Analysis of U.S. Storm Event Data and the Impact on Population Health and the Economy

Ajay Kirthik G

16 April, 2024

Contents

Synonpsis

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 report contains the results of an analysis where the goal was to identify the most hazardous weather events with respect to population health and those with the greatest economic impact in the U.S. based on data collected from the U.S. National Oceanic and Atmospheric Administration's (NOAA).

The storm database includes weather events from 1950 through the year 2011 and contains data estimates such as the number fatalities and injuries for each weather event as well as economic cost damage to properties and crops for each weather event.

The estimates for fatalities and injuries were used to determine weather events with the most harmful impact to population health. Property damage and crop damage cost estimates were used to determine weather events with the greatest economic consequences.

Environment Setup

Load packages used in this analysis.

```
if (!require(ggplot2)) {
    install.packages("ggplot2")
    library(ggplot2)
}
```

```
## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
if (!require(dplyr)) {
    install.packages("dplyr")
    library(dplyr, warn.conflicts = FALSE)
}
```

```
## Loading required package: dplyr

## Warning: package 'dplyr' was built under R version 4.3.3

##
## 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
```

```
if (!require(xtable)) {
    install.packages("xtable")
    library(xtable, warn.conflicts = FALSE)
}
```

```
## Loading required package: xtable
```

Display session information.

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_India.utf8  LC_CTYPE=English_India.utf8
## [3] LC_MONETARY=English_India.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_India.utf8
##
## time zone: Asia/Calcutta
## tzcode source: internal
##
## attached base packages:
```

```
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] xtable_1.8-4  dplyr_1.1.4   ggplot2_3.5.0
##
## loaded via a namespace (and not attached):
##  [1] vctrs_0.6.5       cli_3.6.2         knitr_1.45        rlang_1.1.3
##  [5] xfun_0.41         generics_0.1.3    glue_1.7.0        colorspace_2.1-0
##  [9] htmltools_0.5.7   scales_1.3.0      fansi_1.0.6       rmarkdown_2.25
## [13] grid_4.3.2        evaluate_0.23     munsell_0.5.0     tibble_3.2.1
## [17] fastmap_1.1.1     yaml_2.3.8        lifecycle_1.0.4   compiler_4.3.2
## [21] pkgconfig_2.0.3   rstudioapi_0.15.0 digest_0.6.34     R6_2.5.1
## [25] tidyselect_1.2.0  utf8_1.2.4        pillar_1.9.0      magrittr_2.0.3
## [29] withr_3.0.0       tools_4.3.2       gtable_0.3.4
```

Load Data

Download the compressed data file from the source URL (if not found locally) and then load the compressed data file via `read.csv`. Prior to processing the data, validate the downloaded data file and loaded dataset by checking the file size and dimensions respectively.

```r
#setwd("~/P:")
stormDataFileURL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
stormDataFile <- "data/storm-data.csv.bz2"
if (!file.exists('data')) {
    dir.create('data')
}
if (!file.exists(stormDataFile)) {
    download.file(url = stormDataFileURL, destfile = stormDataFile)
}
stormData <- read.csv(stormDataFile, sep = ",", header = TRUE)
stopifnot(file.size(stormDataFile) == 49177144)
stopifnot(dim(stormData) == c(902297,37))
```

Display dataset summary

```r
names(stormData)
```

```
##  [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"
```

```r
str(stormData)
```

```
## 'data.frame':    902297 obs. of  37 variables:
##  $ STATE__   : num  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 0:00:00" "6/8/1951 0:00:00" .
```

```
##  $ BGN_TIME  : chr  "0130" "0145" "1600" "0900" ...
##  $ TIME_ZONE : chr  "CST" "CST" "CST" "CST" ...
##  $ COUNTY    : num  97 3 57 89 43 77 9 123 125 57 ...
##  $ COUNTYNAME: chr  "MOBILE" "BALDWIN" "FAYETTE" "MADISON" ...
##  $ STATE     : chr  "AL" "AL" "AL" "AL" ...
##  $ EVTYPE    : chr  "TORNADO" "TORNADO" "TORNADO" "TORNADO" ...
##  $ BGN_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ BGN_AZI   : chr  "" "" "" "" ...
##  $ BGN_LOCATI: chr  "" "" "" "" ...
##  $ END_DATE  : chr  "" "" "" "" ...
##  $ END_TIME  : chr  "" "" "" "" ...
##  $ COUNTY_END: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ COUNTYENDN: logi  NA NA NA NA NA NA ...
##  $ END_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ END_AZI   : chr  "" "" "" "" ...
##  $ END_LOCATI: chr  "" "" "" "" ...
##  $ LENGTH    : num  14 2 0.1 0 0 1.5 1.5 0 3.3 2.3 ...
##  $ WIDTH     : num  100 150 123 100 150 177 33 33 100 100 ...
##  $ F         : int  3 2 2 2 2 2 2 1 3 3 ...
##  $ MAG       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ FATALITIES: num  0 0 0 0 0 0 0 0 1 0 ...
##  $ INJURIES  : num  15 0 2 2 2 6 1 0 14 0 ...
##  $ PROPDMG   : num  25 2.5 25 2.5 2.5 2.5 2.5 2.5 25 25 ...
##  $ PROPDMGEXP: chr  "K" "K" "K" "K" ...
##  $ CROPDMG   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CROPDMGEXP: chr  "" "" "" "" ...
##  $ WFO       : chr  "" "" "" "" ...
##  $ STATEOFFIC: chr  "" "" "" "" ...
##  $ ZONENAMES : chr  "" "" "" "" ...
##  $ LATITUDE  : num  3040 3042 3340 3458 3412 ...
##  $ LONGITUDE : num  8812 8755 8742 8626 8642 ...
##  $ LATITUDE_E: num  3051 0 0 0 0 ...
##  $ LONGITUDE_: num  8806 0 0 0 0 ...
##  $ REMARKS   : chr  "" "" "" "" ...
##  $ REFNUM    : num  1 2 3 4 5 6 7 8 9 10 ...
```

```r
head(stormData)
```

STATE__	BGN_DATE	BGN_TIME	TIME_ZONE	COUNTY	COUNTYNAME	STATE	EVTYPE	BGN_RANGE	BGN_AZI	BGN_LOCATI	END_DATE	END_TIME	COUNTY_END	COUNTYENDN	END_RANGE	END_AZI	END_LOCATI	LENGTH	WIDTH	F	MAG	FATALITIES	INJURIES	PROPDMG	PROPDMGEXP	CROPDMG	CROPDMGEXP	WFO	STATEOFFIC	ZONENAMES	LATITUDE	LONGITUDE	LATITUDE_E	LONGITUDE_	REMARKS	REFNUM
1	4/18/1950 0:00:00	0130	CST	97	MOBILE	AL	TORNADO	0					0	NA	0			14.0	100	3	0	0	15	25.0	K	0					3040	8812	3051	8806		1
1	4/18/1950 0:00:00	0145	CST	3	BALDWIN	AL	TORNADO	0					0	NA	0			2.0	150	2	0	0	0	2.5	K	0					3042	8755	0	0		2
1	2/20/1951 0:00:00	1600	CST	57	FAYETTE	AL	TORNADO	0					0	NA	0			0.1	123	2	0	0	2	25.0	K	0					3340	8742	0	0		3
1	6/8/1951 0:00:00	0900	CST	89	MADISON	AL	TORNADO	0					0	NA	0			0.0	100	2	0	0	2	2.5	K	0					3458	8626	0	0		4
1	11/15/1951 0:00:00	1500	CST	43	CULLMAN	AL	TORNADO	0					0	NA	0			0.0	150	2	0	0	2	2.5	K	0					3412	8642	0	0		5
1	11/15/1951 0:00:00	2000	CST	77	LAUDERDALE	AL	TORNADO	0					0	NA	0			1.5	177	2	0	0	6	2.5	K	0					3450	8748	0	0		6

Data Processing

Create Subset of Data

When processing a large dataset, compute performance can be improved by taking a subset of the variables required for the analysis. For this analysis, the dataset will be trimmed to only include the necessary variables (listed below). In addition, only observations with `value > 0` will be included.

Variable	Description
EVTYPE	Event type (Flood, Heat, Hurricane, Tornado, ...)
FATALITIES	Number of fatalities resulting from event
INJURIES	Number of injuries resulting from event
PROPDMG	Property damage in USD
PROPDMGEXP	Unit multiplier for property damage (K, M, or B)
CROPDMG	Crop damage in USD
CROPDMGEXP	Unit multiplier for property damage (K, M, or B)
BGN_DATE	Begin date of the event
END_DATE	End date of the event
STATE	State where the event occurred

```
stormDataTidy <- subset(stormData, EVTYPE != "?"
                                   &
                                   (FATALITIES > 0 | INJURIES > 0 | PROPDMG > 0 | CROPDMG > 0),
                                   select = c("EVTYPE",
                                              "FATALITIES",
                                              "INJURIES",
                                              "PROPDMG",
                                              "PROPDMGEXP",
                                              "CROPDMG",
                                              "CROPDMGEXP",
                                              "BGN_DATE",
                                              "END_DATE",
                                              "STATE"))
dim(stormDataTidy)
```

```
## [1] 254632     10
```

```
sum(is.na(stormDataTidy))
```

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

The working (tidy) dataset contains 254632 observations, 10 variables and no missing values.

Clean Event Type Data

There are a total of 487 unique Event Type values in the current tidy dataset.

```
length(unique(stormDataTidy$EVTYPE))
```

```
## [1] 487
```

Exploring the Event Type data revealed many values that appeared to be similar; however, they were entered with different spellings, pluralization, mixed case and even misspellings. For example, `Strong Wind`, `STRONG WIND`, `Strong Winds`, and `STRONG WINDS`.

The dataset was normalized by converting all Event Type values to uppercase and combining similar Event Type values into unique categories.

```
stormDataTidy$EVTYPE <- toupper(stormDataTidy$EVTYPE)
```

```
# AVALANCHE
stormDataTidy$EVTYPE <- gsub('.*AVALANCE.*', 'AVALANCHE', stormDataTidy$EVTYPE)

# BLIZZARD
stormDataTidy$EVTYPE <- gsub('.*BLIZZARD.*', 'BLIZZARD', stormDataTidy$EVTYPE)

# CLOUD
stormDataTidy$EVTYPE <- gsub('.*CLOUD.*', 'CLOUD', stormDataTidy$EVTYPE)

# COLD
stormDataTidy$EVTYPE <- gsub('.*COLD.*', 'COLD', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*FREEZ.*', 'COLD', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*FROST.*', 'COLD', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*ICE.*', 'COLD', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*LOW TEMPERATURE RECORD.*', 'COLD', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*LO.*TEMP.*', 'COLD', stormDataTidy$EVTYPE)

# DRY
stormDataTidy$EVTYPE <- gsub('.*DRY.*', 'DRY', stormDataTidy$EVTYPE)

# DUST
stormDataTidy$EVTYPE <- gsub('.*DUST.*', 'DUST', stormDataTidy$EVTYPE)

# FIRE
stormDataTidy$EVTYPE <- gsub('.*FIRE.*', 'FIRE', stormDataTidy$EVTYPE)

# FLOOD
stormDataTidy$EVTYPE <- gsub('.*FLOOD.*', 'FLOOD', stormDataTidy$EVTYPE)

# FOG
stormDataTidy$EVTYPE <- gsub('.*FOG.*', 'FOG', stormDataTidy$EVTYPE)

# HAIL
stormDataTidy$EVTYPE <- gsub('.*HAIL.*', 'HAIL', stormDataTidy$EVTYPE)

# HEAT
stormDataTidy$EVTYPE <- gsub('.*HEAT.*', 'HEAT', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*WARM.*', 'HEAT', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*HIGH.*TEMP.*', 'HEAT', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*RECORD HIGH TEMPERATURES.*', 'HEAT', stormDataTidy$EVTYPE)

# HYPOTHERMIA/EXPOSURE
stormDataTidy$EVTYPE <- gsub('.*HYPOTHERMIA.*', 'HYPOTHERMIA/EXPOSURE', stormDataTidy$EVTYPE)

# LANDSLIDE
```

```r
stormDataTidy$EVTYPE <- gsub('.*LANDSLIDE.*', 'LANDSLIDE', stormDataTidy$EVTYPE)

# LIGHTNING
stormDataTidy$EVTYPE <- gsub('^LIGHTNING.*', 'LIGHTNING', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('^LIGNTNING.*', 'LIGHTNING', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('^LIGHTING.*', 'LIGHTNING', stormDataTidy$EVTYPE)

# MICROBURST
stormDataTidy$EVTYPE <- gsub('.*MICROBURST.*', 'MICROBURST', stormDataTidy$EVTYPE)

# MUDSLIDE
stormDataTidy$EVTYPE <- gsub('.*MUDSLIDE.*', 'MUDSLIDE', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*MUD SLIDE.*', 'MUDSLIDE', stormDataTidy$EVTYPE)

# RAIN
stormDataTidy$EVTYPE <- gsub('.*RAIN.*', 'RAIN', stormDataTidy$EVTYPE)

# RIP CURRENT
stormDataTidy$EVTYPE <- gsub('.*RIP CURRENT.*', 'RIP CURRENT', stormDataTidy$EVTYPE)

# STORM
stormDataTidy$EVTYPE <- gsub('.*STORM.*', 'STORM', stormDataTidy$EVTYPE)

# SUMMARY
stormDataTidy$EVTYPE <- gsub('.*SUMMARY.*', 'SUMMARY', stormDataTidy$EVTYPE)

# TORNADO
stormDataTidy$EVTYPE <- gsub('.*TORNADO.*', 'TORNADO', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*TORNDAO.*', 'TORNADO', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*LANDSPOUT.*', 'TORNADO', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*WATERSPOUT.*', 'TORNADO', stormDataTidy$EVTYPE)

# SURF
stormDataTidy$EVTYPE <- gsub('.*SURF.*', 'SURF', stormDataTidy$EVTYPE)

# VOLCANIC
stormDataTidy$EVTYPE <- gsub('.*VOLCANIC.*', 'VOLCANIC', stormDataTidy$EVTYPE)

# WET
stormDataTidy$EVTYPE <- gsub('.*WET.*', 'WET', stormDataTidy$EVTYPE)

# WIND
stormDataTidy$EVTYPE <- gsub('.*WIND.*', 'WIND', stormDataTidy$EVTYPE)

# WINTER
stormDataTidy$EVTYPE <- gsub('.*WINTER.*', 'WINTER', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*WINTRY.*', 'WINTER', stormDataTidy$EVTYPE)
stormDataTidy$EVTYPE <- gsub('.*SNOW.*', 'WINTER', stormDataTidy$EVTYPE)
```

After tidying the dataset, the number of unique Event Type values were reduced to 81

```r
length(unique(stormDataTidy$EVTYPE))
```

```
## [1] 81
```

Clean Date Data

Format date variables for any type of optional reporting or further analysis.

In the raw dataset, the `BNG_START` and `END_DATE` variables are stored as factors which should be made available as actual *date* types that can be manipulated and reported on. For now, time variables will be ignored.

Create four new variables based on date variables in the tidy dataset:

Variable	Description
DATE_START	Begin date of the event stored as a date type
DATE_END	End date of the event stored as a date type
YEAR	Year the event started
DURATION	Duration (in hours) of the event

```
stormDataTidy$DATE_START <- as.Date(stormDataTidy$BGN_DATE, format = "%m/%d/%Y")
stormDataTidy$DATE_END <- as.Date(stormDataTidy$END_DATE, format = "%m/%d/%Y")
stormDataTidy$YEAR <- as.integer(format(stormDataTidy$DATE_START, "%Y"))
stormDataTidy$DURATION <- as.numeric(stormDataTidy$DATE_END - stormDataTidy$DATE_START)/3600
```

Clean Economic Data

According to the "National Weather Service Storm Data Documentation" (page 12), information about Property Damage is logged using two variables: `PROPDMG` and `PROPDMGEXP`. `PROPDMG` is the mantissa (the significand) rounded to three significant digits and `PROPDMGEXP` is the exponent (the multiplier). The same approach is used for Crop Damage where the `CROPDMG` variable is encoded by the `CROPDMGEXP` variable.

The documentation also specifies that the `PROPDMGEXP` and `CROPDMGEXP` are supposed to contain an alphabetical character used to signify magnitude and logs "K" for thousands, "M" for millions, and "B" for billions. A quick review of the data, however, shows that there are several other characters being logged.

```
table(toupper(stormDataTidy$PROPDMGEXP))
```

```
##
##              -      +      0      2      3      4      5      6      7      B
##  11585      1      5    210      1      1      4     18      3      3     40
##      H      K      M
##      7 231427  11327
```

```
table(toupper(stormDataTidy$CROPDMGEXP))
```

```
##
##             ?      0      B      K      M
## 152663      6     17      7  99953   1986
```

In order to calculate costs, the `PROPDMGEXP` and `CROPDMGEXP` variables will be mapped to a multiplier factor which will then be used to calculate the actual costs for both property and crop damage. Two new variables will be created to store damage costs:

- PROP_COST
- CROP_COST

```r
# function to get multiplier factor
getMultiplier <- function(exp) {
    exp <- toupper(exp);
    if (exp == "")  return (10^0);
    if (exp == "-") return (10^0);
    if (exp == "?") return (10^0);
    if (exp == "+") return (10^0);
    if (exp == "0") return (10^0);
    if (exp == "1") return (10^1);
    if (exp == "2") return (10^2);
    if (exp == "3") return (10^3);
    if (exp == "4") return (10^4);
    if (exp == "5") return (10^5);
    if (exp == "6") return (10^6);
    if (exp == "7") return (10^7);
    if (exp == "8") return (10^8);
    if (exp == "9") return (10^9);
    if (exp == "H") return (10^2);
    if (exp == "K") return (10^3);
    if (exp == "M") return (10^6);
    if (exp == "B") return (10^9);
    return (NA);
}

# calculate property damage and crop damage costs (in billions)
stormDataTidy$PROP_COST <- with(stormDataTidy, as.numeric(PROPDMG) * sapply(PROPDMGEXP, getMultiplier))/
stormDataTidy$CROP_COST <- with(stormDataTidy, as.numeric(CROPDMG) * sapply(CROPDMGEXP, getMultiplier))/
```

Summarize Data

Create a summarized dataset of health impact data (fatalities + injuries). Sort the results in descending order by health impact.

```r
healthImpactData <- aggregate(x = list(HEALTH_IMPACT = stormDataTidy$FATALITIES + stormDataTidy$INJURIE
                                  by = list(EVENT_TYPE = stormDataTidy$EVTYPE),
                                  FUN = sum,
                                  na.rm = TRUE)
healthImpactData <- healthImpactData[order(healthImpactData$HEALTH_IMPACT, decreasing = TRUE),]
```

Create a summarized dataset of damage impact costs (property damage + crop damage). Sort the results in descending order by damage cost.

```r
damageCostImpactData <- aggregate(x = list(DAMAGE_IMPACT = stormDataTidy$PROP_COST + stormDataTidy$CROP_
                                  by = list(EVENT_TYPE = stormDataTidy$EVTYPE),
                                  FUN = sum,
                                  na.rm = TRUE)
damageCostImpactData <- damageCostImpactData[order(damageCostImpactData$DAMAGE_IMPACT, decreasing = TRU
```

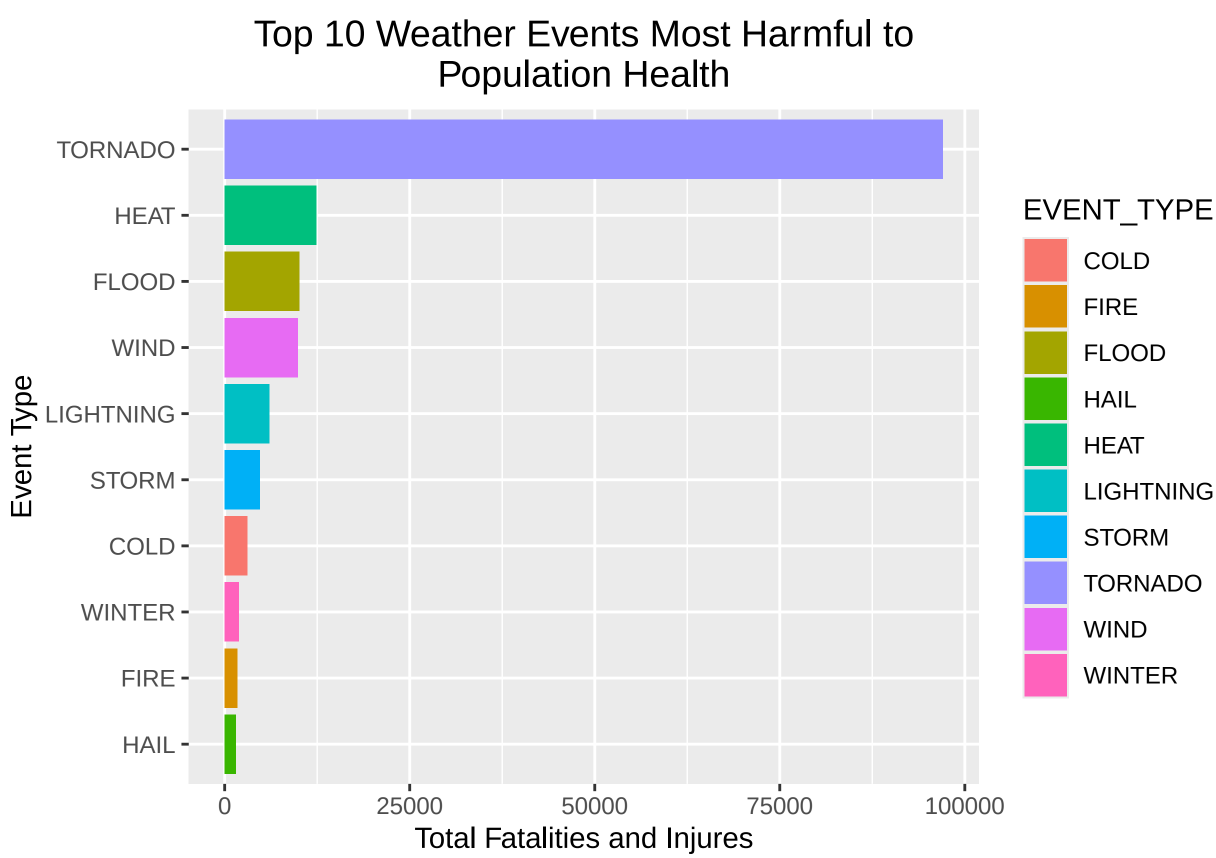
Results

Event Types Most Harmful to Population Health

Fatalities and injuries have the most harmful impact on population health. The results below display the 10 most harmful weather events in terms of population health in the U.S.

```
print(xtable(head(healthImpactData, 10),
             caption = "Top 10 Weather Events Most Harmful to Population Health"),
             caption.placement = 'top',
             type = "html",
             include.rownames = FALSE,
             html.table.attributes='class="table-bordered", width="100%"')
```

```
healthImpactChart <- ggplot(head(healthImpactData, 10),
                            aes(x = reorder(EVENT_TYPE, HEALTH_IMPACT), y = HEALTH_IMPACT, fill = EVENT_
                            coord_flip() +
                            geom_bar(stat = "identity") +
                            xlab("Event Type") +
                            ylab("Total Fatalities and Injures") +
                            theme(plot.title = element_text(size = 14, hjust = 0.5)) +
                            ggtitle("Top 10 Weather Events Most Harmful to\nPopulation Health")
print(healthImpactChart)
```

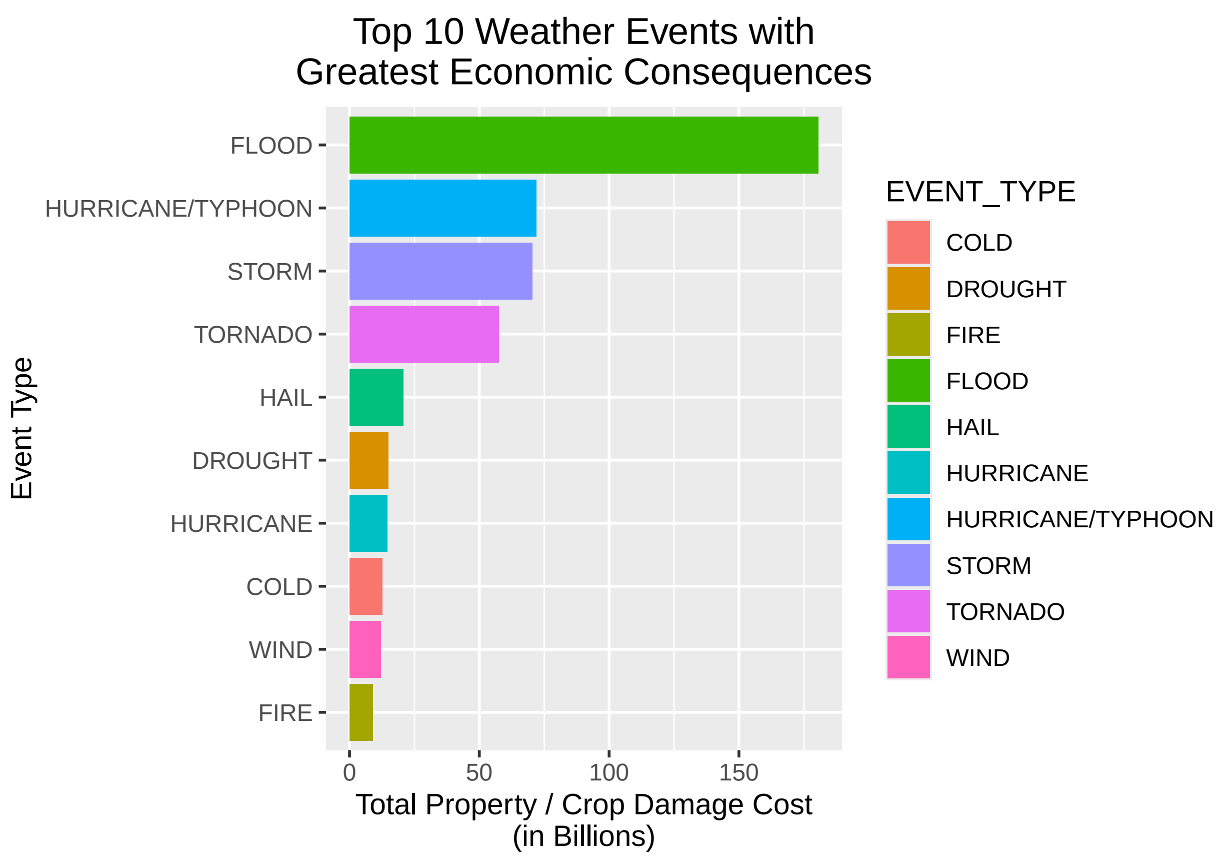


Event Types with Greatest Economic Consequences

Property and crop damage have the most harmful impact on the economy. The results below display the 10 most harmful weather events in terms economic consequences in the U.S.

```
print(xtable(head(damageCostImpactData, 10),
             caption = "Top 10 Weather Events with Greatest Economic Consequences"),
             caption.placement = 'top',
             type = "html",
             include.rownames = FALSE,
             html.table.attributes='class="table-bordered", width="100%"')
```

```
damageCostImpactChart <- ggplot(head(damageCostImpactData, 10),
                            aes(x = reorder(EVENT_TYPE, DAMAGE_IMPACT), y = DAMAGE_IMPACT, fill = EVENT_
                            coord_flip() +
                            geom_bar(stat = "identity") +
                            xlab("Event Type") +
                            ylab("Total Property / Crop Damage Cost\n(in Billions)") +
                            theme(plot.title = element_text(size = 14, hjust = 0.5)) +
                            ggtitle("Top 10 Weather Events with\nGreatest Economic Consequences")
print(damageCostImpactChart)
```



Conclusion

Based on the evidence demonstrated in this analysis and supported by the included data and graphs, the following conclusions can be drawn:

- **Which types of weather events are most harmful to population health?**

  Tornadoes are responsible for the greatest number of fatalities and injuries.

- **Which types of weather events have the greatest economic consequences?**

  Floods are responsible for causing the most property damage and crop damage costs.