# Assessment of economic and public health consequences of adverse weather events in the United States

J. Varberg

2/26/2022

## Synopsis

This report uses data from the National Oceanic and Atmospheric Administration (NOAA) Storm Database to examine the economic and public health impacts that different types of weather events have in the United States.

## Data Processing

The raw data were obtained here and read into R using the `read_csv` function from the `readr` package, which can directly handle reading/import of zipped files.

```
checkFile <- file.exists("./data/repdata-data-StormData.csv.bz2")

if (!checkFile) {
    download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2",
        destfile = "./data/repdata-data-StormData.csv.bz2")
}
rawData <- read_csv("./data/repdata-data-StormData.csv.bz2")
glimpse(rawData)
```

```
## Rows: 902,297
## Columns: 37
## $ STATE__    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ BGN_DATE   <chr> "4/18/1950 0:00:00", "4/18/1950 0:00:00", "2/20/1951 0:00:0~
## $ BGN_TIME   <chr> "0130", "0145", "1600", "0900", "1500", "2000", "0100", "09~
## $ TIME_ZONE  <chr> "CST", "CST", "CST", "CST", "CST", "CST", "CST", "CST", "CS~
## $ COUNTY     <dbl> 97, 3, 57, 89, 43, 77, 9, 123, 125, 57, 43, 9, 73, 49, 107,~
## $ COUNTYNAME <chr> "MOBILE", "BALDWIN", "FAYETTE", "MADISON", "CULLMAN", "LAUD~
## $ STATE      <chr> "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL",~
## $ EVTYPE     <chr> "TORNADO", "TORNADO", "TORNADO", "TORNADO", "TORNADO", "TOR~
## $ BGN_RANGE  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ BGN_AZI    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BGN_LOCATI <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ END_DATE   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ END_TIME   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ COUNTY_END <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ COUNTYENDN <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ END_RANGE  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

```
## $ END_AZI    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ END_LOCATI <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ LENGTH     <dbl> 14.0, 2.0, 0.1, 0.0, 0.0, 1.5, 1.5, 0.0, 3.3, 2.3, 1.3, 4.7~
## $ WIDTH      <dbl> 100, 150, 123, 100, 150, 177, 33, 33, 100, 100, 400, 400, 2~
## $ F          <dbl> 3, 2, 2, 2, 2, 2, 2, 1, 3, 3, 1, 1, 3, 3, 3, 4, 1, 1, 1, 1,~
## $ MAG        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ FATALITIES <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 4, 0, 0, 0, 0,~
## $ INJURIES   <dbl> 15, 0, 2, 2, 2, 6, 1, 0, 14, 0, 3, 3, 26, 12, 6, 50, 2, 0, ~
## $ PROPDMG    <dbl> 25.0, 2.5, 25.0, 2.5, 2.5, 2.5, 2.5, 2.5, 25.0, 25.0, 2.5, ~
## $ PROPDMGEXP <chr> "K", "K", "K", "K", "K", "K", "K", "K", "K", "K", "M", "M",~
## $ CROPDMG    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ CROPDMGEXP <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ WFO        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ STATEOFFIC <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ ZONENAMES  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ LATITUDE   <dbl> 3040, 3042, 3340, 3458, 3412, 3450, 3405, 3255, 3334, 3336,~
## $ LONGITUDE  <dbl> 8812, 8755, 8742, 8626, 8642, 8748, 8631, 8558, 8740, 8738,~
## $ LATITUDE_E <dbl> 3051, 0, 0, 0, 0, 0, 0, 0, 3336, 3337, 3402, 3404, 0, 3432,~
## $ LONGITUDE_ <dbl> 8806, 0, 0, 0, 0, 0, 0, 0, 8738, 8737, 8644, 8640, 0, 8540,~
## $ REMARKS    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ REFNUM     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ~
```

Our task for analysis is to answer the following two questions:

1. Across the United States, which types of events (as indicated in the `EVTYPE` variable) are most harmful with respect to population health?
2. Across the United States, which types of events have the greatest economic consequences?

To answer these questions, we will be most interested in examining all of the event types for their values in the columns for fatalities, injuries, property damage, and crop damage. For the property and crop damage, we will also need the values stored in `PROPDMGEXP` and `CROPDMGEXP`, which encode information about the multiplier for values in the `PROPDMG` and `CROPDMG` columns.

First, let's look at the entered values for `EVTYPE`.

```r
sample(unique(rawData$EVTYPE), 50)
```

```
##  [1] "HEAVY RAIN/SNOW"          "THUNDERSTORM WIND 59 MPH"
##  [3] "THUNDERSTORM WINDS G"     "Monthly Rainfall"
##  [5] "RECORD WARM TEMPS."       "Summary June 6"
##  [7] "Hot and Dry"              "Summary of May 31 pm"
##  [9] "Summary of June 6"        "SNOW/RAIN"
## [11] "WINTER MIX"               "SNOW/RAIN/SLEET"
## [13] "GUSTY WIND/HAIL"          "Summary September 20"
## [15] "MONTHLY SNOWFALL"         "COLD/WIND CHILL"
## [17] "Extreme Cold"             "SNOW FREEZING RAIN"
## [19] "EARLY FREEZE"             "HEAVY SNOW"
## [21] "Snow Squalls"             "LATE SNOW"
## [23] "HIGH WINDS 82"            "Light Snow"
## [25] "RECORD/EXCESSIVE HEAT"    "THUNDERSTORM WINDS."
## [27] "LARGE WALL CLOUD"         "SNOW SQUALL"
## [29] "Ice/Snow"                 "LOW TEMPERATURE RECORD"
## [31] "VOLCANIC ASH"             "Light Snow/Flurries"
```

```
## [33] "RAIN"                        "PROLONG COLD"
## [35] "THUNERSTORM WINDS"           "FLASH FLOOD/ FLOOD"
## [37] "EXCESSIVE HEAT"              "Mudslide"
## [39] "HIGH SEAS"                   "TSTM WIND AND LIGHTNING"
## [41] "ICE STORM AND SNOW"          "Summary of July 22"
## [43] "LANDSLIDE"                   "BLIZZARD AND HEAVY SNOW"
## [45] "STREET FLOOD"                "URBAN/STREET FLOODING"
## [47] "HIGH  WINDS"                 "Summary August 10"
## [49] "Early Frost"                 "RIVER FLOOD"
```

The entries for EVTYPE are messy - there is a mix of upper and lower class characters used, typos, event types that are combined, etc. We can do a first pass clean up by converting to all upper case to allow combining of types that are similar but coded in different case.

```
length(unique(rawData$EVTYPE))
```

```
## [1] 977
```

```
length(unique(toupper(rawData$EVTYPE)))
```

```
## [1] 890
```

This quick fix resolves 87 coding errors. According to the NOAA documentation, there are only 55 specific event types that should be entered into the database. We will try to match the entered event type with the most relevant allowed event type. First, let's create a vector of the allowed event types, and visually inspect the EVTYPE entries.

```
allowed_events <- toupper(c("Astronomical Low Tide", "Avalanche",
    "Blizzard", "Coastal Flood", "Cold/Wind Chill", "Debris Flow",
    "Dense Fog", "Dense Smoke", "Drought", "Dust Devil",
    "Dust Storm", "Excessive Heat", "Extreme Cold/Wind Chill",
    "Flash Flood", "Flood", "Frost/Freeze", "Funnel Cloud",
    "Freezing Fog", "Hail", "Heat", "Heavy Rain", "Heavy Snow",
    "High Surf", "High Wind", "Hurricane (Typhoon)", "Ice Storm",
    "Lake-Effect Snow", "Lakeshore Flood", "Lightning",
    "Marine Dense Fog", "Marine Hail", "Marine Heavy Freezing Spray",
    "Marine High Wind", "Marine Hurricane/Typhoon", "Marine Lightning",
    "Marine Strong Wind", "Marine Thunderstorm Wind", "Marine Tropical Depression",
    "Marine Tropical Storm", "Rip Current", "Seiche", "Sleet",
    "Sneaker Wave", "Storm Surge/Tide", "Strong Wind", "Thunderstorm Wind",
    "Tornado", "Tropical Depression", "Tropical Storm",
    "Tsunami", "Volcanic Ash", "Waterspout", "Wildfire",
    "Winter Storm", "Winter Weather"))

head(unique(toupper(rawData$EVTYPE)), n = 25)
```

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

```
## [11] "RECORD COLD"                   "HURRICANE ERIN"
## [13] "HURRICANE OPAL"                "HEAVY RAIN"
## [15] "LIGHTNING"                     "THUNDERSTORM WIND"
## [17] "DENSE FOG"                     "RIP CURRENT"
## [19] "THUNDERSTORM WINS"             "FLASH FLOOD"
## [21] "FLASH FLOODING"                "HIGH WINDS"
## [23] "FUNNEL CLOUD"                  "TORNADO F0"
## [25] "THUNDERSTORM WINDS LIGHTNING"
```

One of the things we can see is that `Thunderstorm` is often encoded in shorthand, as `TSTM`. This can easily be replaced with `mutate` and the `str_replace` function. While tidying up the data, let's also select just the columns of interest that have health or economic impact values. Then, we'll convert the `EVTYPE` column values to upper case and date columns from character to date types. We'll also add a column coding whether or not the value is one of the allowed event types.

During this step, we will also convert the values in the `PROPDMG` and `CROPDMG` fields to their full values by multiplying by the values in the corresponding `EXP` columns (see this for explanation of EXP values):

```r
tidyData <- rawData %>%
    select(BGN_DATE, COUNTY, COUNTYNAME, STATE, EVTYPE,
        END_DATE, FATALITIES, INJURIES, PROPDMG, PROPDMGEXP,
        CROPDMG, CROPDMGEXP, REMARKS) %>%
    mutate(BGN_DATE = mdy_hms(BGN_DATE), END_DATE = mdy_hms(END_DATE),
        EVTYPE = toupper(EVTYPE), EVTYPE = str_replace(EVTYPE,
            "TSTM", "THUNDERSTORM"), ALLOWED = if_else(EVTYPE %in%
            allowed_events, true = "ALLOWED", false = "NOT_ALLOWED"),
        PROPDMG = case_when(PROPDMGEXP == "H" | PROPDMGEXP ==
            "h" ~ PROPDMG * 100, PROPDMGEXP == "K" ~ PROPDMG *
            1000, PROPDMGEXP == "M" | PROPDMGEXP == "m" ~
            PROPDMG * 1e+06, PROPDMGEXP == "B" ~ PROPDMG *
            1e+09, is.numeric(PROPDMGEXP) ~ PROPDMG * 10,
            TRUE ~ PROPDMG), CROPDMG = case_when(CROPDMGEXP ==
            "H" | CROPDMGEXP == "h" ~ CROPDMG * 100, CROPDMGEXP ==
            "K" ~ CROPDMG * 1000, CROPDMGEXP == "M" | CROPDMGEXP ==
            "m" ~ CROPDMG * 1e+06, CROPDMGEXP == "B" ~ CROPDMG *
            1e+09, is.numeric(CROPDMGEXP) ~ CROPDMG * 10,
            TRUE ~ CROPDMG))
```

Now, let's see how many events are allowed vs. not allowed event types:

```r
with(tidyData, table(ALLOWED))
```

```
## ALLOWED
##     ALLOWED NOT_ALLOWED
##      861476       40821
```

There are still quite a few entries (~6%) are not properly classified for event type. We only care for ones that have a public health or economic impact, so let's filter for those and then see how many need to be fixed.

```r
# add columns coding if there was health or economic
# damages, filter to keep only rows with at least one
# type of damages
```

```r
tidyDataDamages <- tidyData %>%
    mutate(HealthImpact = if_else(condition = FATALITIES >
        0 | INJURIES > 0, true = TRUE, false = FALSE), EconImpact = if_else(condition = PROPDMG >
        0 | CROPDMG > 0, true = TRUE, false = FALSE)) %>%
    filter(HealthImpact == TRUE | EconImpact == TRUE)
with(tidyDataDamages, table(ALLOWED))
```

```
## ALLOWED
##     ALLOWED NOT_ALLOWED
##      236219       18414
```

Removing to only keep events with health or economic damages did not resolve the problem: still have ~8% of events that are not properly classified. We will first remove all of the correctly classified entries, then focus on the improperly classified entries to try to match them to the appropriate allowed event type.

```r
# filter to only keep allowed event types.
tidyDataDamagesAllowed <- tidyDataDamages %>%
    filter(ALLOWED == "ALLOWED")

# get data that we need to fix event type i.e.
# NON-ALLOWED
tidyDataDamagesNonAllowed <- tidyDataDamages %>%
    filter(ALLOWED == "NOT_ALLOWED")

# look at which event types still need to be corrected
head(unique(tidyDataDamagesNonAllowed$EVTYPE), n = 25)
```

```
##  [1] "ICE STORM/FLASH FLOOD"          "HURRICANE OPAL/HIGH WINDS"
##  [3] "THUNDERSTORM WINDS"             "HURRICANE ERIN"
##  [5] "HURRICANE OPAL"                 "THUNDERSTORM WINS"
##  [7] "FLASH FLOODING"                 "TORNADO F0"
##  [9] "THUNDERSTORM WINDS LIGHTNING"   "THUNDERSTORM WINDS/HAIL"
## [11] "HIGH WINDS"                     "WIND"
## [13] "HEAVY RAINS"                    "LIGHTNING AND HEAVY RAIN"
## [15] "THUNDERSTORM WINDS HAIL"        "COLD"
## [17] "HEAVY RAIN/LIGHTNING"           "FLASH FLOODING/THUNDERSTORM WI"
## [19] "FLOODING"                       "EXTREME COLD"
## [21] "LIGHTNING/HEAVY RAIN"           "BREAKUP FLOODING"
## [23] "FREEZE"                         "RIVER FLOOD"
## [25] "HIGH WINDS HEAVY RAINS"
```

We will use a "fuzzy join" approach to try to match the coded event type to the closest allowed event type. This essentially works by calculating a distance matrix between the coded string and each of the strings in the allowed events vector, then returns the value with the shortest distance. It is implemented with the `fuzzyjoin` package, for which more details can be found here.

```r
allowedEvents <- as_tibble(allowed_events)
colnames(allowedEvents) <- c("EVTYPE")

not_allowed <- as_tibble(unique(tidyDataDamagesNonAllowed$EVTYPE))
colnames(not_allowed) <- c("EVTYPE")
```

There are multiple methods for fuzzy joining, let's see which one works best to accurately find matches for our non-allowed events. We'll create a custom function to loop through all of the available methods, and return a dataframe containing the method name, number of remaining unmatched event types, and then number of event types that a fuzzy join found a corresponding match for.

```
string_match_test <- function(x, y, method = "lv", ...) {

    match <- stringdist_join(x, y, mode = "left", ignore_case = FALSE,
        method = method)
    colnames(match)[1] <- c("Test")
    colnames(match)[2] <- c("Matched")

    matches <- match %>%
        mutate(Match = case_when(Test == Matched ~ "Exact",
            is.na(Matched) ~ "Unmatched", TRUE ~ "Replaced")) %>%
        count(Match)

    data.frame(matches)
}

match_methods <- c("osa", "lv", "dl", "hamming", "lcs",
    "qgram", "cosine", "jaccard", "jw", "soundex")

test <- map(match_methods, string_match_test, x = not_allowed,
    y = allowedEvents)
names(test) <- match_methods

out <- bind_rows(test, .id = "Method")
pivot_wider(out, id_cols = Method, names_from = Match, values_from = n)
```

```
## # A tibble: 10 x 3
##    Method  Replaced Unmatched
##    <chr>      <int>     <int>
##  1 osa           39       353
##  2 lv            39       353
##  3 dl            39       353
##  4 hamming        7       385
##  5 lcs           38       354
##  6 qgram         40       352
##  7 cosine     21560        NA
##  8 jaccard    21560        NA
##  9 jw         21560        NA
## 10 soundex      219       186
```

From this output, it looks like the **soundex** method found the most matches. You can read more about how this method works here. Let's look at the matches from the soundex method to make sure that it is finding accurate matches.

```
soundex <- stringdist_join(not_allowed, allowedEvents, method = "soundex")
colnames(soundex) <- c("Not_Allowed_EVTYPE", "Matched_Allowed_EVTYPE")
kable(head(soundex, n = 30), booktabs = TRUE) %>%
    kable_styling(latex_options = "striped")
```

| Not_Allowed_EVTYPE | Matched_Allowed_EVTYPE |
|---|---|
| ICE STORM/FLASH FLOOD | ICE STORM |
| HURRICANE OPAL/HIGH WINDS | HURRICANE (TYPHOON) |
| THUNDERSTORM WINDS | THUNDERSTORM WIND |
| HURRICANE ERIN | HURRICANE (TYPHOON) |
| HURRICANE OPAL | HURRICANE (TYPHOON) |
| THUNDERSTORM WINS | THUNDERSTORM WIND |
| FLASH FLOODING | FLASH FLOOD |
| TORNADO F0 | TORNADO |
| THUNDERSTORM WINDS LIGHTNING | THUNDERSTORM WIND |
| THUNDERSTORM WINDS/HAIL | THUNDERSTORM WIND |
| HIGH WINDS | HIGH WIND |
| HEAVY RAINS | HEAVY RAIN |
| LIGHTNING AND HEAVY RAIN | LIGHTNING |
| THUNDERSTORM WINDS HAIL | THUNDERSTORM WIND |
| HEAVY RAIN/LIGHTNING | HEAVY RAIN |
| FLASH FLOODING/THUNDERSTORM WI | FLASH FLOOD |
| EXTREME COLD | EXTREME COLD/WIND CHILL |
| LIGHTNING/HEAVY RAIN | LIGHTNING |
| HIGH WINDS HEAVY RAINS | HIGH WIND |
| HIGH WIND/SEAS | HIGH WIND |
| HIGH WINDS/HEAVY RAIN | HIGH WIND |
| HEAVY SNOW/WIND | HEAVY SNOW |
| THUNDERSTORM WINDS/FUNNEL CLOU | THUNDERSTORM WIND |
| WILD FIRES | WILDFIRE |
| WINTER STORM HIGH WINDS | WINTER STORM |
| WINTER STORM HIGH WINDS | WINTER WEATHER |
| WINTER STORMS | WINTER STORM |
| WINTER STORMS | WINTER WEATHER |
| THUNDERSTORMS WINDS | THUNDERSTORM WIND |
| THUNDERSTORMS | THUNDERSTORM WIND |

|  | ALLOWED | NONALLOWED | TOTAL | FRAC.MISSING |
|---|---|---|---|---|
| FATALITIES | 14577 | 638 | 15215 | 0.0419323 |
| INJURIES | 138640 | 2241 | 140881 | 0.0159070 |
| PROPDMG | 462417891325 | 8304406937 | 470722298262 | 0.0176418 |
| CROPDMG | 42863410562 | 6267401055 | 49130811617 | 0.1275656 |

These look like good matches! We will use the matched events from the soundex approach to replace the non-allowed event types in the dataset.

```
tidyDataDamagesNonAllowed <- left_join(tidyDataDamagesNonAllowed,
    soundex, by = c(EVTYPE = "Not_Allowed_EVTYPE"))

tidyDataDamagesFixed <- tidyDataDamagesNonAllowed %>%
    mutate(EVTYPE = Matched_Allowed_EVTYPE, ALLOWED = if_else(EVTYPE %in%
        allowed_events, true = "ALLOWED", false = "NOT_ALLOWED")) %>%
    select(-Matched_Allowed_EVTYPE)
with(tidyDataDamagesFixed, table(ALLOWED))
```

```
## ALLOWED
##     ALLOWED NOT_ALLOWED
##       15999        2798
```

Now that that we've fixed the event types, there are only 2798 non-allowed event types out of a total of 254,633 total events with damages. That works out to ~1.1% of data that still isn't an allowed type. Let's see how many fatalaties, injuries, and financial damages aren't accounted for in the remaining non-allowed events.

```
# recombine fixed with allowed for full dataframe

final_df <- bind_rows(tidyDataDamagesAllowed, tidyDataDamagesFixed)

damagesSummary <- final_df %>%
    select(ALLOWED, FATALITIES, INJURIES, PROPDMG, CROPDMG) %>%
    group_by(ALLOWED) %>%
    summarise(across(everything(), ~sum(.x))) %>%
    select(-ALLOWED) %>%
    t() %>%
    as.data.frame() %>%
    rename(ALLOWED = V1, NONALLOWED = V2) %>%
    mutate(TOTAL = NONALLOWED + ALLOWED, FRAC.MISSING = NONALLOWED/TOTAL)
kable(damagesSummary, booktabs = TRUE) %>%
    kable_styling(latex_options = "striped")
```

From this, it looks like we've accounted for ~95% of fatalaties, and ~98% of injuries, property damage and crop damage. We will go ahead and move forward with this dataset where the majority of the damages have been properly assigned into an allowed event type.

### Revisiting our Objectives

Our task for analysis is to answer the following two questions:

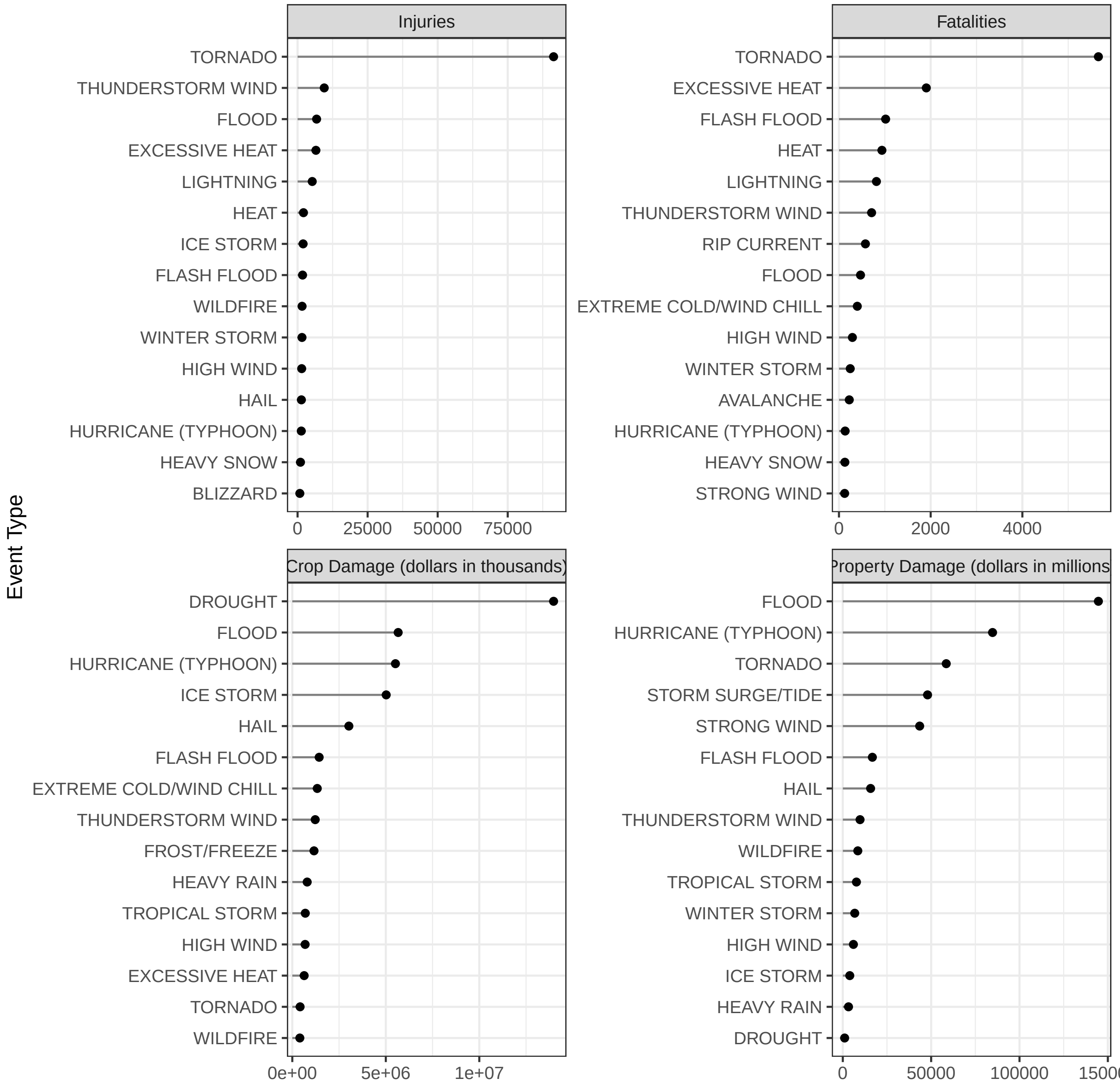
1. Across the United States, which types of events (as indicated in the `EVTYPE` variable) are most harmful with respect to population health?
2. Across the United States, which types of events have the greatest economic consequences?

Let's address the first question first with respect to population health.

```
# let's make a summary table grouped by event type

summary_output <- final_df %>%
    filter(!is.na(EVTYPE)) %>%
    group_by(EVTYPE) %>%
    summarise(across(.cols = c("FATALITIES", "INJURIES",
        "PROPDMG", "CROPDMG"), ~sum(.))) %>%
    mutate(PROPDMG = PROPDMG/1e+06, CROPDMG = CROPDMG/1000) %>%
    rename(`Event Type` = EVTYPE, Fatalities = FATALITIES,
        Injuries = INJURIES, `Property Damage (dollars in millions)` = PROPDMG,
        `Crop Damage (dollars in thousands)` = CROPDMG)
kable(summary_output, booktabs = TRUE) %>%
    kable_styling(latex_options = "striped", font_size = 7)
```

```
# pivot the summary table to long format for ggploting

plot_df <- summary_output %>%
    pivot_longer(cols = c(2:5)) %>%
    group_by(name) %>%
    slice_max(order_by = value, n = 15) %>%
    ungroup() %>%
    mutate(name = factor(name, levels = c("Injuries", "Fatalities",
        "Crop Damage (dollars in thousands)", "Property Damage (dollars in millions)")))



p <- ggplot(plot_df, aes(x = value, y = reorder_within(`Event Type`,
    value, name))) + geom_segment(aes(yend = reorder_within(`Event Type`,
    value, name)), xend = 0, colour = "grey50") + geom_point() +
    scale_y_reordered() + facet_wrap(~name, scales = "free") +
    xlab("") + ylab("Event Type") + theme_bw()

p
```

| Event Type | Fatalities | Injuries | Property Damage (dollars in millions) | Crop Damage (dollars in thousands) |
|---|---|---|---|---|
| ASTRONOMICAL LOW TIDE | 0 | 0 | 9.74500 | 0.00 |
| AVALANCHE | 225 | 170 | 3.72180 | 0.00 |
| BLIZZARD | 101 | 805 | 659.71395 | 112060.00 |
| COASTAL FLOOD | 10 | 9 | 428.88206 | 56.00 |
| COLD/WIND CHILL | 112 | 12 | 1.99000 | 66600.00 |
| DENSE FOG | 18 | 342 | 9.67400 | 0.00 |
| DENSE SMOKE | 0 | 0 | 0.10000 | 0.00 |
| DROUGHT | 2 | 4 | 1046.10600 | 13972571.78 |
| DUST DEVIL | 2 | 43 | 0.71913 | 0.00 |
| DUST STORM | 22 | 440 | 5.59900 | 3600.00 |
| EXCESSIVE HEAT | 1905 | 6548 | 9.68870 | 634402.00 |
| EXTREME COLD/WIND CHILL | 400 | 415 | 77.30540 | 1335023.00 |
| FLASH FLOOD | 1018 | 1785 | 16732.86918 | 1437153.16 |
| FLOOD | 470 | 6789 | 144657.70981 | 5661968.45 |
| FREEZING FOG | 11 | 38 | 10.59850 | 0.00 |
| FROST/FREEZE | 1 | 3 | 15.99500 | 1160686.00 |
| FUNNEL CLOUD | 0 | 3 | 0.19460 | 0.00 |
| HAIL | 15 | 1361 | 15732.81954 | 3025627.89 |
| HEAT | 937 | 2100 | 1.79700 | 401461.50 |
| HEAVY RAIN | 98 | 255 | 3230.99814 | 795755.80 |
| HEAVY SNOW | 129 | 1034 | 952.92715 | 134673.10 |
| HIGH SURF | 104 | 156 | 90.15500 | 0.00 |
| HIGH WIND | 293 | 1471 | 6003.35604 | 686301.90 |
| HURRICANE (TYPHOON) | 135 | 1328 | 84756.18001 | 5515292.80 |
| ICE STORM | 89 | 1977 | 3948.42786 | 5022113.50 |
| LAKE-EFFECT SNOW | 0 | 0 | 40.18200 | 0.00 |
| LAKESHORE FLOOD | 0 | 0 | 7.54000 | 0.00 |
| LIGHTNING | 817 | 5232 | 933.73745 | 12092.09 |
| MARINE HAIL | 0 | 0 | 0.00400 | 0.00 |
| MARINE HIGH WIND | 2 | 3 | 1.34701 | 0.00 |
| MARINE STRONG WIND | 15 | 24 | 0.46833 | 0.00 |
| MARINE THUNDERSTORM WIND | 19 | 34 | 5.85740 | 50.00 |
| RIP CURRENT | 577 | 529 | 0.16300 | 0.00 |
| SEICHE | 0 | 0 | 0.98000 | 0.00 |
| SLEET | 2 | 0 | 0.00000 | 0.00 |
| STORM SURGE/TIDE | 32 | 64 | 47967.33879 | 5855.00 |
| STRONG WIND | 124 | 339 | 43501.41024 | 69958.50 |
| THUNDERSTORM WIND | 712 | 9509 | 9762.92226 | 1225454.99 |
| TORNADO | 5658 | 91368 | 58541.93248 | 417461.47 |
| TROPICAL DEPRESSION | 8 | 43 | 12.23700 | 16550.00 |
| TROPICAL STORM | 66 | 383 | 7714.39055 | 694896.00 |
| TSUNAMI | 33 | 129 | 144.06200 | 20.00 |
| VOLCANIC ASH | 0 | 0 | 0.50000 | 0.00 |
| WATERSPOUT | 6 | 72 | 60.73020 | 0.00 |
| WILDFIRE | 90 | 1606 | 8491.56350 | 402781.63 |
| WINTER STORM | 246 | 1570 | 6755.44175 | 32444.00 |
| WINTER WEATHER | 73 | 647 | 87.81050 | 20500.00 |


Injuries
Fatalities
Crop Damage (dollars in thousands)
Property Damage (dollars in millions
Event Type
TORNADO
THUNDERSTORM WIND
FLOOD
EXCESSIVE HEAT
LIGHTNING
HEAT
ICE STORM
FLASH FLOOD
WILDFIRE
WINTER STORM
HIGH WIND
HAIL
HURRICANE (TYPHOON)
HEAVY SNOW
BLIZZARD
0
25000
50000
75000
TORNADO
EXCESSIVE HEAT
FLASH FLOOD
HEAT
LIGHTNING
THUNDERSTORM WIND
RIP CURRENT
FLOOD
EXTREME COLD/WIND CHILL
HIGH WIND
WINTER STORM
AVALANCHE
HURRICANE (TYPHOON)
HEAVY SNOW
STRONG WIND
0
2000
4000
DROUGHT
FLOOD
HURRICANE (TYPHOON)
ICE STORM
HAIL
FLASH FLOOD
EXTREME COLD/WIND CHILL
THUNDERSTORM WIND
FROST/FREEZE
HEAVY RAIN
TROPICAL STORM
HIGH WIND
EXCESSIVE HEAT
TORNADO
WILDFIRE
0e+00
5e+06
1e+07
FLOOD
HURRICANE (TYPHOON)
TORNADO
STORM SURGE/TIDE
STRONG WIND
FLASH FLOOD
HAIL
THUNDERSTORM WIND
WILDFIRE
TROPICAL STORM
WINTER STORM
HIGH WIND
ICE STORM
HEAVY RAIN
DROUGHT
0
50000
100000