be rounded to three significant digits, followed by an alphabetical character signifying the magnitude of the number, i.e., 1.55B for $1,550,000,000. Alphabetical characters used to signify magnitude include “K” for thousands, “M” for millions, and “B” for billions.”

This leaves 'h' and 'H', the special characters, those left blank, and the digits as unknown factors.

I am unable to find anything in the documentation detailing what any of these other factors may mean. However, each event can be manually checked at https://www.ncdc.noaa.gov/stormevents/ (https://www.ncdc.noaa.gov/stormevents/). So I searched for these other factors to see what I could find. I started by changing all character factors to lowercase.

```
data$PDExp <- tolower(data$PDExp)
data$CDExp <- tolower(data$CDExp)
```

CDExp only contains 4 unknown factors (missing factor, ?, 0, 2), while PDExp has more. So I used PDExp in my search.

```
hasPD <- filter(data, (PDExp != 'b') & (PDExp != 'm') & (PDExp != 'k') & (PropDamage
!= 0))
cat('There are', NROW(distinct(hasPD, PDExp)), 'distinct factors for PropDamage that
have a value listed uner PropDamage\nThey are: \n')
```

```
## There are 11 distinct factors for PropDamage that have a value listed uner PropDam
age
## They are:
```

```
print(as.list(distinct(hasPD, PDExp)))
```

```
## $PDExp
##  [1] "+" "0" ""  "5" "6" "4" "h" "2" "7" "3" "-"
```

Starting with the factor 7, I gather all the rows with 7 for the factor and property damage not equal to 0.

```
seven <- filter(data, (PDExp == 7) & (PropDamage != 0))
cat('Number of events with property damage and a factor of 7:', NROW(seven))
```

```
## Number of events with property damage and a factor of 7: 2
```

```
head(seven)
```

```
##       Date Time      County State        Type Fatalities Injuries
## 1 6/7/1995 1210    FRANKLIN    MO        Wind          0        0
## 2 7/4/1995 1930 NORTHAMPTON    NC Flash Flood          0        0
##   PropDamage PDExp CropDamage CDExp
## 1         14     7          0
## 2         68     7          0
```

This gives two results, Franklin, MO and Northampton, NC. with PropDamage-PDExp of 14-7 and 6-7 respectively. Searching for the NC event does not produce any results. The MO search yields three results that appear to be identical except for damage costs. The property damages came in at $400, $300, and $147. The

$147 looks similar to the 14-7 PropDamage-PDExp results.

Next I search for factor 6.

```
six <- filter(data, (PDExp == 6) & (PropDamage != 0))
cat('Number of events with property damage and a factor of 6:', NROW(six))
```

```
## Number of events with property damage and a factor of 6: 3
```

```
head(six)
```

```
##        Date Time  County State Type Fatalities Injuries PropDamage PDExp
## 1 5/27/1995 1620 CLINTON    IL Wind          0        0         24     6
## 2  6/8/1995 0612 MADISON    IL Wind          0        0         26     6
## 3 5/18/1995 1255  MARION    IL Wind          0        0         15     6
##   CropDamage CDExp
## 1          0
## 2          0
## 3          0
```

Manually searching all three of these possibilities gives results similar to those found for MO for factor 7. Thus a pattern has emerged that if a factor is a digit, it is merely the ones place while the PropDamage shoud be multiplied by ten. So 24-6 becomes $246. This leaves missing factors, ‘h/H’, ‘+’ and ‘-’.

```
plus <- filter(data, (PDExp == '+') & (PropDamage != 0))
cat('Number of events with property damage and a factor of +:', NROW(plus))
```

```
## Number of events with property damage and a factor of +: 5
```

```
head(plus)
```

```
##        Date Time                         County State    Type Fatalities
## 1  5/1/1995 0000                         AKZ001    AK   Flood          0
## 2 12/1/1994 0000             AKZ024 - 006 - 005    AK    Wind          0
## 3  3/9/1995 1000                         CAZ001    CA   Flood          0
## 4 6/16/1995 0330 NVZ001 - 002 - 003 - 004 - 005    NV    Wind          0
## 5  6/5/1995 1304             NVZ003 - 004 - 007    NV Tornado          0
##   Injuries PropDamage PDExp CropDamage CDExp
## 1        0         20     +          0
## 2        0         20     +          0
## 3        0          2     +          0
## 4        0         15     +          0
## 5        0         60     +          0
```

Only one of these five showed up during the manual search (06/05/1995 NV) and the damage was $60 which would result in the ‘+’ being a factor of 1.

```
minus <- filter(data, (PDExp == '-') & (PropDamage != 0))
cat('Number of events with property damage and a factor of -:', NROW(minus))
```

```
## Number of events with property damage and a factor of -: 1
```

```
head(minus)
```

```
##          Date Time                     County State Type Fatalities
## 1 12/12/1995 1000 ORZ004 - 05 - 06 - 08 - 09    OR Wind          2
##   Injuries PropDamage PDExp CropDamage CDExp
## 1        0         15     -          0
```

Only one result for this search, and it could not be found manually. Best guess would be to use a factor of 1 like in the '+' factor.

```
missing <- filter(data, (PDExp == '') & (PropDamage != 0))
cat('Number of events with property damage and a missing factor:', NROW(missing))
```

```
## Number of events with property damage and a missing factor: 76
```

```
head(missing)
```

```
##        Date Time      County State        Type Fatalities Injuries
## 1  3/9/1995 0301       MARIN    CA        Wind          0        0
## 2  1/8/1993 1130       UNION    FL     Tornado          0        0
## 3 8/19/1995 1700     DECATUR    GA        Wind          0        0
## 4 3/31/1993 2015        HART    GA        Wind          0        0
## 5 5/12/1993 1630        CASS    IN        Wind          0        0
## 6 5/14/1995 0300 VANDERBURGH    IN Flash Flood          0        0
##   PropDamage PDExp CropDamage CDExp
## 1       0.41                0     ?
## 2       3.00                0
## 3       2.00                0
## 4       4.00                0
## 5       4.00                0
## 6      10.00                0
```

There are 76 results for this search. After choosing a few (Union, Fl 01/08/93, Decauter, GA 08/19/95, and Hart, GA 03/31/93), it appears that if the factor is missing, then it should be a 1.

```
hundred <- filter(data, (PDExp == 'h') & (PropDamage != 0))
cat('Number of events with property damage and a factor of h:', NROW(hundred))
```

```
## Number of events with property damage and a factor of h: 7
```

```
head(hundred)
```

```
##        Date Time   County State Type Fatalities Injuries PropDamage PDExp
## 1 9/16/1994 1630  CLINTON    MI Wind          0        0          2     h
## 2 7/14/1995 1923  SHERMAN    NE Wind          0        0          5     h
## 3 1/14/1995 1050  LAURENS    SC Wind          0        0          5     h
## 4  3/8/1995 1305  LAURENS    SC Wind          0        0          2     h
## 5 1/19/1995 1815  PICKENS    SC Wind          0        0          3     h
## 6 7/12/1995 1300 BIG HORN    WY Wind          0        0          5     h
##   CropDamage CDExp
## 1          0
## 2          0
## 3          0
## 4          0
## 5          0
## 6          0
```

There are 7 results with factor of ‘h’. Clinton, MI 09/16/1994, has a PropDamage of 2 which should be 200. Sherman, NE 07/14/1995 also indicates that ‘h’ is a factor of 100. These are the expected results and can be applied to others with a factor of ‘h’.

Change the factors to their appropriate numeric values. Since we are dealing with values in the billions, the ones place is irrelevant. So for digit factors, I simply used a factor of 10 and discarded the ones place.

```
data$PDExp <- gsub('[0-9]', 10, data$PDExp)
data$CDExp <- gsub('[0-9]', 10, data$CDExp)
data$PDExp <- gsub('[[:punct:]]',1,data$PDExp)
data$CDExp <- gsub('[[:punct:]]',1,data$CDExp)
replacements <- (c('b'='1000000000', 'm'='1000000', 'k'='1000', 'h'='100'))
data$PDExp <- str_replace_all(data$PDExp, c(replacements))
data$CDExp <- str_replace_all(data$CDExp, c(replacements))
data$PDExp <- as.integer(data$PDExp)
data$CDExp <- as.integer(data$CDExp)
data$PDExp[is.na(data$PDExp)] <- 1
data$CDExp[is.na(data$CDExp)] <- 1
```

There should now be only 6 factors (1e09, 1e06, 1000, 100, 10, 1).

```
factors <- full_join(distinct(data, PDExp), distinct(data, CDExp), by=c('PDExp'='CDEx
p'))
cat('There are', NROW(factors), 'distinct factors for PropDamage and CropDamage \nThe
y are: \n')
```

```
## There are 6 distinct factors for PropDamage and CropDamage
## They are:
```

```
print(as.list(factors))
```

```
## $PDExp
## [1] 1e+03 1e+06 1e+00 1e+09 1e+01 1e+02
```

In order to calculate the economic damage, the property and crop damage needs to be totaled.

```
data <- mutate(data, EconomicDamage=CropDamage*CDExp + PropDamage*PDExp)
```

2.D.3. Fatalities and Injuries Exclusive?

Are fatalities considered injuries? In other words, is the column ‘Injuries’ inclusive of ‘Fatalities’? If so, then you couldn’t have more fatalities than injuries.

```
cat('There are', NROW(filter(data, Injuries < Fatalities)), 'events with more fatalit
ies than injuries')
```

```
## There are 4459 events with more fatalities than injuries
```

So, Injuries are exclusive of Fatalities.

3. Results

3.A. Results from a Single Event

Which single event has caused the most fatalities?

```
singlemostfatal <- filter(data, Fatalities == max(data$Fatalities))
cat('Single most fatal event was', singlemostfatal[1,'Type'], 'on', singlemostfatal
[1,'Date'], 'in',
    singlemostfatal[1,'State'], 'with', singlemostfatal[1,'Fatalities'], 'fatalitie
s.')
```

```
## Single most fatal event was Extreme Heat on 7/12/1995 in IL with 583 fatalities.
```

```
rm(singlemostfatal)
```

Which single event has caused the most injuries?

```
singlemostinjury <- filter(data, Injuries == max(data$Injuries))
cat('Single most injurious event was', singlemostinjury[1,'Type'], 'on', singlemostin
jury[1,'Date'],
    'in', singlemostinjury[1,'State'], 'with', singlemostinjury[1,'Injuries'], 'injur
ies')
```

```
## Single most injurious event was Tornado on 4/10/1979 in TX with 1700 injuries
```

```
rm(singlemostinjury)
```

Which single event has caused the most fatalities and injuries?

```
data <- mutate(data, TotalFI=Fatalities+Injuries)
singlemostfi <- filter(data, TotalFI == max(data$TotalFI))
cat('Single event with most total fatalities and injuries', singlemostfi[1,'Type'], '
on', singlemostfi[1,'Date'], 'in',
    singlemostfi[1,'State'], 'with', singlemostfi[1,'TotalFI'], 'fatalities and injur
ies')
```

```
## Single event with most total fatalities and injuries Tornado on 4/10/1979 in TX wi
th 1742 fatalities and injuries
```

```
rm(singlemostfi)
```

Which single event has caused the most economic damage?

```
singlemosted <- filter(data, EconomicDamage == max(data$EconomicDamage))
cat('Single costliest event was', singlemosted[1,'Type'], 'on', singlemosted[1,'Date
'], 'in', singlemosted[1,'State'], 'which caused $', format(singlemosted[1,'EconomicD
amage'], big.mark=',', big.interval=3L), 'in damages')
```

```
## Single costliest event was Flood on 1/1/2006 in CA which caused $ 115,032,500,000
in damages
```

```
rm(singlemosted)
```

3.B. Total Results from All Events

Which type of event has caused the most total fatalities?

```
totalmostfatal <- arrange(aggregate(Fatalities ~ Type, data, sum), desc(Fatalities))
cat('Events causing the most total fatalities is', totalmostfatal[1,'Type'],
    'with', totalmostfatal[1,'Fatalities'], 'total fatalaties')
```

```
## Events causing the most total fatalities is Tornado with 5661 total fatalaties
```

```
rm(totalmostfatal)
```

Which type of event has caused the most total injuries?

```
totalmostinjury <- arrange(aggregate(Injuries ~ Type, data, sum), desc(Injuries))
cat('Events causing the most total injuries is', totalmostinjury[1,'Type'],
    'with', totalmostinjury[1,'Injuries'], 'total injuries')
```

```
## Events causing the most total injuries is Tornado with 91407 total injuries
```

```
rm(totalmostinjury)
```

Which type of event has caused the most total fatalities and injuries?

```
totalmostfi <- arrange(aggregate(TotalFI ~ Type, data, sum), desc(TotalFI))
cat('Events causing the most total fatalities and injuries is', totalmostfi[1,'Type
'],
    'with', totalmostfi[1,'TotalFI'], 'total fatalaties and injuries')
```

```
## Events causing the most total fatalities and injuries is Tornado with 97068 total
fatalaties and injuries
```

Which type of event has caused the most economic damage?

```
totalmosted <- arrange(aggregate(EconomicDamage ~ Type, data, sum), desc(EconomicDama
ge))
cat('Events that has caused the most economic damages is', totalmosted[1,'Type'], 'wh
ich has caused $', format(totalmosted[1,'EconomicDamage'], big.mark=',', big.interval
=3L), 'in damages')
```

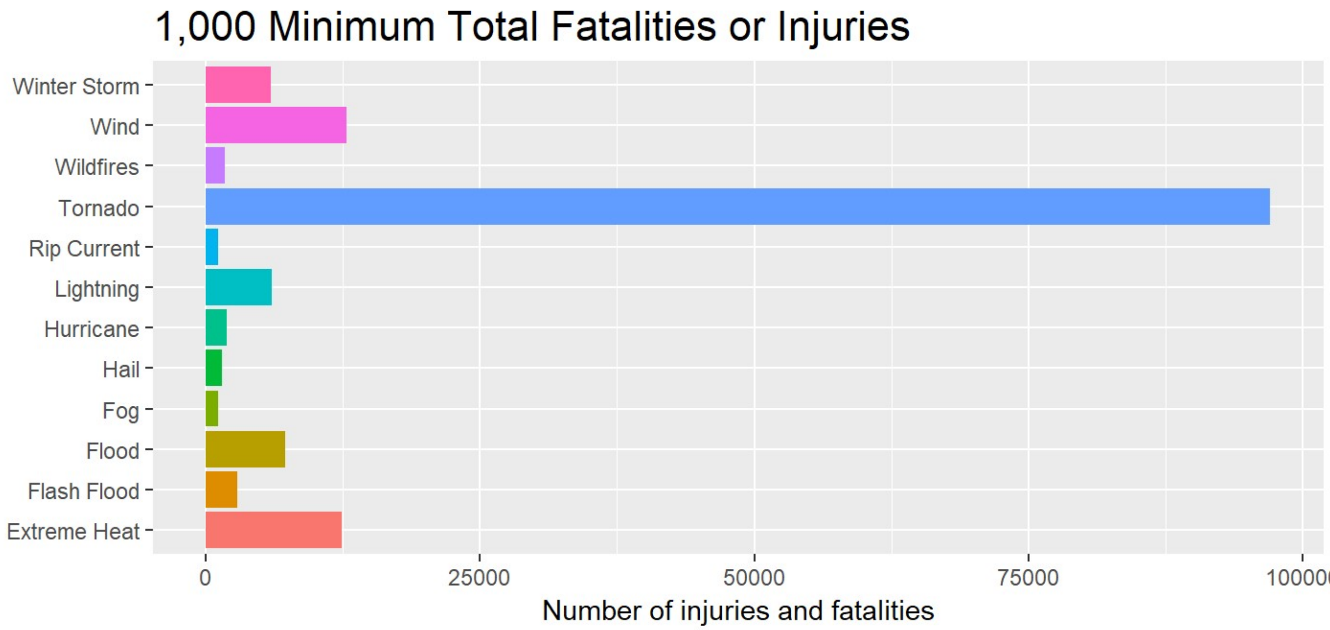
```
## Events that has caused the most economic damages is Flood which has caused $ 160,9
30,137,689 in damages
```

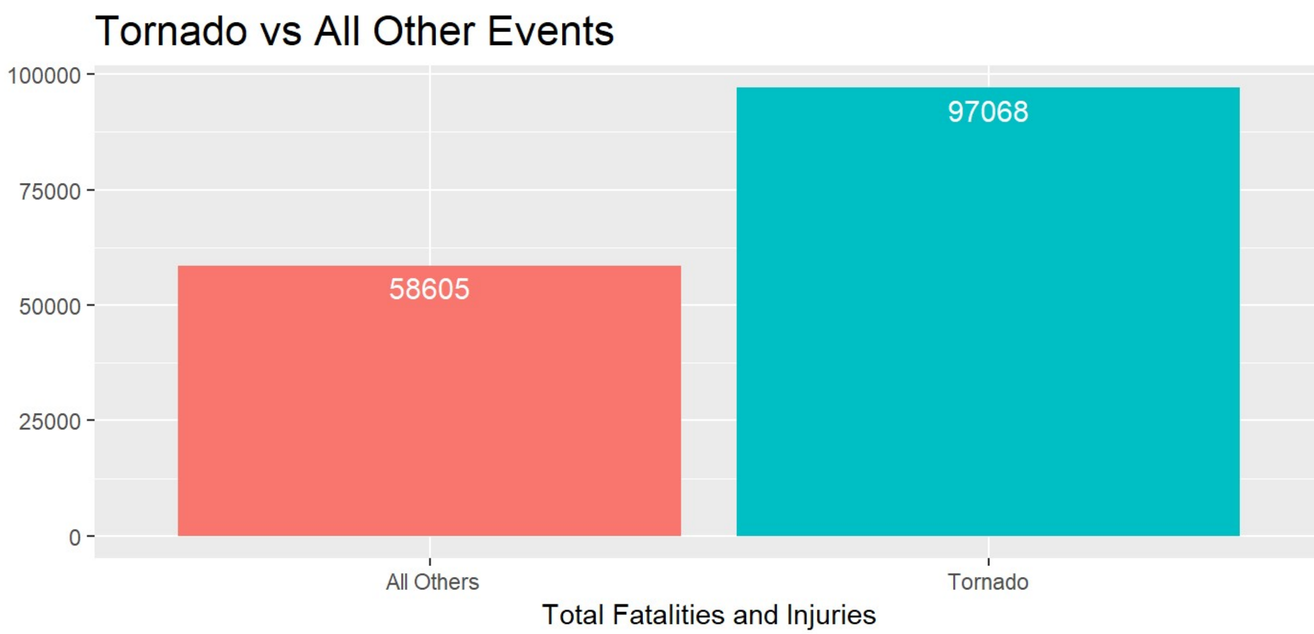
These three plots show the health results and economic results from various events. Due to the size of the plots necessary to display 31 different categories, the plots only show the type of events that meet a minimum threshold. The first plot details categories where the total from all events is over 1000 fatalities and injuries. As the plot shows, Tornado events are by far the largest cause of fatalities and injuries. In fact, Tornadoes have caused more fatalities and injuries than all other weather events combined, as is shown in the second plot. The final plot shows a breakdown of total economic damage from weather types with a minimum of one billion dollars in damages.

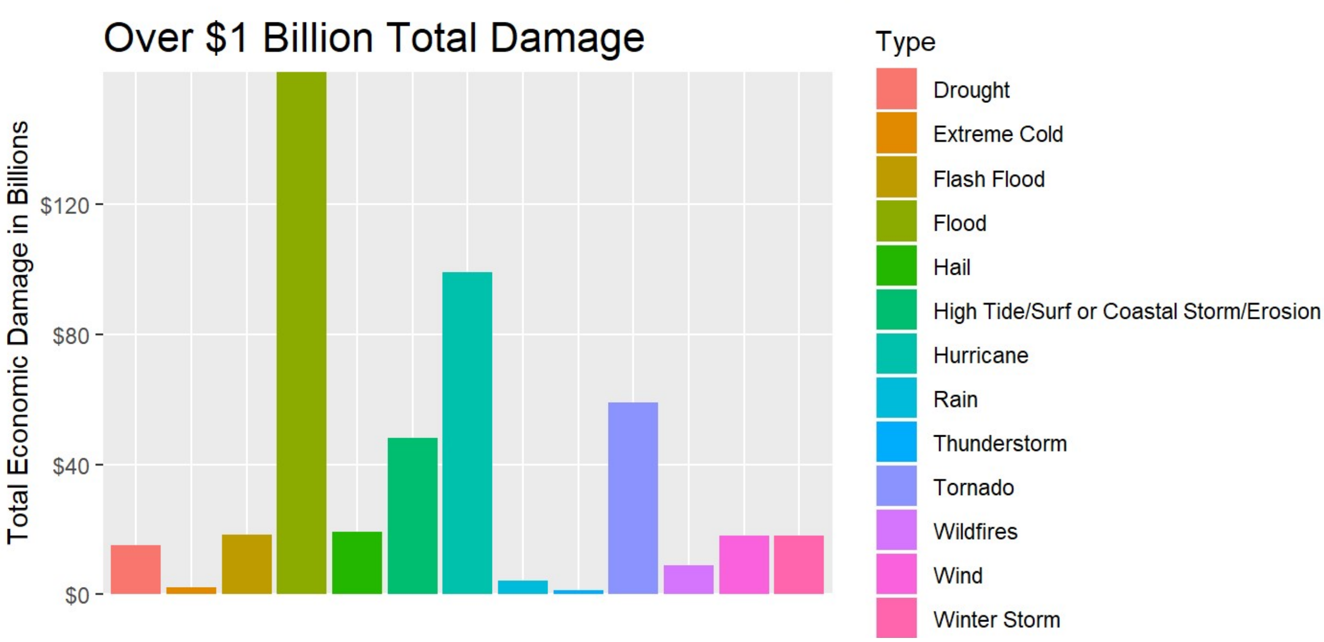
```
over999 <- filter(totalmostfi, TotalFI > 999)
totalfi <- ggplot(over999, aes(x=Type, y=TotalFI, fill=Type)) +
  geom_bar(stat='identity') + guides(fill=FALSE) + coord_flip() +
  ggtitle('1,000 Minimum Total Fatalities or Injuries') +
  xlab('') + ylab('Number of injuries and fatalities') +
  theme(plot.title=element_text(size=16))

tornadovsall <- totalmostfi
tornadovsall$Type <- replace(tornadovsall$Type, !grepl('Tornado', tornadovsall$Type),
' All Others')
tornadovsall <- aggregate(TotalFI ~ Type, tornadovsall, sum)
tornadofi <- ggplot(tornadovsall, aes(x=Type, y=TotalFI, fill=Type)) +
  geom_bar(stat='identity') + guides(fill=FALSE) +
  geom_text(aes(label = TotalFI), vjust = 1.5, color = 'white', size = 4) +
  ylab('') + xlab('Total Fatalities and Injuries') +
  ggtitle('Tornado vs All Other Events') +
  theme(plot.title = element_text(size=16))

over1b <- filter(totalmosted, EconomicDamage > 1e+09)
totaled <- ggplot(over1b, aes(x=Type, y=EconomicDamage, fill=Type)) +
  geom_bar(stat='identity') + xlab('') +
  ggtitle('Over $1 Billion Total Damage') +
  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank(),
  plot.title = element_text(size=16)) +
  scale_y_discrete(name='Total Economic Damage in Billions', limits=c(0,
  40e+09, 80e+09, 120e+09), labels=c('$0', '$40', '$80', '$120'))
plot_grid(totalfi, tornadofi, totaled, ncol=1)
```


1,000 Minimum Total Fatalities or Injuries
Winter Storm
Wind
Wildfires
Tornado
Rip Current
Lightning
Hurricane
Hail
Fog
Flood
Flash Flood
Extreme Heat
0
25000
50000
75000
100000
Number of injuries and fatalities


Tornado vs All Other Events
100000
75000
50000
25000
0
58605
97068
All Others
Tornado
Total Fatalities and Injuries


Over $1 Billion Total Damage
Total Economic Damage in Billions
$120
$80
$40
$0
Type
Drought
Extreme Cold
Flash Flood
Flood
Hail
High Tide/Surf or Coastal Storm/Erosion
Hurricane
Rain
Thunderstorm
Tornado
Wildfires
Wind
Winter Storm

3.C. Average Results Per Occurence

Which type of event causes the most fatalities per occurence?

```
meanfatal <- arrange(aggregate(Fatalities ~ Type, data, mean), desc(Fatalities))
meanfatal$Fatalities <- round(meanfatal[,'Fatalities'], 1)
cat('Events causing the most fatalities on average is', meanfatal[1,'Type'],
    'with', meanfatal[1,'Fatalities'], 'average fatalities per occurence')
```

```
## Events causing the most fatalities on average is Marine or Sea Weather with 2.1 av
erage fatalities per occurence
```

```
rm(meanfatal)
```

Which type of event causes the most injuries per occurence?

```
meaninjury <- arrange(aggregate(Injuries ~ Type, data, mean), desc(Injuries))
meaninjury$Injuries <- round(meaninjury[,'Injuries'], 1)
cat('Events causing the most injuries on average is', meaninjury[1,'Type'],
    'with', meaninjury[1,'Injuries'], 'average injuries per occurence')
```

```
## Events causing the most injuries on average is Tsunami with 6.2 average injuries p
er occurence
```

```
rm(meaninjury)
```

Which type of event causes the most fatalities and injuries per occurence?

```
meanfi <- arrange(aggregate(TotalFI ~ Type, data, mean), desc(TotalFI))
meanfi$TotalFI <- round(meanfi[,'TotalFI'], 1)
cat('Events causing the most fatalities and injuries on average is', meanfi[1,'Type
'], 'with', meanfi[1,'TotalFI'], 'average fatalities and injuries per occurence')
```

```
## Events causing the most fatalities and injuries on average is Tsunami with 7.8 ave
rage fatalities and injuries per occurence
```

Which type of event causes the most economic damage per occurence?

```
meaned <- arrange(aggregate(EconomicDamage ~ Type, data, mean), desc(EconomicDamage))
meaned$EconomicDamage <- trunc(meaned$EconomicDamage)
cat('Events causing the most economic damage on average is', meaned[1,'Type'], 'which
causes $', format(meaned[1,'EconomicDamage'], big.mark=',', big.interval=3L), 'averag
e economic damage per occurence')
```

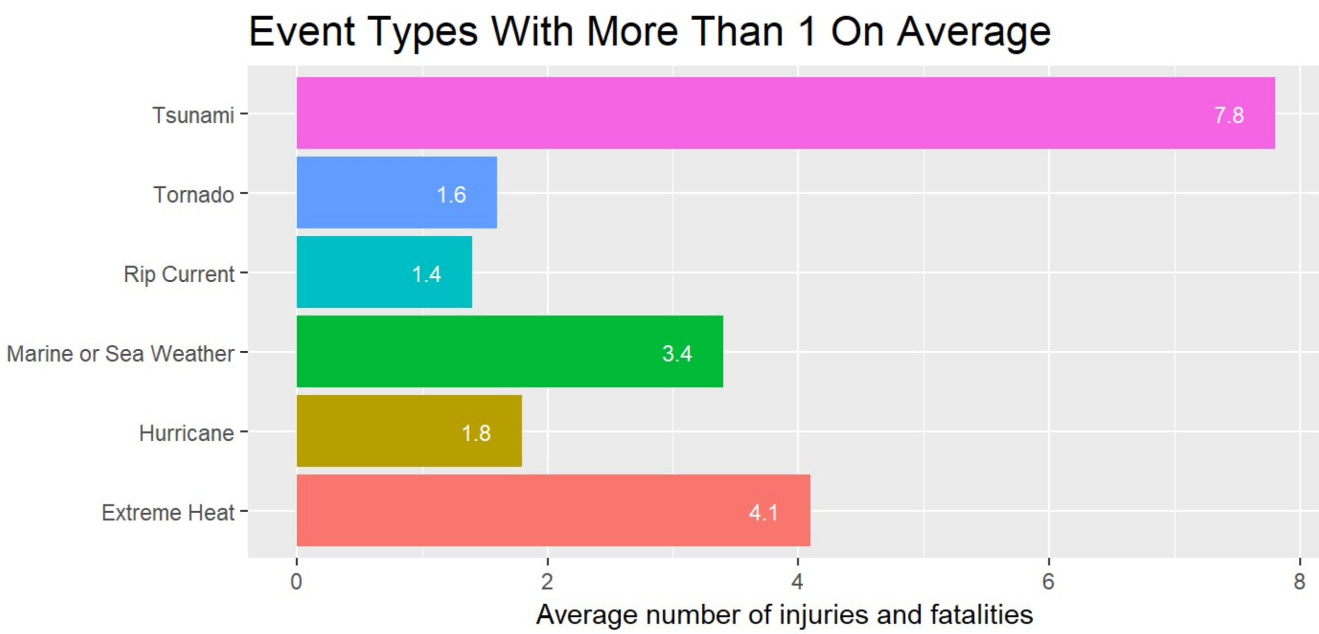
```
## Events causing the most economic damage on average is Hurricane which causes $ 93,
840,785 average economic damage per occurence
```

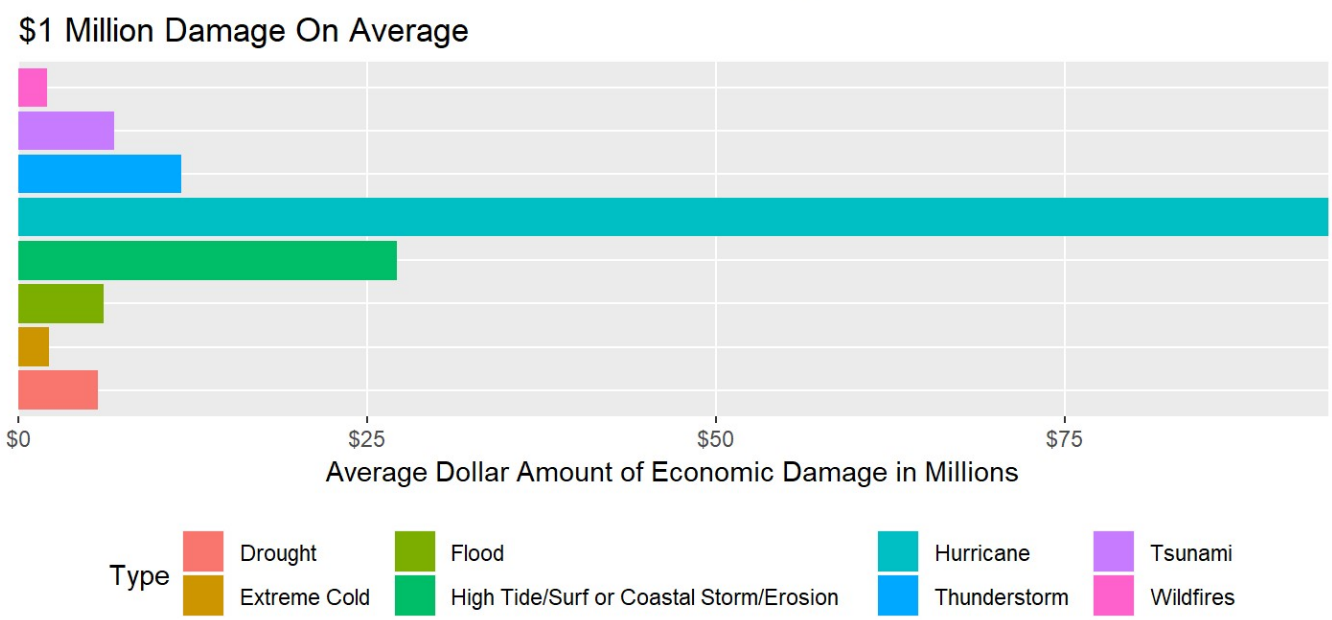
The first of the two following plots shows the average number of fatalities and injuries per occurence of weather types that average a minimum of 1 fatality/injury. The second one details types that average one million or more per occurence in economic damages.

```
over1 <- filter(meanfi, TotalFI >= 1)
meanfiplot <- ggplot(over1, aes(x=Type, y=TotalFI, fill=Type)) +
  geom_bar(stat='identity') + guides(fill=FALSE) + coord_flip() +
  ggtitle('Event Types With More Than 1 On Average') +
  xlab('') + ylab('Average number of injuries and fatalities') +
  geom_text(aes(label = TotalFI), hjust = 2, color = 'white', size = 3) +
  theme(plot.title = element_text(size=16))

over1m <- filter(meaned, EconomicDamage >= 1000000)
meanedplot <- ggplot(over1m, aes(x=Type, y=EconomicDamage, fill=Type)) + geom_bar(sta
t='identity') + xlab('') + ggtitle('$1 Million Damage On Average') + coord_flip() + t
heme(legend.position='bottom', axis.text.y=element_blank(), axis.ticks.y=element_blan
k()) + scale_y_discrete(name='Average Dollar Amount of Economic Damage in Millions',
limits=c(0,25000000, 50000000, 75000000), labels=c('$0', '$25', '$50', '$75'))

plot_grid(meanfiplot, meanedplot, ncol=1)
```


Event Types With More Than 1 On Average
Tsunami
Tornado
Rip Current
Marine or Sea Weather
Hurricane
Extreme Heat
7.8
1.6
1.4
3.4
1.8
4.1
0
2
4
6
8
Average number of injuries and fatalities


$1 Million Damage On Average
$0
$25
$50
$75
Average Dollar Amount of Economic Damage in Millions
Type
Drought
Extreme Cold
Flood
High Tide/Surf or Coastal Storm/Erosion
Hurricane
Thunderstorm
Tsunami
Wildfires