Reproducible ResearchAssignment 2

Adlin Fisher Abell Ball

2023-06-27

The purpose of this document is to provide an analysis of the data provided which was taken from the National Oceanic and Atmospheric Administration and contains information from the 1950 through to November of 2011. In 1996 the NOAA began recording any event types that caused injury to persons or damage to property. See the NOAA Website for more information. Then for the sake of consistency of the data collection process, this analysis will use the data available from the year 1996 and onward. We will begin by preparing the data for analysis:

Data Processing

The first step is naturally to load in the data and take the section that we want for analysis.

```r
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
damage = read.csv(bzfile("repdata_data_StormData.csv.bz2"))
damage$BGN_DATE = strptime(damage$BGN_DATE, tz='', '%m/%d/%Y %H:%M:%OS')
damage96 = damage[damage$BGN_DATE > strptime('1996-01-01', '%F'),
                  1:length(colnames(damage))]
remove(damage)
```

Now, to begin we must prepare the data for calculating the financial impact. For this, I drop all the irrelevant rows in order to save time in operating on the data. Note that the `PROPDMGEXP` and `CROPDMGEXP` columns contain a factor that encodes the amount of damage in the `PROPDMG` and `CROPDMG` columns respectively. In this scheme, `K` indicates that the column damages are in thousands of dollars, `M` indicates millions, and `B` indicates billions. There are other values in the encoding columns, but since these are not accounted for in the data documentation, the rows associated with these entries are discarded.

```
# First, for efficiency's sake I select only the relevant columns for this
# analysis.
columns = c('BGN_DATE', 'PROPDMG', 'PROPDMGEXP', 'CROPDMG', 'CROPDMGEXP',
            'EVTYPE', 'REFNUM')
QuantDam = damage96[columns]

#Next, I will discard all of the data that does not have its costs properly
# prepared.
QuantDam = QuantDam[(QuantDam$PROPDMG == 0)|
                      (QuantDam$PROPDMGEXP %in% c('K', 'M', 'B', '')),
                    1:length(colnames(QuantDam)) ]
QuantDam = QuantDam[(QuantDam$CROPDMG == 0)|
                      (QuantDam$CROPDMGEXP %in% c('K', 'M', 'B', '')),
                    1:length(colnames(QuantDam)) ]

totaldamage = rep(0, dim(QuantDam)[1])
ttldmg2023 = rep(0, dim(QuantDam)[1])

# Below I combine the total damages to both crops and property while calculating
# the costs.

pks = (QuantDam$PROPDMGEXP == 'K' | QuantDam$PROPDMGEXP == 'k')
totaldamage = totaldamage + QuantDam$PROPDMG * pks * 10^3
remove(pks)

pms = (QuantDam$PROPDMGEXP == 'M')
totaldamage = totaldamage + QuantDam$PROPDMG * pms * 10^6
remove(pms)

pbs = (QuantDam$PROPDMGEXP == 'B')
totaldamage = totaldamage + QuantDam$PROPDMG * pbs * 10^9
remove(pbs)

cks = (QuantDam$CROPDMGEXP == 'K' | QuantDam$CROPDMGEXP == 'k')
totaldamage = totaldamage + QuantDam$CROPDMG * cks * 10^3
remove(cks)

cms = (QuantDam$CROPDMGEXP == 'M')
totaldamage = totaldamage + QuantDam$CROPDMG * cms * 10^6
remove(cms)

cbs = (QuantDam$CROPDMGEXP == 'B')
totaldamage = totaldamage + QuantDam$CROPDMG * cbs * 10^9
remove(cbs)
```

Since the value of a dollar is not constant accross time, I also wish to account for inflation in my analysis of economic damages. To do this, I used the Bureau of Labor Statistics CPI Database (accessed June 12, 2023) to get the inflation index. The calculations to convert the damages to May 2023 dollars are below:

```
# First, I initialize a couple vectors for our calculations.
eventCPI = rep(0, dim(QuantDam)[1])
ttldmg2023 = rep(0, dim(QuantDam)[1])
```

```r
# Now I read in the CPI data.
QuantDam = cbind(QuantDam, TOTALDMG = totaldamage)
CPI = read.csv('CPI.csv')
```

Now I convert the times to date-time objects. Then I take the weather event beginning dates and convert them to the first of their month in order to match the CPI for that date to the data from the CPI database. Then I remove superfluous vectors since they are a non-trivial amount of storage.

```r
CPI$Label = strptime(paste(CPI$Label, '01'), '%Y %b %d')
CPIperiod = strptime(paste(format(QuantDam$BGN_DATE, "%Y-%m"),
                           '01', sep= '-'), format = '%Y-%m-%d')
periodind = match(CPIperiod, CPI$Label)
for (i in 1:length(CPIperiod)) {
  eventCPI[i] = CPI$Value[periodind[i]]
}
remove(CPIperiod, periodind)

# Lastly, take the most current CPI at time of writing and calculate the total
# total damages for each event in May 2023 Dollars.
NowCPI = CPI$Value[match(strptime('2023-05-01', '%Y-%m-%d'), CPI$Label)]
ttldmg2023 = (NowCPI/eventCPI) * totaldamage
QuantDam = cbind(QuantDam, TOTAL2023DMG = as.numeric(ttldmg2023))

remove(totaldamage, CPI, NowCPI)
```

Next, since the primary purpose of this analysis is to evaluate events by their type, (`EVTYPE` in the data itself), we need to address some problems with this column. Firstly, the events appear to be inconsistently labelled, with capitalization and spacing varying inconsistently across labels, causing R to not recognize them as the same type of event. Secondly, there appear to be tags that are used either in the case of having one type of event *or* another, which are also used as labels for events that are *combinations* of those events. By reviewing the events with the highest amount of economic damages, I've selected some event tags below to relabel to broader event tags in which they fit.

```r
# By using the `grep` function, I search each event label for occurrences of the
# first argument and change the label to a more inclusive one.

QuantDam$EVTYPE[grep('hurricane', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = 'HURRICANE/TYPHOON'
QuantDam$EVTYPE[grep('typhoon', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = 'HURRICANE/TYPHOON'
QuantDam$EVTYPE[grep('damaging freeze', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = 'DAMAGING FREEZE'
QuantDam$EVTYPE[grep('flood', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = 'FLOOD'
QuantDam$EVTYPE[grep('fld', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = 'FLOOD'
QuantDam$EVTYPE[grep('thunderstorm', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = "THUNDERSTORM"
QuantDam$EVTYPE[grep('tstm', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = "THUNDERSTORM"
QuantDam$EVTYPE[grep('freeze', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = "FROST/FREEZE"
QuantDam$EVTYPE[grep('frost', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = "FROST/FREEZE"
QuantDam$EVTYPE[grep('storm surge', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = 'STORM SURGE/TIDE'
QuantDam$EVTYPE[grep('high tide', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = 'STORM SURGE/TIDE'
QuantDam$EVTYPE[grep('wild fire', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = 'WILD/FOREST FIRE'
QuantDam$EVTYPE[grep('wildfire', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = 'WILD/FOREST FIRE'
coldtags = unique(QuantDam$EVTYPE)[grep('cold',
                                        unique(QuantDam$EVTYPE),
                                        ignore.case = TRUE)]
coldtags
```

```
##  [1] "EXTREME COLD"                 "Unseasonable Cold"
##  [3] "Record Cold"                  "Extreme Cold"
##  [5] "Excessive Cold"               "Extended Cold"
##  [7] "COLD"                         "Cold"
##  [9] "Cold Temperature"             "COLD AND SNOW"
## [11] "UNSEASONABLY COLD"            "Prolong Cold"
## [13] "RECORD COLD"                  "PROLONG COLD"
## [15] "COLD TEMPERATURES"            "COLD WIND CHILL TEMPERATURES"
## [17] "RECORD  COLD"                 "UNUSUALLY COLD"
## [19] "COLD WEATHER"                 "EXTREME COLD/WIND CHILL"
## [21] "COLD/WIND CHILL"
```

As we can see above, there are multiple redundant tags for cold whether or windchill, but not every event descriptor that contains the word "cold" is equivalent. Since all of these event titles except "COLD AND SNOW" appear to be used to describe events that are characterized by cold tempuratures, windchill, or some combination of the two, it is intuitive to combine them. Note that since cold and snow includes snow in its characteristic conditions rather than simply cold temperatures, it will be excluded.

```
coldtags = coldtags[coldtags != 'COLD AND SNOW']

for (i in 1:length(coldtags)) {
  QuantDam$EVTYPE[grep(coldtags[i], QuantDam$EVTYPE,
                     ignore.case = TRUE)] = "COLD/WIND CHILL"
}
QuantDam$EVTYPE[grep('windchill', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = "COLD/WIND CHILL"

QuantDam$EVTYPE[grep('wind chill', QuantDam$EVTYPE,
                     ignore.case = TRUE)] = "COLD/WIND CHILL"

# Remove entries that missused the EVTYPE entry.
QuantDam = QuantDam[grep('summary', QuantDam$EVTYPE, ignore.case = TRUE,
                         invert = TRUE),]

# Now I consolidate the damages by the aggregate function to select the top 15
#

totaldamages = aggregate(QuantDam$TOTAL2023DMG, list(QuantDam$EVTYPE),
                         FUN = sum)
```

```
names(totaldamages) = c('evtype', 'dmg')
top15 = order(totaldamages$dmg, decreasing = TRUE)[1:15]
QuantDam1 = QuantDam[QuantDam$EVTYPE %in% totaldamages$evtype[top15] &
                       QuantDam$TOTALDMG != 0, ]

#Below I calculate some summary statistics for use in the final analysis.

hurmed = median(QuantDam1$TOTAL2023DMG[QuantDam1$EVTYPE == 'HURRICANE/TYPHOON'])
drtmed = median(QuantDam1$TOTAL2023DMG[QuantDam1$EVTYPE == 'DROUGHT'])
fstmed = median(QuantDam1$TOTAL2023DMG[QuantDam1$EVTYPE == 'FROST/FREEZE'])
napadmg = format(max(QuantDam1$TOTAL2023DMG[QuantDam1$EVTYPE == 'FLOOD']),
                 big.mark = ',', scientific = FALSE)
fldmed = format(median(QuantDam1$TOTAL2023DMG[QuantDam1$EVTYPE == 'FLOOD']),
                big.mark = ',')

# Tossing objects that are no longer needed
remove(eventCPI, totaldamages, QuantDam)
```

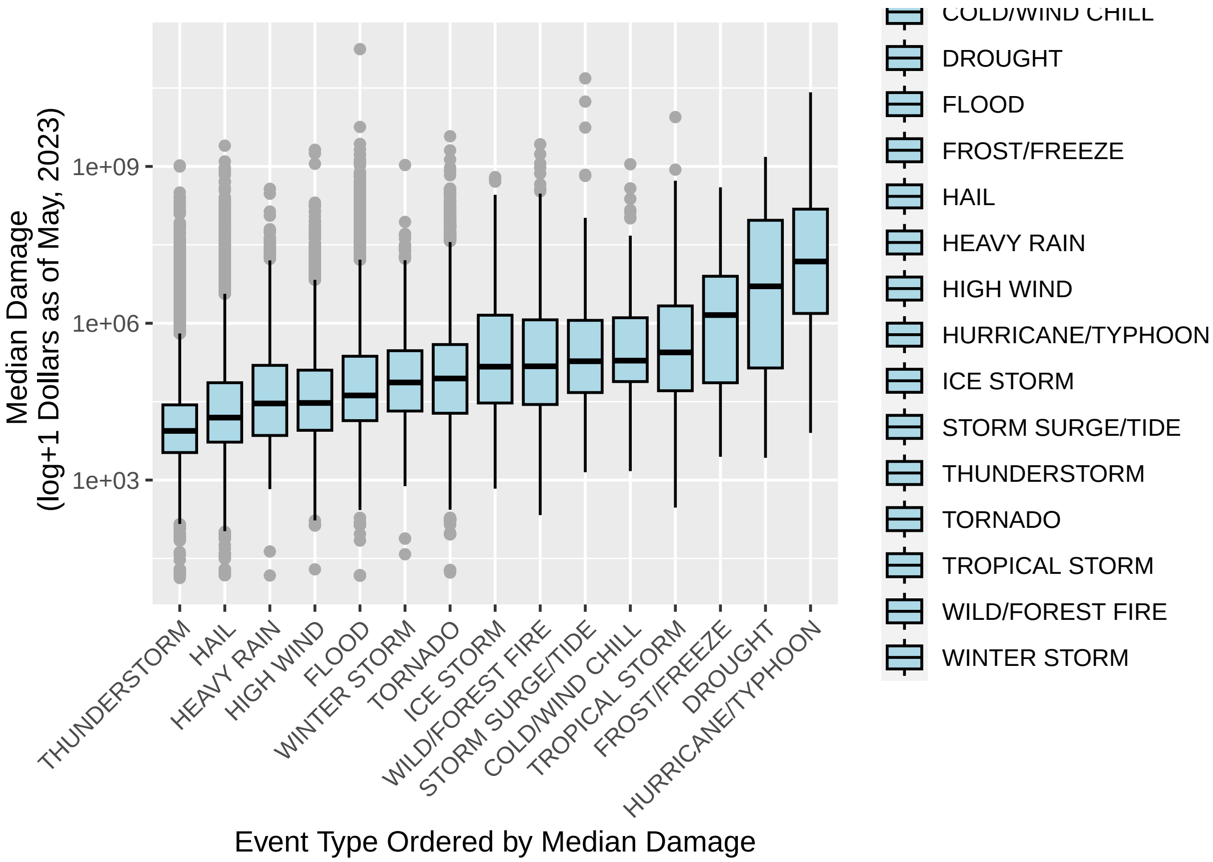
Now for the human damage of these events. For this we will once again drop the irrelevant categories. For consistency's sake, we will consolidate the event type tags the same way as we did previously. Not surprisingly, the events that cause the most economic damages have a large amount of overlap with those that caused the most casualties. Note that the table for casualties drops the total casualties column and reformats the data below for plotting.

```
HumDmg = damage96[, c('REFNUM', 'EVTYPE', 'INJURIES', 'FATALITIES')]
HumDmg$EVTYPE[grep('hurricane', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = 'HURRICANE/TYPHOON'
HumDmg$EVTYPE[grep('typhoon', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = 'HURRICANE/TYPHOON'
HumDmg$EVTYPE[grep('damaging freeze', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = 'DAMAGING FREEZE'
HumDmg$EVTYPE[grep('flood', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = 'FLOOD'
HumDmg$EVTYPE[grep('fld', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = 'FLOOD'
HumDmg$EVTYPE[grep('thunderstorm', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = "THUNDERSTORM"
HumDmg$EVTYPE[grep('tstm', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = "THUNDERSTORM"
HumDmg$EVTYPE[grep('freeze', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = "FROST/FREEZE"
HumDmg$EVTYPE[grep('frost', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = "FROST/FREEZE"
HumDmg$EVTYPE[grep('storm surge', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = 'STORM SURGE/TIDE'
HumDmg$EVTYPE[grep('high tide', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = 'STORM SURGE/TIDE'
HumDmg$EVTYPE[grep('wild fire', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = 'WILD/FOREST FIRE'
HumDmg$EVTYPE[grep('wildfire', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = 'WILD/FOREST FIRE'
coldtags = unique(HumDmg$EVTYPE)[grep('cold',
                                        unique(HumDmg$EVTYPE),
```

```r
                                        ignore.case = TRUE)]
coldtags = coldtags[coldtags != 'COLD AND SNOW']

for (i in 1:length(coldtags)) {
  HumDmg$EVTYPE[grep(coldtags[i], HumDmg$EVTYPE,
                     ignore.case = TRUE)] = "COLD/WIND CHILL"
}
HumDmg$EVTYPE[grep('windchill', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = "COLD/WIND CHILL"

HumDmg$EVTYPE[grep('wind chill', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = "COLD/WIND CHILL"
HumDmg$EVTYPE[grep('rip currents', HumDmg$EVTYPE,
                     ignore.case = TRUE)] = "RIP CURRENT"
HumDmg$EVTYPE[HumDmg$EVTYPE != "COLD/WIND CHILL"][grep('wind',
                   HumDmg$EVTYPE[HumDmg$EVTYPE != "COLD/WIND CHILL"],
                   ignore.case = TRUE)]= 'WIND'
HumDmg$EVTYPE[HumDmg$EVTYPE == 'WND' ]= 'WIND'
HumDmg$EVTYPE[grep('record temperature', HumDmg$EVTYPE,
                   ignore.case = TRUE)] = 'EXCESSIVE HEAT'
HumDmg$EVTYPE[grep('record temperature', HumDmg$EVTYPE,
                   ignore.case = TRUE)] = 'EXCESSIVE HEAT'
HumDmg$EVTYPE[grep('heat', HumDmg$EVTYPE,
                   ignore.case = TRUE)] = 'EXCESSIVE HEAT'
NotWet = grep('wet', HumDmg$EVTYPE, ignore.case = TRUE, invert = TRUE)

HumDmg$EVTYPE[NotWet][grep('warm', HumDmg$EVTYPE[NotWet],
                           ignore.case = TRUE)] = 'EXCESSIVE HEAT'
remove(NotWet)

# Remove entries that missused the EVTYPE entry.
HumDmg = HumDmg[grep('summary', HumDmg$EVTYPE, ignore.case = TRUE,
                     invert = TRUE),]

CASUALTIES = HumDmg$INJURIES + HumDmg$FATALITIES

HumDmg = cbind(HumDmg, CASUALTIES)

totalcasualties = aggregate(HumDmg[,c('INJURIES', 'FATALITIES', 'CASUALTIES')],
                            list(HumDmg$EVTYPE), sum)

#holding on for troubleshooting
#totalcasualties$EVTYPE[order(totalcasualties$CASUALTIES, decreasing = TRUE)]

top15 = order(totalcasualties$CASUALTIES, decreasing = TRUE)[1:15]
colnames(totalcasualties) = c('EVTYPE', 'INJURIES', 'FATALITIES', 'CASUALTIES')
top15casualties = totalcasualties$EVTYPE[top15]

HumDmg = HumDmg[HumDmg$CASUALTIES != 0,c('EVTYPE', 'INJURIES', 'FATALITIES')]
graphhmdmg = pivot_longer(HumDmg, c('FATALITIES', 'INJURIES'))
graphhmdmg = graphhmdmg[graphhmdmg$EVTYPE %in% top15casualties,]
graphhmdmg = graphhmdmg[graphhmdmg$value != 0,]
```

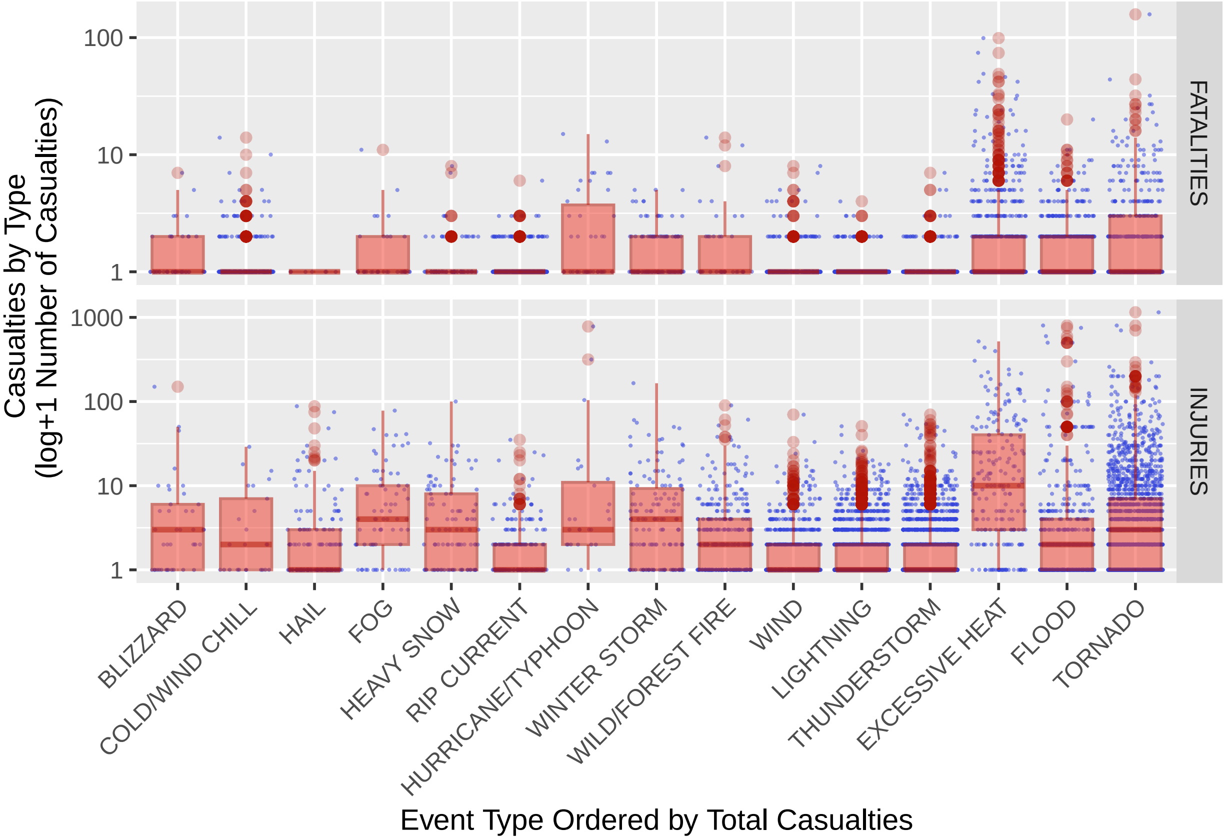
Results

```r
ggplot(data = QuantDam1) +
  geom_boxplot(mapping = aes(x = reorder(EVTYPE , TOTAL2023DMG+1, median),
                             TOTAL2023DMG, fill = EVTYPE,
                             color = EVTYPE),
               outlier.color = 'dark gray') +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
   scale_fill_manual(values = rep("light blue", 15)) +
   scale_color_manual(values = rep('black',15)) +
  scale_y_log10(name = "Median Damage \n (log+1 Dollars as of May, 2023)") +
  scale_x_discrete(name = 'Event Type Ordered by Median Damage')
```



Note that the above graph is of only the events that did some amount of damage, with events that did no damage excluded. With that in mind, we can see that the event category that has the highest median damages is hurricanes and typhoons with a median of $15,196,761. The categories with the second and third highest median damages are drought, with median damages of $5,096,849, and freezing/frosts, with a median of $1,434,424. This graph also reveals a number of outlying points. The most prominent is the event with the single highest damage, which comes from the flooding category. This event's damages total to $175,056,031,385, the highest of any single event. Compare this with the flood category's median of $41,544.09. This event is the Napa Valley flooding of 2005.

```r
ggplot(data = graphhmdmg[graphhmdmg$value != 0,]) +
   geom_jitter(mapping = aes(x = reorder(EVTYPE, value, sum),
               y = value), size = .05, color = '#3545db88') +
  geom_boxplot(mapping = aes(x = reorder(EVTYPE , value, sum),
               y = value), color = '#b3140588', fill = '#f0403088',
               outlier.colour = '#b3140544') +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
   scale_fill_manual(values = rep(c("light blue", 'red'), 15)) +
   scale_color_manual(values = rep('black',30)) +
  scale_y_log10(name = "Casualties by Type \n (log+1 Number of Casualties)") +
  scale_x_discrete(name = 'Event Type Ordered by Total Casualties') +
  facet_grid(name ~ ., scales = 'free_y')
```



Due to the fact that boxplots alone don't appear to give a full view of these data, I plotted both a box plot and the individual points on the above graph. Once again, I've omitted events where no damage was recorded. We can see above that while tornadoes, flood, and abnormal heat lead all other event types in terms of total casualties, they are not as a rule the events with the highest mean damages by either category. While tornadoes have the highest combined casualties of any event, the median number of injuries for that category is 3, compared with a median of 10 injuries in the category of excessive heat events. Note also that both fog and winter storm events have higher median injuries at 4 as well. The medians in the category of fatalities are notable in that they are all the same, with a median of 1. Therefore, excessive heat is one of the most hazardous weather event types for human health, along with floods and tornadoes. While these events don't consistently do damage more than other event types, the amount of damage that they do incur when they do have impacts on humans total to a higher amount than any other events.